DeepSynthBody: the beginning of the end for data deficiency in medicine

Vajira Lasantha Bandara Thambawita

Ph.D. Programme in Engineering Science

Department of Computer Science

Faculty of Technology, Art and Design

Oslo Metropolitan University, Norway

July 11, 2021

Preface

I wrote this thesis, titled "DeepSynthBody: the beginning of the end for data deficiency in medicine," to fulfill the requirement for completing my Ph.D. for the Ph.D. program in Engineering Science Faculty of Technology, Art and Design, Oslo Metropolitan University, Oslo, Norway. The total time for thesis was around three years. I carried out my work under the supervision of Professor Michael A. Riegler, Professor Pål Halvorsen, and Professor Hugo L. Hammer. I have completed the thesis in the Department of Holistic Systems in Simula Metropolitan Center for Digital Engineering (SimulaMet), which provided the infrastructure and all the financial support to this full research.

This Ph.D. time became a golden period in my life because I have been exploring the real research world which is not limited to a thesis. As a result, I felt my research works and perceived them, which forced me to learn new things every day until I am writing this preface. In addition to the general responsibilities of my life, I was a responsible person for performing quality research works in the medical domain, which is the field no one can argue the importance of it. I was forced to be responsible for this field because the success of our research can save human life and a fault of our research can indirectly cause death.

I hope that you love this thesis reading.

Vajira Thambawita
May, 2021 at Oslo, Norway

Acknowledgments

First, I would like to thank my principal supervisor and two co-supervisors, Michael Alexander Riegler, Pål Halvorsen, and Hugo Lewi Hammer, for their support, motivation, and always behind me. Without them, this would be only a dream. After joining the HOST department as a Ph.D. candidate in 2018, I started experiencing a completely new environment with new people from different countries and cultures.

Michael Riegler became my principal supervisor. I did not know anything about him despite his academic background. However, after few weeks, I realized that he is more than my principal supervisor for my life. Within few months, he became a game-changer in my life. I do not have words to express his qualities and how his advice, encouragement, kindness, and motivations are important to my life. Therefore, I would like to give my most enormous thanks to my primary supervisor, Michael, who is always behind me to support my academic life and get advice for my personal life.

Pål Halvorsen is the department head and one of my co-supervisors, and he is a very kind person supporting us always silently. His advice and encouragements make me better and better every day in my academic life. Then, I would like to thank Pål for his kindness showed me through my Ph.D. journey and the wordless help given to me. Hugo Hammer is my second co-supervisor who supports me in handling advanced statistical problems in my Ph.D. research. Then, I would like to thank Hugo for his friendly help, as always.

The research has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which the Research Council of Norway financially supports under contract 270053. Tore was the person who helped me to use these infrastructures. So, I would like to give my special thanks to him for his outstanding support.

I also would like to thank the closest colleague, Steven Hicks, a good friend met within the department and in my life. Moreover, I thank all the other colleagues, Debesh, Hanna, Pia, Hakon, and Daniel, who work closely in my Ph.D. periods. There are more people

Abstract-English

Recent advancements in technology have made artificial intelligence (AI) a popular tool in the medical domain, especially machine learning (ML) methods, which is a subset of AI. In this context, a goal is to research and develop generalizable and well-performing ML models to be used as the main component in computer-aided diagnosis (CAD) systems. However, collecting and processing medical data has been identified as a major obstacle to produce AI-based solutions in the medical domain. In addition to the focus on the development of ML models, this thesis also aims at finding a solution to the data deficiency problem caused by, for example, privacy concerns and the tedious medical data annotation process.

To accomplish the goals of the thesis, we investigated case studies from three different medical branches, namely cardiology, gastroenterology, and andrology. Using data from these case studies, we developed ML models. Addressing the scarcity of medical data, we collected, analyzed, and developed medical datasets and performed benchmark analyses. A framework for generating synthetic medical data has been developed using generative adversarial networks (GANs) as a solution to address the data deficiency problem. Our results indicate that our generated synthetic data may be a solution to the data challenge. As an overarching concept, we introduced the DeepSynthBody as a basis for structured and centralized synthetic medical data generation. The studies presented in the thesis, such as generating synthetic electrocardiograms (ECGs), gastrointestinal (GI)-tract images and videos with and without polyps, and sperm samples, showed that DeepSynthBody can help to overcome data privacy concerns, the time-consuming and costly data annotation process, and the data imbalance problem in the medical domain. Our experiments showed that our generative models generate realistic synthetic data providing comparable results to experiments using real data to tackle the identified problems. The final DeepSynthBody framework is available as an open-source project that allows researchers, industry, and practitioners to use the system and contribute to future developments.

Abstract-Norwegian

Teknologiske fremskritt har gjort kunstig intelligens til et populært verktøy innen medisin. Spesielt metoder innen maskinlæring, en underkategori av kunstig intelligens, er mye brukt. Et mål i denne fobindelse er å utvikle gode, generaliserbare modeller for bruk i systemer for datamaskinassistert-diagnose, men en stor utfordring her er innsamling og behandling av medisinske data på grunn av for eksempel personvernhensyn og kostbare annoteringsprosesser. Denne oppgaven fokuserer derfor både på utvikling av maskinlæringsmodeller og å finne en løsning på problemet med manglende medisinske data.

For å nå oppgavens mål har vi undersøkt tre forskjellige medisinske eksempler, nemlig kardiologi, gastroenterologi og andrologi. Ved hjelp av data fra disse medisinske områdenehar vi utviklet maskinlæringsmodeller. For å løse mangelen på medisinsk data, har vi samlet inn, analysert og utviklet medisinske datasett, og vi har utført referanseanalyser. I tillegg, et rammeverk for generering av syntetiske medisinske data er utviklet ved hjelp av "generative adversarial networks" for å løse problemet med datamangel, hvor resultatene våre indikerer at slike genererte data kan være en mulig løsning. Som et overordnet konsept introduserer vi DeepSynthBody som grunnlag for strukturert og sentralisert generering av syntetisk medisinsk data. Studiene presentert i oppgaven, slik som generering av syntetiske elektrokardiogram, bilder og videoer fra tarmsystemet og sædprøver, viser at DeepSynthBody kan bidra til å overvinne personvernproblemer, redusere tid og ressursbruk innen dataanmerkingsprosessene, og utjevne problemene med data ubalanse innen det medisinske domenet. Våre eksperimenter viser at vi kan generere realistiske syntetiske data som gir sammenlignbare resultater med eksperimenter hvor man bruker reelle data. Det endelige DeepSynthBody-rammeverket er tilgjengelig som et åpent kildekode-prosjekt som gjør det mulig for både forskere og industri å bruke systemet og å bidra til fremtidig utvikling.

x

Contents

Contents

Contents

Contents

Acronyms

1-D one-dimensional. 16, 28, 46–49, 51

2-D two-dimensional. 16, 28, 29, 46–49, 53

3-D three-dimensional. 16, 28, 29, 46–49, 53, 54

4-D four-dimensional. 16, 28, 98

AI artificial intelligence. vii, 5–7, 9, 10, 12–14, 16, 18, 22, 24, 27, 39, 41, 42, 50, 61, 88

CAD computer-aided diagnosis. vii, 6–8, 13–15, 17–25, 27, 29, 39–41, 57, 58, 86, 89–92, 94–96, 98–101

CNN convolutional neural network. 56, 58–60, 62, 66, 69

DL deep learning. 5, 6, 13, 35, 69, 77, 79

DNN deep neural network. 5, 9, 34

ECG electrocardiogram. vii, 9, 15, 19–22, 24, 28, 43, 46–48, 51–53, 56, 61–65, 78, 82, 83, 87, 91–93, 95, 96, 98, 100

EHR electronic health records. 37, 38, 40

FID Frechet inception distance. 67

GAN generative adversarial network. vii, 14–19, 21–24, 27, 33–41, 47, 50, 53, 54, 60–62, 64, 66, 67, 69–71, 74, 75, 77–83, 87–90, 92–95, 98–101

GDPR general data protection regulation. 10

Acronyms

GI gastrointestinal. vii, 8, 15, 20–22, 24, 31, 32, 43, 46, 47, 51, 53, 54, 57–59, 61, 62, 66–69, 74, 78, 83, 84, 87, 88, 91–94, 98, 99

GMCNN generative multi-column convolutional neural networks. 70

GUI graphical user interface. 61, 79

IOU intersection over union. 33

MAE mean absolute error. 33, 56

MCC Matthews correlation coefficient. 33, 57

ML machine learning. vii, 5–9, 12–15, 17–25, 27, 29–33, 39, 40, 49, 50, 55–60, 66, 78, 79, 84, 86, 87, 89–93, 95, 96, 98–101

MRI magnetic resonance imaging. 16, 29, 43, 48, 82, 98

MSE mean square error. 33

N-D N-dimensional. 47, 48

PyPI Python package index. 24, 77, 95, 100

RMSE root-mean-squared error. 33, 56

VAE variational autoencoder. 33, 34

XAI explainable artificial intelligence. 6, 13, 84, 86

Chapter 1

Introduction

"The data-driven world will be always on, always tracking, always monitoring, always listening and always watching – because it will be always learning" (Rydning [1]).

Artificial intelligence (AI) has become a popular tool in most of the leading industries, for example, financial service [2, 3], manufacturing [4, 5], media and entertainment [6, 7], transportation [8, 9], and healthcare [10, 11]. As a result, AI interacts more closely with the day-to-day life of people. While AI has many definitions, the main goal of AI today is to enable faster, more reliable, and more accurate data analysis. Additionally, AI applies to tasks that humans cannot proceed with, such as operations in space, in deep oceans, or deep underground. These AI applications are successful due to improvements in machine learning (ML) algorithms [12] used in AI, particularly deep learning (DL) [13], and tremendous advances in computational hardware running the compute-heavy ML algorithms, such as deep neural networks (DNNs). Despite such advancements, the algorithms need data to learn. The limited availability of data to train the ML algorithms [14, 15] is crucial in developing successful AI solutions in all domains. The interconnections between the terminology, AI, ML, and DL used in this section are depicted in Figure 1.1.

With the success of applying AI as a tool in the leading industries, using AI in the medical domain has received more attention in the recent decade, such as the news headings[1] and quotes[2] about AI and medicine presented in Figure 1.2. The news shows contradictory ideas about AI in medicine, such as some believe that AI will replace human doctors

---

[1]https://futurism.com/ai-medicine-doctor
[2]https://news.harvard.edu/gazette/story/2020/11/
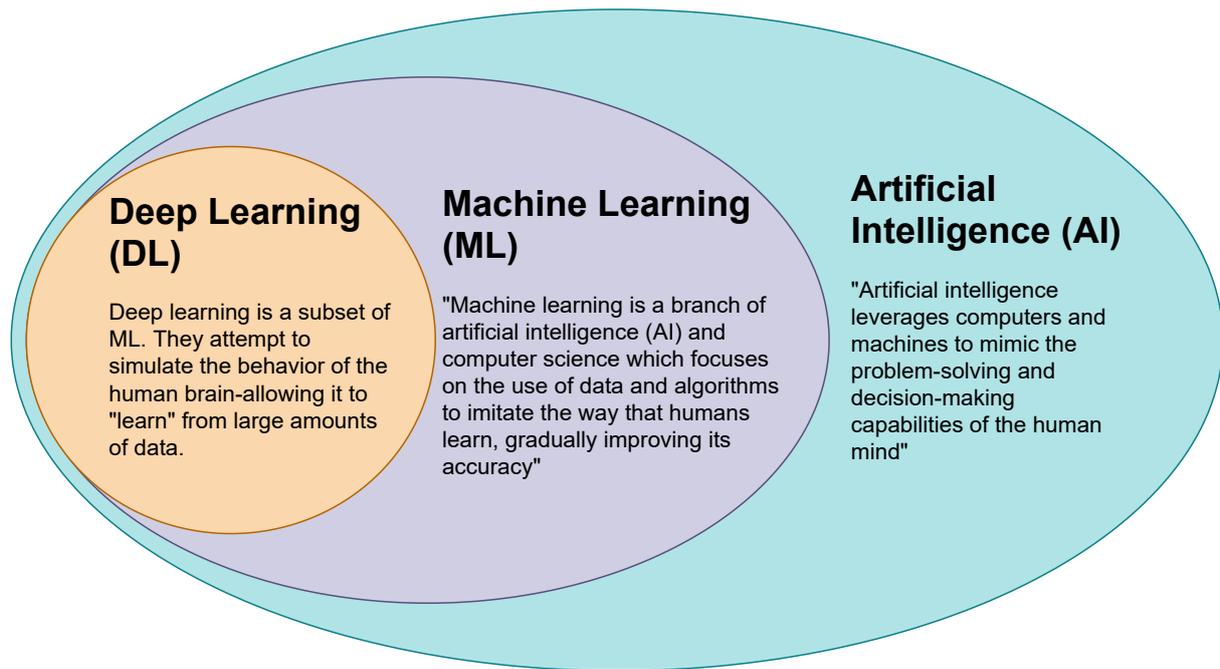
Figure 1.1: Definitions [16] and relations between AI, ML and DL.

and others believe that AI will "just" become a supportive tool for human doctors. Nevertheless, it seems like many believe that AI will become more popular in the coming years. Thus, applying AI in medicine is important because it may directly affect humans' personal lives, and successful medical systems directly correlate with life expectancy and quality. Therefore, producing AI systems with reliability and integrity is important in the medical domain. To understand applying AI in medicine for developing computer-aided diagnosis (CAD) systems, we should understand the complete medical AI pipeline. A simplified version of this pipeline is depicted in Figure 1.3 with four steps: (I) collecting data, (II) annotating data using experts, (III) applying ML methods, and (IV) final product and explainable artificial intelligence (XAI). These four steps are discussed further in the next section.

1.1   Background and Motivation

AI-based solutions are used in the medical domain for different purposes, such as to develop treatment protocols, drugs, personalized medicine, patient monitoring systems, robotics, and diagnosis processes [11]. Among these, AI-based diagnosis processes or CAD systems [17] got more attention from AI researchers. CAD systems aid doctors as the "second opinion" to finalize decisions.

"Getting diversity in the training of these algorithms is going to be incredibly important, otherwise we will be in some sense pouring concrete over whatever current distortions exist."

– Isaac Kohane, head of Harvard Medical School's Department of Biomedical Informatics

"You're not expecting this AI doctor that's going to cure all ills but rather AI that provides support so better decisions can be made."

– Finale Doshi-Velez, John L. Loeb Associate Professor of Engineering and Applied Sciences at the Harvard John A. Paulson School of Engineering and Applied Sciences

Futurism

FUTURISM | 1. 31. 18 *by* ABBY NORMAN

Your Future Doctor May Not be Human. This Is the Rise of AI in Medicine.

From mental health apps to robot surgeons, artificial intelligence is already changing the practice of medicine.

Figure 1.2: Some quotes and headings about AI and medicine in news articles

In this regard, we started to research ML-based solutions for CAD systems by following the above four steps pipeline to help medical experts more correctly and efficiently detect anomalies in medical data from real examinations to save lives ultimately. The goals were to both address large miss-rates [18, 19, 20] and observer variations [21, 22]. The process of researching and developing ML solutions is presented using Step III (Figure 1.3). However, we soon realized a considerable lack of medical data to develop good ML models in the domain for various reasons, increasing the importance of the first two steps in Figure 1.3. Therefore, we have studied how datasets should be collected, composed, and published
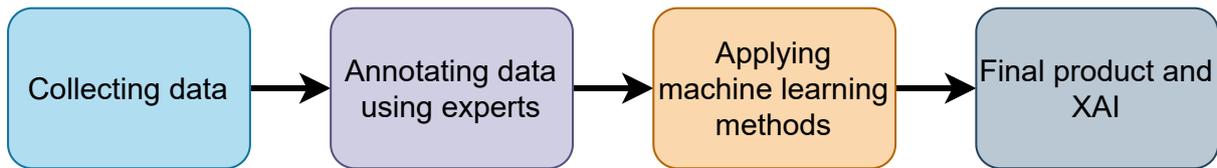
Figure 1.3: The main four steps of applying ML solution in the medical domain.

as open datasets. Within the three years of Ph.D. time, a total of seven datasets [23, 24, 25, 26, 27, 28, 29] were successfully collected and published. Medical experts have labeled or annotated data (Step II) in these datasets, but not all the datasets because the annotation process is costly and time-consuming. For example, our gastrointestinal (GI)-tract dataset [23] has labeled images and pixel-wise annotated polyp images performed by experienced colonoscopists. However, the biggest part of the GI-tract dataset is still unlabelled data because of the costly and time-consuming data annotation process. We analyzed three branches in medicine, *gastroenterology*, *andrology*, and *cardiology* in parallel to the data collection process. The main motivation for choosing different domains was to show that our methods can work on different problems (are generalizable) and to produce ML-based CAD solutions to help experts by providing more efficient and accurate automated assistance for their tasks.

The gastroenterology branch investigated classification models [30, 31, 32, 33, 34] to classify GI-tract findings and segmentation models [35, 36] to segment polyp regions. When producing these ML solutions, we identified that generalizability is one of the main issues for classification and segmentation due to the lack of labeled and annotated data to train ML models. The classification models introduced in our studies [30, 31] showed good performance when the validation and testing data are a subset of the same dataset used to prepare the training dataset. However, the performance of the best models showed poor performance for completely new datasets collected from different hospitals. The problem was caused as a result of the over-fitting [37]. In addition to the data bias problem, we also identified that an imbalanced number of images of different classes makes less accurate ML models. Detailed discussion on this issue can be found in [31], where we analyzed and experimented with different datasets. Similar to the classification models, polyp segmentation models show poor performance due to small datasets to train segmentation models. We tried to solve the problem by introducing a novel data augmentation method

called PYRA[3] [36] and a novel segmentation model called DivergentNets [35]. However, we had only small datasets to train segmentation models compared to the training datasets used in classification models. Due to the time-consuming and costly pixel-wise image annotation process, researchers or data providers usually provide only small segmentation datasets for medical image segmentation tasks. The medical image annotation process is more challenging than the general image annotation process because experts of the specific medical domain should perform these manual segmentations or review them. These experts are often rare or do not have much time.

In addition to providing ML solutions in gastroenterology, we have investigated ML solutions [38, 39, 40] to predict motility and morphology level of sperm samples which are videos recorded using microscopic analysis. These research works are considered under the andrology branch. The proposed models show acceptable performance, but those performance values were insufficient to use the solution practically. By researching ML solutions to predict motility and morphology levels of sperm samples, we identified a possibility of improving our models if we could prepare pixel-wise annotated datasets to perform segmentation before predicting morphology and motility levels. However, performing pixel-wise annotations for a sperm-like medical dataset is a complicated problem for experts because of having hundreds of sperms in a single frame of the dataset. A possible solution is annotating sperms using an unsupervised way and processing those annotated sperm samples to find motility and morphology levels.

In cardiology, we built an electrocardiogram (ECG) analysis system [41] using ML models to predict the properties of ECGs. This experiment used a big ECG dataset to train the ML models and showed that the ML models could outperform experts' analyses. Unfortunately, the dataset used to train our models is a private dataset, and publishing them to reproduce our solutions is not possible due to privacy concerns. In this context, we noticed that there should be a way for omitting privacy concerns. In this ECG study, we have presented an explainable AI mechanism called Grad-CAM [42] to find the most important regions for DNNs to predict the properties of ECGs. However, we could use only the explainable methods that do not expose the real dataset to the public because of privacy concerns. Suppose we have a method to omit and work around the privacy concerns. In that case, we can use any explainable method which uses the real dataset,

---

[3]`https://vlbthambawita.github.io/PYRA/`

for example, to explain using examples [43].

The success of AI solutions in medicine is highly dependent on the data to train the AI algorithms. However, collecting and sharing medical data is harder than other general data because of the privacy restrictions attached to the medical data. The collection of medical data (Step I) is presented using the first box in Figure 1.3. If the training data cannot provide useful information to AI algorithms, the algorithms become less accurate and generalizable. Therefore, medical data is essential for developing successful AI solutions. However, medical data collection and preparation are not straightforward. The unrolled cumbersome internal process of Step I is presented in the first seven steps depicted in Figure 1.4, as discussed by Willemink et al. [44]. However, following these steps is a complex task because of privacy concerns such as ethical approval and data de-identification process, in addition to the data preparation process. Medical data need post prepossessing because the raw medical data producing from medical instruments are not designed for sharing. Many research discusses the protection of digital data in a learning health system [45], the privacy of big medical data [46, 47, 48], and balancing health data access and privacy [49]. These research discussions show the importance of considering privacy rules and regulations with health data. As a result, the privacy restrictions applied with the medical data make the process in Step I harder and slow down the whole pipeline depicted in Figure 1.3.

The rules and regulations for producing open access medical data vary from country to country and region to region according to data protection regulations introduced in the specific regions. For example, Norway should follow the rules given by the Norwegian data protection authority (NDPA) [50] and enforce the personal data act [51] in addition to following general data protection regulation (GDPR) [52], which is the common guideline for European countries. While there is no central level privacy protection guideline in the US like GDPR in Europe, rules and regulations in the US are coming through other US privacy laws, such as Health Insurance Portability and Accountability Act (HIPAA) [53] and California Consumer Privacy Act (CCPA) [54]. In Asian counties, they follow their own rules country-wise, such as Japan's Act on Protection of Personal Information [55], South Korea's Personal Information Protection Commission [56], and the Personal Data Protection Bill in India [57]. If researchers can perform research with these privacy restrictions, the papers published are often theoretical methods only. As a consequence, the
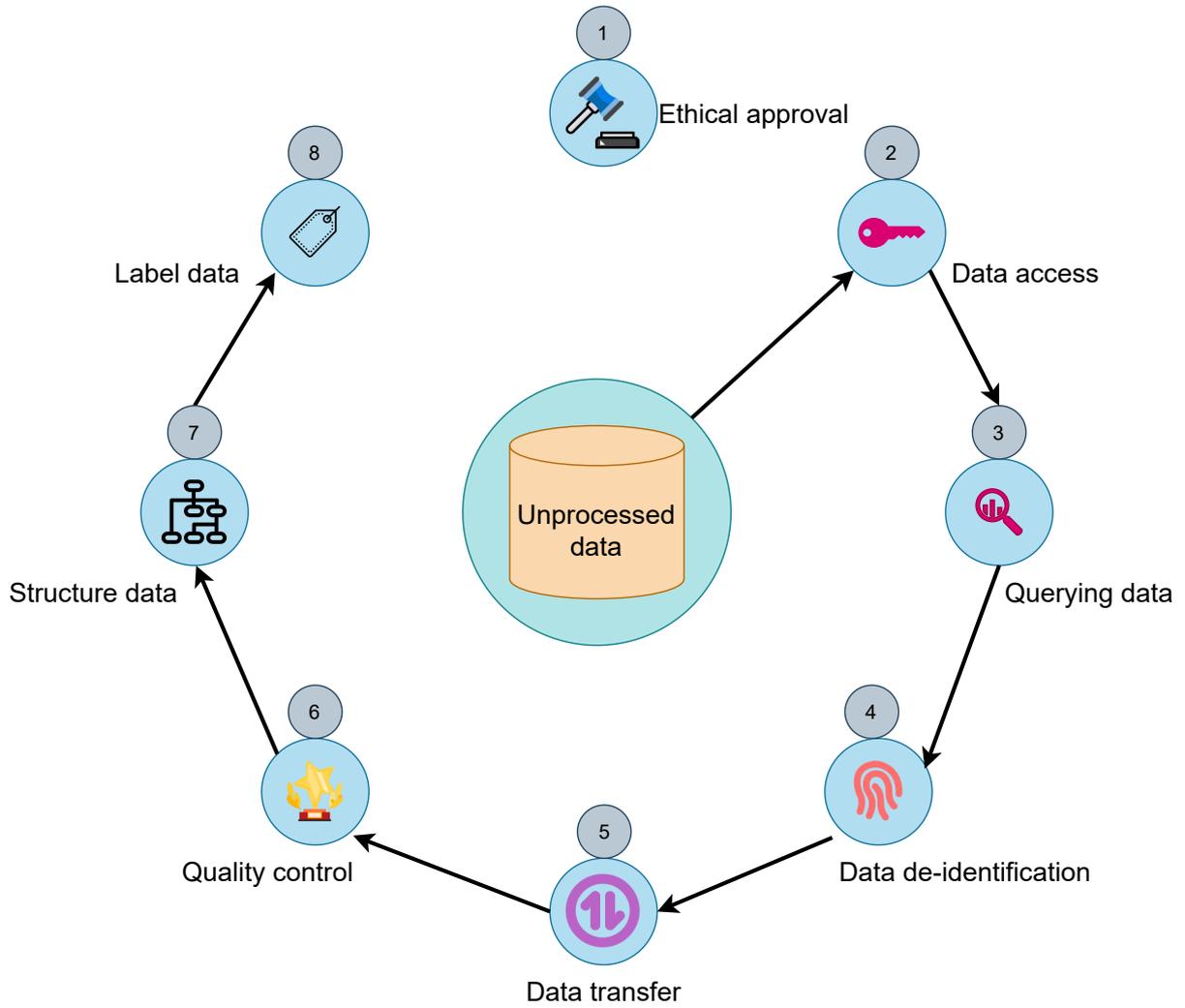
Figure 1.4: Medical data preparation process as discussed in [44]

results of those studies are not reproducible, and fair and correct comparisons between methods are hard to achieve. All these consequences are due to a lack of available data and sharing restrictions. Furthermore, universities or other research institutes that use medical domain data for teaching purposes use the same medical domain datasets for years, which affects the quality of education. Therefore, data sharing restrictions resulting from privacy protocols are identified as one of the main problems and obstacles, and we have researched to address this challenge in this thesis.

In addition to the privacy concerns, the cost of medical domain experts for extracting useful information from medical data is another obstacle to producing big datasets, which are helpful for AI. This stage is presented as the second box in Figure 1.3 and task number 8 in Figure 1.4. For example, to train the most common supervised ML techniques, ground truth data are needed. In other words, annotated datasets are essential. Because of this necessity of annotated data, new companies and job opportunities are opened to perform data annotations for datasets used to train AI algorithms [58, 59]. For example, the pricing list in Google for annotating datasets is presented in Table 1.1. However, medical data annotation (or producing ground truth) is not easy as making ground truth for general datasets. Medical data annotation is more challenging than other general data annotations because only the experts in the medical domain can perform the annotations fully trustable in terms of correctness. If the data annotation by experts is impossible, the experts should do at least a review process to make the annotations trustable before using them in AI algorithms. The importance of having accurate annotations from experts for medical data is, for example, discussed by Yu et al. [60] using a mandible segmentation dataset of CT images. Because only the medical experts can accurately do the medical data annotation process, the expert annotation process becomes expensive. Additionally, this annotation process takes considerable time to produce ground truth data precisely [44], consuming time that clinicians usually rather spend on treating patients.

The third step in Figure 1.3 represents applying ML methods after collecting medical data and annotating the data using domain experts. However, due to privacy protocols and the aforementioned complex data retrieval and annotation problems, researchers and industry, who apply ML solutions for medical data, do not have access to open-access expert-annotated datasets. Because of this limited data problem, the models become less reliable [31] (due to poor generalizability) and have fewer functionalities such as limited

Table 1.1: Google labelling cost (to date: 05-05-2021). [61]

Data type	Objective	Unit	Tier 1	Tier 2
Image	Classification	Image	$35	$25
	Bounding box	Bounding box	$63	$49
	Segmentation	Segment	$870	$850
	Rotated box	Bounding box	$86	$60
	Polygon/polyline	Polygon/Polyline	$257	$180
Video	Classification	5sec video	$86	$60
	Object tracking	Bounding box	$86	$60
	Event	Event in 30sec video	$214	$150
Text	Classification	50 words	$129	$90
	Entity extraction	Entity	$86	$60

interpretability [62]. These limitations and our own experience of developing ML models for CAD systems emphasize the requirement of having an alternative fast track to getting medical data into the third step (Step III) of applying ML.

The fourth step in Figure 1.3 represents the final stage of producing products using ML to use in clinical settings. In this stage, explaining the prediction results (XAI) is an important step because it is the only step in which one can convince doctors to accept decisions made by ML solutions. Explanation by example is currently a preferred XAI method by non-experts [63]. Privacy issues can limit these XAI functionalities, such as explaining DL solutions by examples [64] when the example data is restricted to publish.

In summary, the problems related to collecting and processing medical data can be identified as a major bottleneck to produce enough open-access medical data for developing well-performing ML solutions for CAD systems. The privacy concerns with the medical data and the costly and time-consuming medical data annotation process are two reasons for the data deficiency problem. In addition, we identified that a lack of true-positive data compared to true-negative data in the medical domain, giving large class imbalances, is a problem for producing AI-based systems. In this regard, this thesis focus on producing well-performing ML models for CAD systems after finding a way to tackle the data deficiency problem by generating synthetic data using a new concept and the framework named DeepSynthBody.

1.2 Research Question and Objectives

The main overall goal of our research is to investigate and develop accurate, generalizable, and well-performing ML models for CAD systems for biomedical applications assisting doctors in clinical practice. In this thesis, we have a particular focus on the problems and challenges coming from medical data. These challenges of collecting and processing medical data, identifying that the lack of medical data due to, for example, privacy issues, resource-consuming data annotation processes, and data imbalance problems are major obstacles for AI-based medical technology research and development. Therefore, we focus on researching a way to address the data deficiency problem in the medical domain while researching and developing well-performing and generalizable ML models for CAD systems for selected three domains as case studies. The overall research question for this study therefore is:

> **What are the problems that emerge from data in computer-aided diagnosis systems, and how can these problems be tackled?**

After identifying the research question, we have defined the objectives of this thesis as follows:

- **Main objective**: Research and develop ML models which are the main component of CAD systems for different medical applications, focusing on the problems of limited availability of biomedical data.

- **Sub-objective I**: Research and develop ML models for CAD systems to assist doctors.

- **Sub-objective II**: Collect, research and develop datasets to develop ML models for CAD systems for biomedical applications.

- **Sub-objective III**: Research and develop benchmark analysis with the medical datasets to identify the problems for producing well-performing ML solutions in the medical domain.

- **Sub-objective IV**: Research and develop deep generative adversarial networks (GANs) that can produce synthetic data to address the data deficiency problem, the major obstacle for developing medical AI-based solutions.

This thesis has used three different medical case studies for Sub-objective I, Sub-objective III, and Sub-objective IV. The medical fields chosen are *cardiology*, *gastroenterology*, and *andrology*. We chose these three domains since they are diverse from each other in terms of data. In Sub-objective II, we have introduced additional datasets in addition to the main three case studies as its main goal is collecting and developing medical datasets.

1.3 Scope and Limitations

This research was started to developing well-performing and generalizable ML models for CAD systems to assist doctors. However, the early identification that the medical data is a major obstacle for developing ML models, solving the data deficiency problem in the medical domain became another objective of this thesis. Therefore, in this thesis, two major development streams can be seen. One is developing ML models for CAD systems, and one is researching and developing GANs to overcome the data deficiency problem. As the main finding of this thesis, we could introduce a novel concept and the framework based on GANs to tackle the data deficiency problem. The framework has been demonstrated with a few selected case studies as a proof of concept. However, the novel concept and the framework are not limited to the presented case studies. All other possible research areas using our concept and framework are discussed in the future work section.

In this thesis, three types of datasets were used. In particular, we have used ECG signals, GI images, and a sperm video dataset as case studies that cover three different medicine branches: gastroenterology, andrology, and cardiology. These three datasets were selected because they were the initial studies used to develop ML models for CAD systems. Additionally, the same datasets were used as proof of concept to demonstrate the potentials of the new concept and the framework introduced as a solution to the data deficiency problem in the medical domain. It is worth mentioning that the new concept is also developed as a big open-source project planning to have contributions worldwide. Therefore, all the case studies and experiments were performed just to prove the new concept. The ECG dataset covers biomedical signal data in the selected case studies, while the GI image datasets cover biomedical images. The sperm dataset is

related to medical video data as well as medical images. In addition to time restriction, the scope of this study is limited to selected data formats such as one-dimensional (1-D), two-dimensional (2-D), and three-dimensional (3-D) because of limited access to other types of medical data such as magnetic resonance imaging (MRI), which are considered four-dimensional (4-D) with a temporal dimension.

The proposed concept consists of a four-step pipeline. These are collecting real data and analysis, developing generative models, generating synthetic data, and explainable DeepSynth AI and DeepSynth Explainable AI. While the thesis covers the first three, the most important steps, data handling, applying GANs, and producing synthetic data via the end functionalities, the last step of researching explainability is not investigated due to time limitations and is regarded as an important future research direction. Additionally, we have published an online platform for the concept. This online platform will be changed in the future as a result of improvements over time.

1.4 Research Methodology

In computer science, it is harder to practice traditional research methodology followed by classic sciences as described by Dodig-Crnkovic [65] because computer science can be identified as a combination of various scientific disciplines. In sciences, we can identify three paradigms, theory, abstraction, and design [66]. Generally, the theory is for mathematical sciences. The abstraction or modeling is for natural sciences. The design or experimentation is for engineering. However, it is not easy to explicitly map computer science for one of these three paradigms. While these three are inseparable from computer science, they are distinct from each other. Therefore, we define this thesis work in each of the above paradigms as follows.

- **Theory:** Major elements of the theory of the concept introduced in this thesis consist of the major theories related to AI presented in the report [66] produced by ACM and IEEE task force. This report has introduced four steps to developing a coherent, valid theory in any science. They are:

  1. Characterize objects of study (definition).

  2. Hypothesize possible relationships among them (theorem).

3. Determine whether the relationships are true (proof).

4. Interpret results.

In this regard, we have introduced our main objective and four sub-objectives to research ML models for CAD systems in the medical domain and a novel concept to overcome the data deficiency problem. We hypothesize that generative models can generate synthetic data to overcome the data deficiency problem of developing ML models in the medical domain. Using three different case studies, we have presented the performance of our ML models. Moreover, using the same case studies, we proved how to use GAN-generated synthetic data to solve the data obstacles in the medical domain.

- **Abstraction (modeling):** is defined based on the experimental scientific methods. In the ACM report, they have described four stages for investigations of phenomenons such as:

  1. Form a hypothesis.

  2. Conduct a model and make a prediction.

  3. Design an experiment and collect data.

  4. Analyze results.

  According to this modeling paradigm, deep generative models can be identified as the main component of modeling our hypothesis. Under different medical data formats, we analyzed generative models and collected synthetic data. To find the best generative models for generating synthetic data, we have studied them qualitatively and quantitatively using experimental prototypes. Not only deep generative models, but we have also experimented with baseline experiments and benchmark experiments, which were performed to develop experimental prototype ML models for CAD systems.

- **Design:** In this paradigm, four stages can also be identified to build a system to solve a specific problem. They are

  1. State requirements.

  2. State specifications.

3. Design and implements the system.

4. Test the system.

The medical data was identified as a key requirement to research and design well-performing ML models for CAD systems. Therefore, we collected real medical datasets and developed synthetic medical datasets. Then, we designed ML models using the real medical datasets and the synthetic medical datasets. Moreover, a complete framework to generate synthetic data in the medical domain was introduced and implemented. We have tested our ML models and GANs introduced in the framework using three different case studies.

1.5   Contributions

The research in this thesis contributes to medical AI technology aimed to assist clinicians in their daily work, improving the quality of the health care systems. We started to research and develop ML models for CAD systems using small existing datasets and collecting our medical datasets, where the developed models performed very well. However, the major challenge identified was the data deficiency problem, where dataset development was cumbersome due to various reasons. This challenge then becomes the major challenge addressed in this thesis while still developing ML models.

In particular, in this thesis, four sub-objectives were introduced to accomplish the main objective, which aims to develop ML models for CAD systems to assist doctors in improving the efficiency of diagnosis. These four sub-objectives were initiated to develop well-performing ML models and solve the data deficiency problem of the current applied machine learning pipeline used in the medical domain, as depicted in Figure 1.3. We started researching and developing ML models for CAD systems to achieve Sub-objective I. Then, in Sub-objective II, collecting data was initiated after finding that data is an important factor for achieving Sub-objective I. Then, the performing benchmark experiments are mainly used to achieve Sub-objective III to study the medical datasets to understand the related problems to research and address in Sub-objective IV. Sub-objective IV was achieved by experimenting and investigating GANs to generate synthetic data to overcome the data deficiency problem in the medical domain. Figure 1.5 shows all the contributions via these four sub-objectives and the main objective. Some contributions
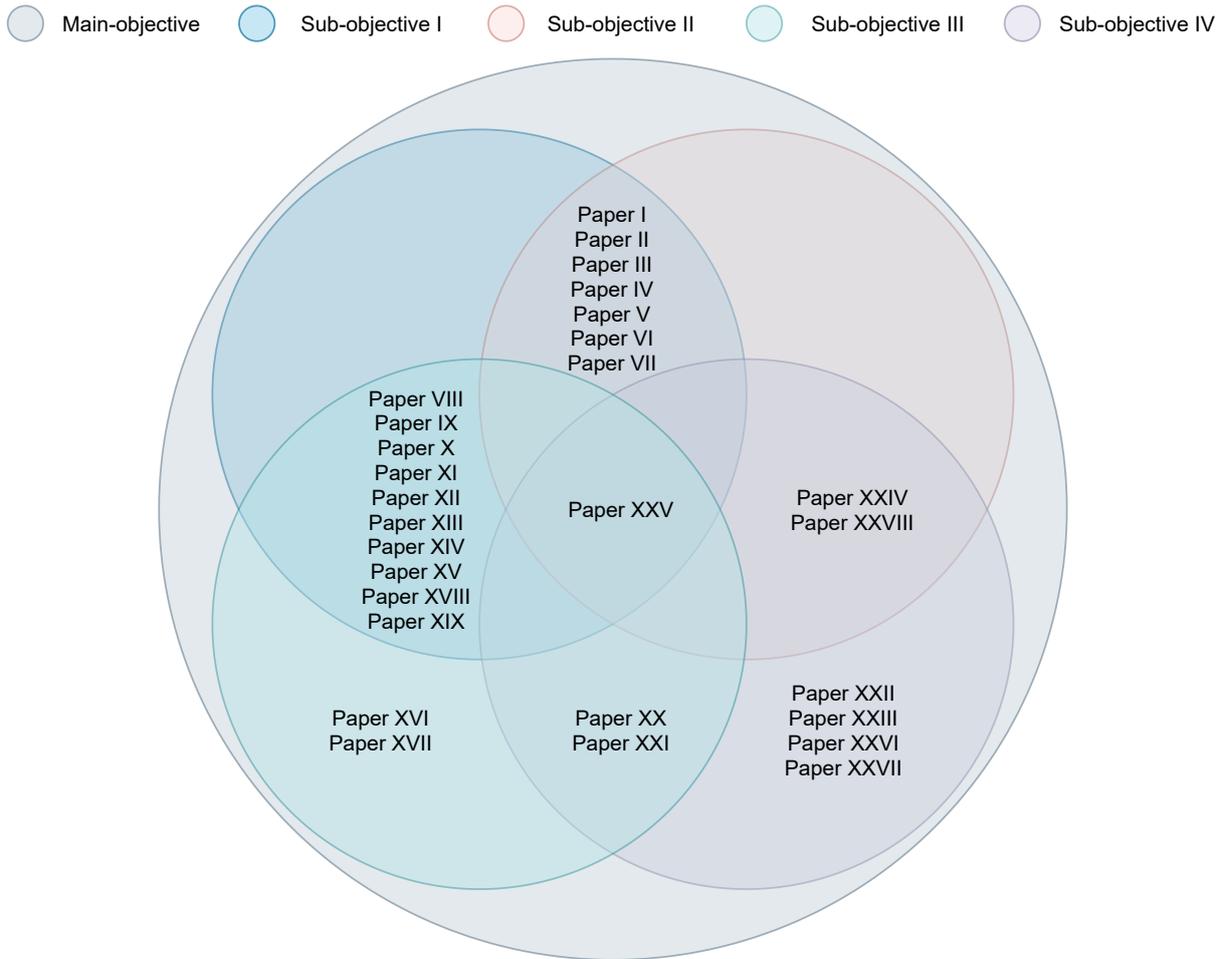
Figure 1.5: Paper-wise contribution to all objectives.

can be identified through two or more objectives, while all the contributions are directly attached to achieve the main objective.

The following bullet points show all contributions to sub-objectives and the main objective. These contributions include dataset papers, ML-based CAD models or benchmark papers, and GAN-related papers can be found. The dataset paper, HyperKvasir [23][4], got much attention from the research community within a short period because of the richness of data diversity. Not only that, the results of most benchmark papers were within the top 5%. For example, we won 1st place for the EndoCV grand challenge[5] 2021. Similarly, GAN-based experiments also became popular within a short period in the research community because of the competitiveness of the presented qualitative and quantitative results of novel methods used to generate synthetic data. For example, The DeepFake ECG paper was read by many people within a few days after publishing the

---

[4] https://www.nature.com/articles/s41597-020-00622-y/metrics
[5] https://endocv2021.grand-challenge.org/

pre-print, and it became a part of news heading about recent developments of interests in cardiovascular medicine[6]. The following section discusses all the contributions toward the objectives of this thesis. The main objective is discussed at the end of the following list to emphasize how sub-objectives contribute to accomplishing the main objective.

- **Sub-objective I**: The main focus of this sub-objective is to research and develop well-performing ML models for CAD systems to assist doctors. As case studies, we have selected three branches of medicine. These are cardiology, gastroenterology, and andrology. In gastroenterology, images collected from colonoscopies were the main data stream to apply ML algorithms which are the core algorithms in CAD systems. In different timeline stages, several classification models [30, 31] and segmentation models [35, 36] were researched and implemented for the gastroenterology branch under this thesis. In addition to real data, we used synthetic data with segmentation models [67] to predict polyps in GI-tract data. Similarly, ML-based regression models were investigated and developed for the andrology branch [38, 39, 40, 68]. For the cardiology branch, an ML-based ECG analysis system [41] was researched and implemented. Moreover, all the dataset papers [23, 24, 25, 26, 27, 28, 29] introduced ML models as baseline experiments which can be considered initial models for developing CAD systems.

- **Sub-objective II**: The main task of this sub-objective is to collect and produce medical datasets, which is identified as the main bottleneck for developing ML-based CAD systems. Moreover, these datasets are the main assets for initiating the novel concept and the corresponding framework, DeepSynthBody, introduced in this thesis. Different types of real medical datasets [23, 24, 25, 26, 27, 28, 29] were collected and published to the research community with the baseline experiments under this thesis to accomplish the sub-objective I. All the datasets contribute to designing ML models for CAD systems (sub-objective I) because of the baseline experiments introduced in every dataset paper.

  In addition to our datasets, two additional datasets were used from outside of the dataset contributions. One is an ECG dataset, which is a private medical signal dataset. The second one is a sperm dataset [69] which represents sperm video data.

---

[6]https://www.medpagetoday.com/cardiology/

The additional datasets were selected to design ML models for CAD systems in completely two different branches: cardiology and andrology. At the end of the thesis, we showed using synthetic datasets to overcome the data deficiency problem. These synthetic datasets consist of a synthetic ECG dataset [70], a synthetic GI-tract landmark dataset [71] and, a synthetic polyp dataset [67] generated using the GAN models introduced as a result of our new concept and the corresponding framework.

- **Sub-objective III**: Initially, we focused on designing generalizable ML models, which are the core of CAD systems to achieve Sub-objective I. Later, we identified that the medical data deficiency in training ML models should be tackled. We have performed benchmark analyses with selected three medical datasets to investigate the data-related problems and investigate them. As a result, a set of benchmark articles for the selected datasets as case studies were published to achieve the benchmark analysis objective (Sub-objective III). These benchmark analyses helped to identify the problems of designing ML models. Additionally, these benchmark experiments give preliminary knowledge about medical datasets, which we will use to generate synthetic data to achieve Sub-objective IV. Different types of quality control benchmark analysis with the GI-tract data [32, 33] were performed to support this objective. Moreover, we can consider the ML models [30, 31, 36, 35] introduced in Sub-objective I as benchmark analysis studies for Sub-objective III because they are correlated with each other. Similarly, the ECG analysis [41] and sperm analyses [38, 39, 40, 68] experiments were considered benchmark analyses to identify data-related problems to address using synthetic data. Without having benchmark analysis or baseline experiments, it is not recommended to researching GANs for the new framework because data problems related to a medical dataset cannot be identified without benchmark analysis. We have also performed benchmark analysis with synthetic data [67, 72, 73] to identify the usability of synthetic data instead of real medical data.

- **Sub-objective IV**: Research and developing GANs is the core of the DeepSynth-Body concept [71] (`www.deepsynthbody.org`) proposed as a solution to the data deficiency problem identified and investigated in this thesis. We started investigating possibilities of using GANs with GI-tract data such as prepossessing GI tract

images using a GAN [72, 73] to fill blank regions and to predict blurry pill cam video frames using a GAN [74], which can predict the future frames for given input frames to solve the data problems of developing ML models. These experiments gave the basic understanding of how GANs use in the medical domain and how hard it of producing synthetic data in the medical domain. Then, an advanced GAN experiment, namely Pulse2pulse [70], which can generate synthetic 12-leads 10-seconds ECG indistinguishable from real ECGs was introduced to overcome the data sharing problem as a result of privacy issues. Ultimately, we proved that our synthetic ECG dataset shows very close characteristics to the real data distribution [70].

Moreover, to address the costly and time-consuming expert's data annotation process, we experimented and introduced novel pipelines [75] of GAN architectures using GI-tract dataset to generate synthetic polyp data from the clean colon to overcome data imbalance problems in the medical domain, such as having more true-negative samples compared to true positive samples. Furthermore, we researched and presented a new pipeline to generate synthetic polyp data with the corresponding mask from a single polyp image [67], namely SinGAN-Seg, and showed that generated synthetic medical data is a solution to overcome data problems in the medical domain. Additionally, we investigated the usability of GANs to produce synthetic sperm data [76] instead of blurry-looking sperm video samples to have a better quality data stream for training AI-based sperm analysis systems in the future. To get active contributions of performing GAN-related research to produce synthetic data from non-computer science people, we have proposed a tool [77] to run GAN experiments without writing a single line of code.

- **Main-objective**: The final objective was to connect these all together and produce well-performing and more accurate ML models for CAD systems to assist doctors for efficient diagnosis by addressing the data deficiency problem. The initial ML models designed to achieve the Sub-objective I showed the effects of the data deficiency problem in the medical domain. Then, we collected, researched, and developed datasets (real and synthetic) to develop ML models for biomedical applications. In Sub-objective III, benchmark analyses were performed to identify the data problem to be addressed. We proposed the new concept and the corresponding framework, DeepSynthBody, based on GANs as a solution to the data deficiency problem in the

medical domain (Sub-objective IV). Finally, we published our solution as an open-source project for getting more collaborations worldwide at `www.deepsynthbody.org`.

As described above, our research addresses the stated objectives. Then, regarding the overall research question, what problems emerge from data in computer-aided diagnosis systems, and how can these problems be tackled? We first identified the problems and proposed the DeepSynthBody concept to tackle them. As the problems, we could identify that data to train ML models in the medical domain is lacking due to several data preparation problems, such as privacy concerns and the costly and time-consuming data annotation process. Then, this data deficiency problem causes generalisability issues and performance issues for ML models, which are the core algorithms used in CAD systems. To answer the data deficiency problem, we have experimented and developed synthetic data and showed that generated synthetic data could solve the data deficiency problem in the medical domain because synthetic data can address some of the restrictions emerging from privacy issues coming with sensitive data. We also show that synthetic data is an alternative way to prepare data and corresponding segmentation masks for the costly and time-consuming real data annotation process.

In addition to the main contributions aligning to this thesis work, the author contributed as a development team member of the Norwegian "Smittestopp" app, which was developed to trace Covid-19 contacts. Algorithms to find contacted regions of interest using GPS coordinates were investigated under this Covid-19 app development project. Moreover, several master students were supervised, and they successfully completed their master's degrees with good grades and publications [24, 72, 73, 74], which were great contributions to the GAN development stage of DeepSynthBody. Not only these, the author contributed to a research study [78], which was focused on detecting soccer events from video clips, but this study is out of the scope of the thesis.

1.6 Outline

Our initial contributions were focused on designing ML models for CAD systems to aid doctors by achieving the Sub-objectives I and II. However, the data-related problems of the current pipeline of applying ML motivated us to find a new way to overcome the data

Chapter 1. Introduction

deficiency problem in the medical domain. Therefore, this thesis mainly focuses on designing a novel concept, DeepSynthBody, and the corresponding framework introduced to bypass the data-related problems such as privacy-related problems with medical data and resource-consuming medical data annotation process. To discuss, research, and present the DeepSynthBody concept, we organized the thesis as follows:

- Chapter 2: Related Work - gives more required background knowledge to follow this thesis. In this chapter, the basic knowledge about ML concepts and corresponding references used in designing CAD systems are given. Then, deep generative models and the state-of-the-art GANs are discussed with greater details to give enough knowledge to understand the new concept introduced in this thesis. Additionally, similar frameworks to DeepSynthBody and other studies about synthetic medical data generations are discussed.

- Chapter 3: DeepSynthBody - In this chapter, the DeepSynthBody concept, which is the new concept introduced in this thesis to overcome the data deficiency problem, is formalized by developing the corresponding framework. The theoretical behavior of the framework is discussed in this chapter with four main sections, which are collecting real data and analysis, developing GANs, producing DeepSynth data, and explainable DeepSynth AI and DeepSynth explainable AI of this framework. The first three sections are explained using three case studies of ECG data, GI-tract data, and sperm data. These use cases are discussed with the significant findings, which were identified as the most influenced results for the success of DeepSynthBody.

  Under the collecting of real data and analysis, data collection procedures and analysis procedures are discussed. Then, the core of this framework, GAN development, is discussed in developing GANs. In the same section, a novel tool, namely GANEx, used to performing GAN experiments, is introduced. The process of producing Python package index (PyPI) packages is explained using the use case studies in the same section. The website `www.deepsynthbody.org`, which is the online platform of this concept, is introduced in the third section. Finally, the optional step, explainable DeepSynth AI and DeepSynth explainable AI, are discussed theoretically.

- Chapter 4: Discussion and Conclusion - discusses limitations, other advanced func-

tionalities, which can be researched with DeepSynthBody as future directions, and the conclusion about how the DeepSynthBody concept and its formal DeepSynthBody framework help to overcome the data deficiency problem related to the development process of ML models for CAD systems.

- Appendix A: All the papers counted as contributed under this thesis are listed here with the publication details and corresponding contribution statements.

Chapter 2

Related Work

This chapter covers the basic concepts of this thesis and a literature review to discuss similar research directions and their limitations. We give appropriate knowledge to understand the development of ML models forCAD systems with limited medical data. The first section provides an overview of medical datasets. Then, the common ML solutions used in medicine are discussed with the corresponding evaluation criteria because they are the basics for developing CAD systems. Afterward, GANs are introduced with their theoretical background because GAN is the basic model used to generate synthetic data to overcome the data deficiency problem, which is identified as a major problem in the medical domain. Finally, a review and discussion about previous studies, which use a similar direction to DeepSynthBody to address the lack of medical data, is provided.

2.1 Medical Data

Data is the most important factor for developing AI solutions [79, 80, 81], and it cannot be separated from the field of AI. In this regard, medical datasets are the key to develop successful ML solutions in the medical domain for CAD systems. Therefore, AI researchers try to collect as much as possible medical data from data providers such as hospitals or medical research institutions. As a result, many public repositories are available for medical data, and a few of them are shown in Table 2.1. As we can see in the table, some medical repositories have a specific type of medical data like NITRC, while some collect all types of data, such as the UC Irvine machine learning repository. However, most datasets in these repositories are smaller than general datasets such as Imagenet [82]

Table 2.1: Sample data repositories with various medical data. Some of the data repositories have specific type of data. Some of them have data collections from multiple domains including the medical domain.

Repository	Link to access	Description
The cancer imaging archive (TCIA)	`https://www.cancerimagingarchive.net/`	A large archive of medical images of cancers.
NeuroMorpho	`NeuroMorpho.Org`	Digitally reconstructed neurons from vaiours animal types. Human is included as one type.
NeuroImaging Tools and Resource Collaboratory (NITRC)	`https://www.nitrc.org/`	Neuroinformatics data, from MR, PET/SPECT, CT, EEG/MEG, optical imaging, clinical neuroimaging.
OpenNEURO	`https://openneuro.org/`	Sharing MRI, MEG, EEG, iEEG, ECoG, and ASL data.
PhysioNet	`https://physionet.org/`	A repository for Physiologic Signals.
OSF.io	`https://osf.io/`	Open datasets from all the domains including the medical domain.
The UC Irvine Machine Learning Repository	`https://archive.ics.uci.edu`	Open access datasets from many domains including the medical domain.
Registry of Open Data on AWS	`https://registry.opendata.aws/`	Open access datasets from many domains including the medical domain.
IEEE DataPort	`https://ieee-dataport.org/`	Datasets from different domains around 25 categories defined by IEEE DataPort such as Biomedical and Health Sciences , Biophysiological Signals, Environmental and more other general categories including health data.

because, for example, collecting medical datasets should follow specific protocols to avoid privacy restrictions, and annotating medical data is costly and time-consuming.

Medical data have different formats, which vary from a simple single value to advanced multi-dimensional data types such as two-dimensional (2-D), three-dimensional (3-D), and four-dimensional (4-D). Multi-dimensional data has more than one value to represent a single data point. Visual representations of sample biomedical data with various data formats are depicted in Figure 2.1. Figure 2.1(A) represents a simple 1-D ECG signal, and Figure 2.1(B) shows an image (2-D) taken from an endoscopy. Some medical data
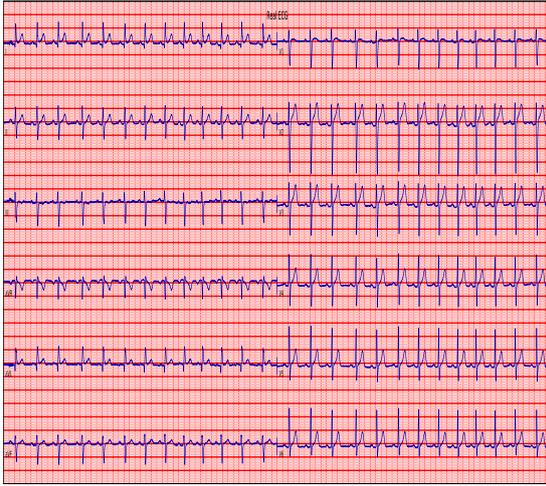
(A) - An ECG signal [41]
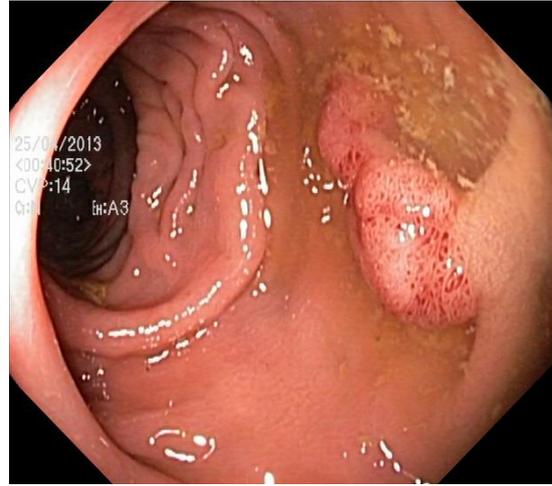
(B) - An endoscopy image [23]
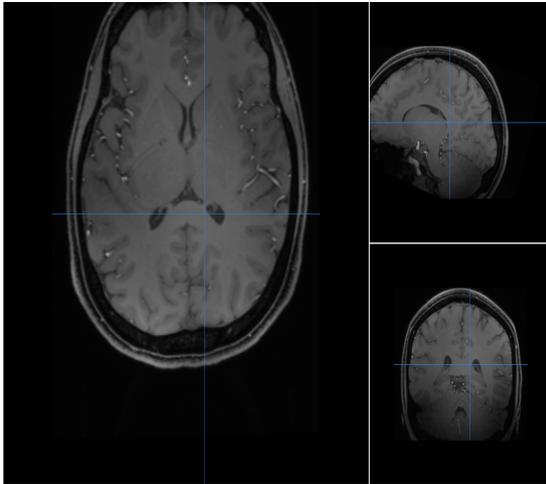
(C) - An MRI representation [83]
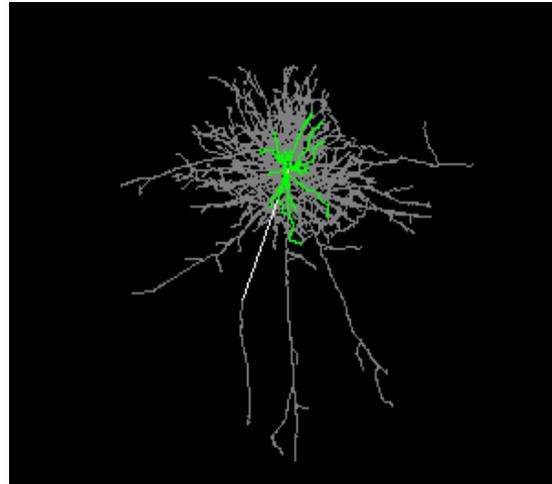
(D) - A digitally reconstructed neuron [84]

Figure 2.1: Visual representations of different types of biomedical data.

cannot be simply presented in a 2-D plane and need software tools to get actual 3-D visualizations such as an MRI as depicted in Figure 2.1(C), and a digitally reconstructed neuron depicted in Figure 2.1(D). Therefore, considering data formats is important in developing ML solutions, such as deep generative models, which will be discussed in later sections.

Medical datasets, which can be either public or private, are the foundation for developing ML models for CAD systems to assist doctors. Therefore, collecting medical data is identified as a key step for the thesis. As a result, several datasets [23, 24, 25, 26, 27, 28, 29] were published. More details about these datasets are discussed in Section 3.1.1. In DeepSynthBody, which is the novel concept introduced to overcome the data-related problems faced during the development stage of ML solutions, all the medical datasets had to be categorized to make a clear data organization process for the contributors and

Table 2.2: Sample datasets for the 11 categories of the biological anatomy classification. These datasets were selected randomly using Google search. These datasets are selected from the outside of the dataset contributions introducing under this DeepSynthBody study.

Data class	Sample datasets
Cardiovascular	Cardiac MRI dataset [88], ECG data [89]
Digestive	Endoscopy dataset [90, 91], Capsule endoscopy [92]
Endocrine	Hyperspectral imaging [93], Thyroid ultrasound image [94]
Integumentary	Skin lesions [95], Skin image dataset (melanomas) [96]
Lymphatic	CT lymph nodes [97], Lymphography Data Set [98]
Muscular	MRI of muscles of the hand [99], Full body data with muscle [100]
Nervous	Brain activity fMRI data [101], PET-MR Dataset [102]
Urinary	Kidney dataset [103], CI images kidney stones [104]
Reproductive	Human sperm images [105], Embryo dataset [106]
Respiratory	Chest X-ray data [107], Chest CT datast [108]
Skeletal	Bone X-ray dataset [109], Knee MRIs [110]

the end-users of the framework. For this, a biological anatomy classification [85] (11 categories) was used to classify most of the medical datasets (except genome data [86, 87] which is related to the full human body. The genome data will be considered for the DeepSynthBody framework in the future. Table 2.2 presents the 11 classes selected as our classification and corresponding example open-access datasets. These example datasets indicate that most of the data can be classified into these 11 categories.

Even if publicly available, medical datasets can come with other challenges that need to be taken into account. One challenge is the sizes and distributions of medical datasets. If the sizes of these datasets are limited, such as having few data samples, then it directly affects the final performance of ML models. Similarly, if a dataset is imbalanced such as one class has more data and another class is lacking data, then it also affects the performance of the ML models [111, 112, 113, 114]. Despite these problems, privacy concerns of the medical data [115], containing information about patients, is another problem. These privacy concerns directly cause problems for publishing the medical data because medical dataset publishers should follow all the protocols related to publishing medical datasets, as discussed in Section 1.1. In addition to the privacy concerns, making the ground truth data for the medical data is costly and time-consuming. In the medical domain, experts (medical doctors) should perform the data annotation process. Therefore, one of the goals of this thesis is to overcome the data annotation problem and introduce an efficient way to produce medical datasets with ground truth to train ML solutions,

i.e., both reducing the need for medical experts to produce ground truths and bypassing the privacy challenges.

2.2   Machine Learning in Medicine

Different types of ML algorithms are applied to medical data. When researchers and other medical data providers publish datasets to train ML models, they have intended goals to achieve using the datasets. For example, when GI-tract polyp datasets are published with the corresponding ground truth masks [116, 117, 118], the main goal of the datasets is to train ML models to perform polyp segmentation tasks. Therefore, baseline experiments (experimental results coming with dataset papers) and benchmark experiments (experiments performing to achieve the best results compared to the state-of-the-art performance) of a particular dataset are essential to know the capabilities of the ML models trained using the dataset and identify the related practical problems, for example, the generalizability issue of an ML model trained using a single dataset. The baseline and benchmark results coming from ML models can be used to identify the limitations of datasets. For example, suppose every machine learning model shows poor performance for a specific class of a data classification problem. In that case, the problem might be with the data of the particular class. In this regard, this thesis discuss baseline experiments and benchmark experiments. The baseline experiments are discussed with our dataset papers [23, 24, 25, 26, 27, 28, 29], and the benchmark experiments are discussed in our benchmark articles [30, 38, 39, 40, 68, 41, 36, 32, 33, 35, 34].

Most of the ML models trained with medical data can be classified into a regression task [119, 120, 121], classification task [122, 123], detection task [124, 125] or segmentation task [126, 127]. These tasks depend on medical datasets and their intended purposes. ML models trained to solve regression tasks want to predict continuous values (parameters) for a given input data such as numerical input, images, or video inputs. For example, predicting motility or morphology level, which are percentage values, of a sperm sample given as a video is a regression model. In the classification task, ML models need to predict class labels of input data, such as predicting the GI-tract landmark for a given image captured from an endoscopy. In detection tasks, ML modules focus on predicting bounding boxes for regions of interest on images or videos (normally, videos are processed

frame by frame, and this video processing also can be considered as image processing), i.e., predicting polyps in an image of GI-tract. Advanced segmentation tasks perform pixel-wise predictions to mark the region of interest, and this task gives greater details than all other three tasks, for example, predicting the exact regions of polyps using the pixel-wise classification of a GI-tract image. These ML methods have specific evaluation methods based on the objectives.

Evaluating ML models have to be performed properly, which means evaluation processes should reflect the real performance of ML models. For example, data leakage problems [128] should be avoided, the generalizability of ML models should be tested using cross-dataset evaluations, and multiple evaluation metrics should be calculated to show the performance from different perspectives. Otherwise, researchers may produce inefficient solutions which cannot be applied in practical scenarios. According to the type of the ML task, the evaluation methods should be selected. A summary of these evaluation methods is presented in Table 2.3.

One of our studies [31] discusses the importance of evaluating ML models with multiple evaluation metrics and cross datasets for producing better generalizable ML solutions. In addition to the cross dataset evaluations, we have discussed problems of current articles with incomplete evaluation metrics using a literature review of polyp classification as a case study [33]. To overcome this incompleteness of the evaluation results, we have introduced an online tool called MediMetric[1], which can be used to get complete evaluation metrics from the incomplete evaluation metrics for binary classification tasks. The evaluation performance of ML models can be found in baseline experiments, which come with dataset papers, and benchmark papers, which aim to produce state-of-the-art results for a particular dataset. In this thesis, these baseline results and benchmark results are essentials to develop ML solutions to achieve our Sub-objective I and develop DeepSynthBody, which is the main solution introduced in this thesis to achieve Sub-objective IV. Therefore, contributions of ML methods with corresponding evaluations are presented in our series of benchmark articles [30, 38, 39, 40, 68, 41, 36, 32, 33, 35, 34] in addition to the evaluations presented in our dataset publications [23, 24, 25, 26, 27, 28, 29].

---

[1]https://medimetrics.no/

Machine learning (ML) type	Evaluation method
Regression	R Squared (Coefficient of Determination), mean square error (MSE) or root-mean-squared error (RMSE), mean absolute error (MAE)
Classification	Accuracy, F1, Recall (sensitivity), Precision, Matthews correlation coefficient (MCC)
Detection	Intersection over union (IOU), Precision, Recall
Segmentation	IOU(Jaccard index) , F1-score (dice coefficient)

Table 2.3: Example evaluation methods using for the most common ML methods applied with medical data.

2.3 Generative Adversarial Networks

In the above section, regression, classification, detection, and segmentation models known as discriminative models were discussed. As a mathematical definition, the discriminative models capture the conditional probability, for example, $p(Y|X)$, in which X represents data instances and Y represents a set of corresponding labels. In this section, generative models are discussed. These generative models are the most important ML model used in DeepSynthBody, which is introduced as a solution to overcome the data deficiency problem. Generative models learn joint probability distribution compared to the conditional probability of discriminative models. In the formal definition of generative models, they capture the joint probability $p(X, Y)$ if both data instances (X) and labels (Y) exist. Otherwise, the generative models capture only data distribution $p(X)$. There are several types of generative models. Autoregressive models, variational autoencoders (VAEs), Latent Dirichlet Allocation (LDA), Hidden Markov Model, Gaussian Mixture Model, Bayesian Network, VAE, and generative adversarial network (GAN) are a few of them. Among these generative models, two deep generative models, namely VAE [129] and GAN [130], have become popular in the recent research studies [131, 132, 133] of generating synthetic data.

VAE [129] consists of two networks, namely encoder and decoder networks. The basic architecture diagram is illustrated in Figure 2.2 with the basic elements. In the training stage, the encoder converts input data into a latent space represented using mean (μ_x) and standard deviation (σ_x). Then, in the inference stage, only the decoder generates data by sampling the latent vector from the latent space. However, the main disadvantage of using VAEs to generate synthetic data is generating blurry output [134]. In synthetic
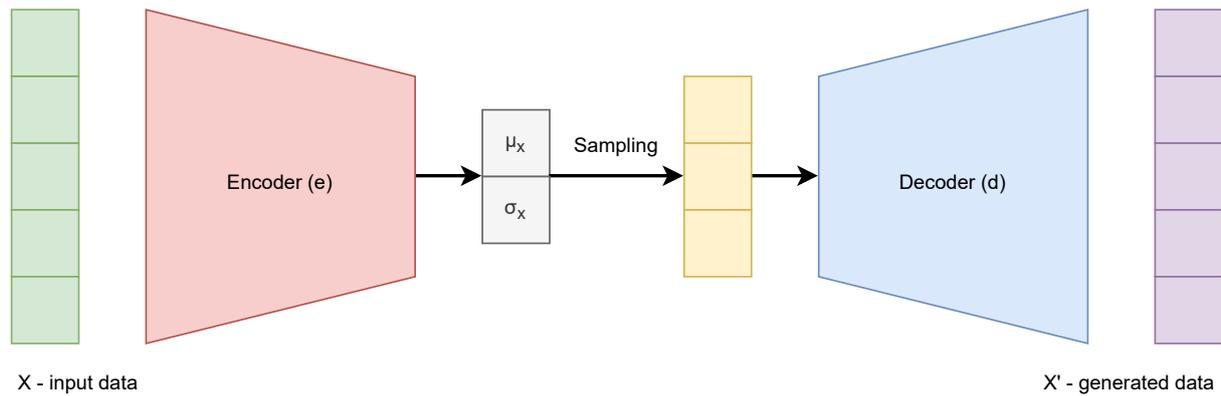
Figure 2.2: Basic architecture of a VAE.

medical data generations, every feature is essential. Therefore, GANs were selected to use as the main generative models to generate synthetic data in this thesis because of high-quality feature-rich generation capabilities. In contrast, GANs are harder to train than VAEs [135].

The basic GAN architecture introduced in 2014 by Ian et al. [130] consists of two DNNs. One is called the generator, and the second one is called the discriminator. The generator's main task is to generate synthetic data by taking a random noise vector as input. The noise vector can be sampled from any statistical distribution, such as normal distribution or Gaussian distribution. Then, the discriminator learns to distinguish generated data from the real data, used to train the GAN architecture. In the training process, the generator and the discriminator are leaning together, which results in a Nash equilibrium [136] problem. If successfully trained, the generator can generate realistic synthetic data samples, which can fool the discriminator. This process is illustrated in Figure 2.3. The objective function (loss function) used in this vanilla GAN architecture is presented in Equation 2.1. However, not every GAN architecture uses the same objective function to optimize the training process. The most common loss functions are summarized in a large study about GAN architectures done by Lucic et al. [137]. Using the most appropriate loss function to generate realistic synthetic data with a stable training process or investigating novel loss functions for a GAN is another important factor in generating realistic synthetic data. Therefore, studying and having comprehensive knowledge about GANs and the corresponding loss functions is essential before developing GANs to generate synthetic data. Otherwise, synthetic data generated from GANs will not cover the real distribution of the training data [138], or the mode collapse behavior [139] of GANs
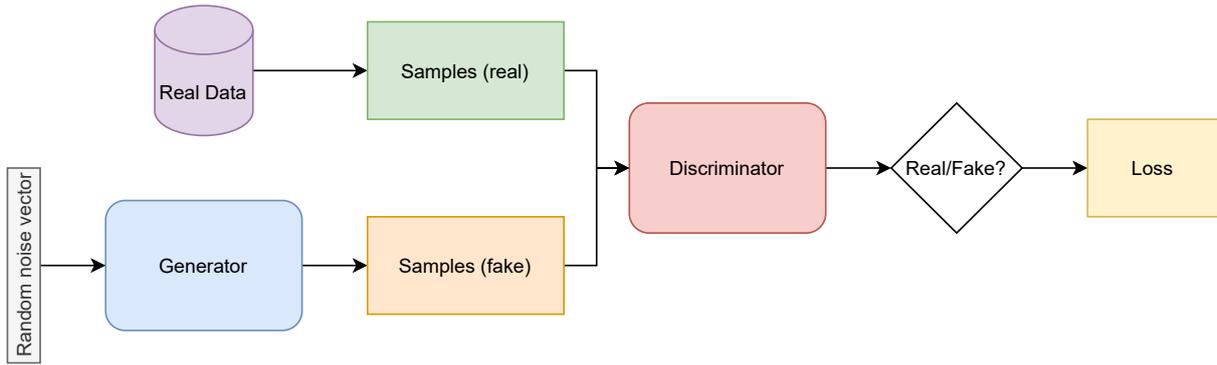
Figure 2.3: A simple representation of the vanila GAN architecture.

may cause.

$$min_G max_D V(D, G) = \mathbb{E}_{x \sim p_d(x)}[log D(x)] + \mathbb{E}_{z \sim p_z(z)}[log(1 - D(G(z)))] \qquad (2.1)$$

After the vanilla GAN, it became one of the trending fields in DL, and different GAN versions for different purposes were published. A summary of the most popular GAN architectures is shown in Table 2.4. For producing better quality synthetic medical data for the DeepSynthBody, the contributors should use the most appropriate GAN architecture. A literature review or preliminary experiments should be conducted to determine the best fitting GAN architecture for a given problem. Good knowledge about the state-of-the-art GANs methods is important for finding a better GAN model for generating synthetic data. In this thesis, novel GANs [70, 75, 67] and modified versions of different GAN architectures [72, 73, 74, 76] were researched and developed. More details about these GANs are presented in Chapter 3.

Not only the designing and implementation of GANs is essential, but also evaluating them. Evaluation of GANs is an active research area by itself. GAN evaluation is not well-defined in terms of how to measure the quality of the generated synthetic data. Theoretically, GANs should produce synthetic data which looks like real data from the whole distribution of the real data used to train the GANs. To measure the performance of GANs, qualitative and quantitative evaluation metrics were introduced in several research papers. Table 2.5 shows standard GAN evaluation metrics presented in the paper [150]. In the synthetic data generation process, the evaluation process plays a significant role in finding suitable GANs to produce synthetic data to replace the real medical data. For example, evaluation metrics can compare two or more GAN models developed for the

Table 2.4: A little from the most popular GAN architectures and their main functionalities. More about other GAN architectures can be found in [140, 141, 142]

GAN name	Description
Vanila GAN architecture [143]	This is the first GAN architecture introduced in 2014. This is capable of generating low resolution images but they are noisy.
Pix2pix [144]	This is a conditional GAN GAN architecture. This model convert an input image from one domain to an output image in another domain. The training process need paired images from two domain which have one to one mapping.
CycleGAN [145]	This paper present a similar mechanism to the Pix2pix implementation. However, the CycleGAN does not need paired training data, then the model can be train using unpaired two datasets from two different domains. Cycle consistency loss was introduced in this study.
StyleGAN, StyleGANv2 [146]	This GAN architecture is capable of generating realistic high-resolution images and the GAN can be controlled to change high-end features as well as fine features. The major drawback of this GAN is, a large training dataset is required to train the model. However, recent advancements introduced to data augmentation method [147] with GANs shows that limited datasets are enough to train new GAN models.
BigGAN [148]	This is another GAN architecture which can generate high-resolution images with high fidelity. A large dataset is required to train BigGAN also, but the quality of generated samples are high.
SinGAN [149]	This GAN architecture is trained using a single image and then, synthetic data is generated similar to the local and global features of training images but different from the training images. As use cases, generating high-resolution images, image editing , harmonization and making animations are focused.

same purpose. However, evaluating GAN models developed by different developers is not an easy task until a common reference calculates evaluation metrics. Therefore, in this thesis, we recommend using qualitative and quantitative criteria to understand the quality of the generated synthetic data.

Table 2.5: A few of GAN evaluation metrics. The complete list of these evaluation metrics and corresponding details with the original references can be found in [150]

GAN evaluation type	Metircs
Qualitative	Average Log-likelihood
	Coverage Metric
	Inception Score (IS)
	Modified Inception Score (m-IS)
	Mode Score (MS)
	AM Score
	Fréchet Inception Distance (FID)
	Maximum Mean Discrepancy (MMD)
	The Wasserstein Critic
Quantitative	Nearest Neighbors
	Rapid Scene Categorization
	Preference Judgment
	Mode Drop and Collapse
	Network Internals

2.4 Synthetic Data in Medicine

Researchers have experimented with GAN in the medical domain for different purposes. In most cases, GAN models have been used as augmentation techniques to increase the size of the medical datasets [151, 152, 153]. Some of them have focused on improving classification [151], detection [154, 155], or segmentation [156] performance using synthetic data generated by GANs. Besides increasing or augmenting data, special types of GANs can perform medical segmentation tasks [157, 158] and generate super-resolution images to make a precise medical diagnosis [159]. AsynDGAN [160], introduced by Chang et al., is another GAN architecture focusing on solving privacy concerns by distributing discriminator networks among data providers to train a GAN architecture.

To the best of our knowledge, there is no other similar concept proposed like the Deep-SynthBody concept, which focuses on producing synthetic medical data for the whole human body to solve the data deficiency problem in the medical domain by addressing, for example, privacy concerns and overcome costly and time-consuming medical data annotation processes. However, few studies developed frameworks to solve privacy concerns of the medical data. The closest framework similar to DeepSynthBody is Synthea[2] [161] which was developed to generate synthetic electronic health records (EHR). Synthea is

---

[2]https://synthetichealth.github.io/synthea/#technology-landing

also running as an open-source project to get contributions from other researchers. This framework focus on generating synthetic EHR to free the medical data from legal, privacy, security, and intellectual property restrictions. Although Synthea focuses its primary goal on producing privacy restriction-free synthetic EHR, which is one of the primary goals of DeepSynthBody, significant differences can be found between Synthea and DeepSynthBody. For example, DeepSynthBody focuses on building a synthetic human body model, while Synthea focuses on making synthetic patient records using synthetic EHRs, which are text-based medical records. Pre-generated records can be downloaded from the Synthea website[3]. The DeepSynthBody concept is not targeting text-based EHR generations like Synthea. Our main focus is on generating realistic medical data similar to the medical data collected from medical instruments used to examine patients, such as biomedical signals and biomedical images.

Moreover, DeepSynthBody provides an advanced well-defined flow from data collection to the end of synthetic data generations focusing on much more advanced additional objectives. These additional objectives provide synthetic data with annotations, define a novel model for the human body, and provide a restriction-free GAN repository for generating synthetic medical data. Additionally, the DeepSynthBody concept publishes GAN models instead of pre-generated synthetic data for the end-users.

Anonymization through data synthesis using generative adversarial networks (ADS-GAN) [162] is another framework to generate synthetic EHR datasets. This framework provides pre-trained GANs to generate synthetic EHR records. Their generation method is based on conditional-GAN, which means to generate synthetic data, real data values should be available. Therefore, they propose to have a trusted intermediate partner to generate synthetic EHR data from real data records. In comparison, DeepSynthBody does not need any intermediate partner because of the in-house GAN training capability introduced in the framework with the corresponding tools. In addition, DeepSynthBody focuses on diverse, complex medical data types compared to normal EHR data considered in the ADS-GAN study.

SynSigGAN [163] was developed by Hazra and Byun to generate privacy restriction-free biomedical signals. However, despite the results in the paper, the GAN is not available in public to generate synthetic data. Similarly, different generative models for dif-

---

[3]`https://synthea.mitre.org/downloads`

ferent types of medical datasets can be found, such as synthetic embryo images [164] and COVID-19 X-ray images [165]. The study of synthesis of COVID-19 chest X-rays shows improvement for ML models used to detect Covid-19 when this synthetic data is used with real data to train the ML model. They also discuss how GAN is used for anonymization. The improvement achieved for the performance motivated us to make a formal framework for synthetic data in the medical domain. DeepSynthBody provides a framework and infrastructure that can share these anonymized data generators compared to the above solutions.

2.5   Summary

Medical data is the key to apply AI solutions in medicine. Therefore, there are many public repositories, which are collecting medical data and share them with researchers. These medical data have different formats. However, the sizes of the datasets are not enough to train a generalizable and well-performing ML model. The sizes of datasets are limited in the medical domain due to, for example, privacy restrictions and the costly and time-consuming data annotation process. These data deficiency problem motivated us to find a solution to tackle the problems. Identifying the correct organ system, the data source, and the medical data formats are essential for developing ML models for CAD systems, such as deep generative models used in our DeepSynthBody concept.

Applying ML techniques and finding suitable models to get better predictions are the main tasks for developing AI-based CAD systems for medical scenarios. The main ML methods include regression, classification, detection, and segmentation. Different ML methods have implicit evaluation techniques, and following them strictly to evaluate ML models is required to find accurate and generalizable AI solutions. Producing ML solutions for baseline experiments or benchmark analyses can give a first idea about a medical dataset and the quality of dataset's content. Additionally, baseline experiments are necessary for analyzing the quality of synthetic data, which will be be used as alternatives.

To generate synthetic data, we selected GANs as the core generative model in this thesis because of the ability of GANs to generate synthetic data with rich features. However, training GANs is more challenging than training other generative models. Therefore, having a good understanding of GAN types and their evaluation methods are important

factors in implementing good generators that can produce synthetic data for solving the data deficiency problem associated with developing ML models for medical CAD systems.

Our proposed DeepSynthBody is a novel concept and a framework addressing the data deficiency problem identified while developing ML models for CAD systems to assist doctors. In this chapter, existing frameworks were explored with similar directions as DeepSynthBody. Most of the solutions focus on text-based EHRs. Our solution, namely DeepSynthBody is designed to generate all the medical data coming through medical instruments except text-based medical data. While some solutions need a third-party data handler to maintain privacy concerns, the DeepSynthBody concept proposes a mechanism to design GANs in-house of the medical data providers. In the next chapter, the DeepSynthBody concept and the corresponding framework are introduced with three case studies.

Chapter 3

DeepSynthBody

In this section, the flow of the DeepSynthBody concept [71], which is the main solution discussed in this thesis to overcome the data deficiency problem, is introduced. The whole framework is discussed under four major steps: collecting real data and analysis, developing generative models, creating DeepSynth data, and explainable DeepSynth AI and DeepSynth Explainable AI. The first section is further divided into two, collecting real data and analyzing real data to discuss the real data collection process and the process of analyzing them, respectively. Under the second step, namely developing generative models, three sub-section are discussed. These are designing generative models and evaluation, publishing deep generative models, and developing a tool called GANEx to perform GAN experiments. In the third section, creating DeepSynth Data is discussed. At the end of the chapter, explainable DeepSynth AI and DeepSynth explainable AI is presented, followed by a summary.

We have developed this framework to tackle the data deficiency problem identified as a major bottleneck to develop AI-based CAD systems in medicine. The main focus of the DeepSynthBody concept is producing synthetic medical data to overcome barriers attached with medical data, such as privacy concerns, the costly and time-consuming medical data annotation process, and the data imbalance problem in the medical domain. The DeepSynthBody concept is not limited to achieve the primary objectives, but it opens new research directions such as finding a synthetic model to define the human body. Additionally, DeepSynthBody can be considered a modern repository to store medical data without any privacy concerns. It can be used as a medical data compression method to store big datasets in limited spaces.

(I) Collecting real data and analysis	Real Medical Data										
	Cardiovascular	Digestive	Endocrine	Integumentary	Lymphatic	Muscular	Nervous	Urinary	Reproductive	Respiratory	Skeletal
	1-D		2-D			3-D			N-D (N > 3)		
	Without annotations			With annotations							
	Real data analysis										
(II) Developing generative models	Generative model designing and benchmarking with qualitative and quantitative analysis reports										
	Publishing DeepSynth models as modules and sample DeepSynth datasets										
	DeepSynthBody (www.deepsynthbody.org)										
(III) Producing DeepSynth data	Cardiovascular	Digestive	Endocrine	Integumentary	Lymphatic	Muscular	Nervous	Urinary	Reproductive	Respiratory	Skeletal
	1-D		2-D			3-D			N-D (N > 3)		
	Unconditional			Conditional							
	DeepSynthBody Generative Functions										

Restricted access → Open access

(IV) Explainable DeepSynth AI and DeepSynth explainable AI

Figure 3.1: Complete framework of DeepSynthBody. Reference for the figure: [71]

An overview of the DeepSynthBody framework is shown in Figure 3.1. There are four major steps namely:

I. collecting real data and analysis.

II. developing generative models.

III. producing deep synthetic data.

IV. explainable DeepSynth AI and DeepSynth explainable AI.

The right-side top arrow in Figure 3.1, *Restricted access*, represents the flow having privacy-related restrictions. The *Open access* arrow represents the open access flow of synthetic data generated to replace real private datasets. These steps are discussed in detail in the following sections.

3.1   Step I: Collecting Real Data and Analysis

Step I in Figure 3.1 is collecting real data and analysis. In this step, real medical data are collected and analyzed for the later steps in DeepSynthBody. Real medical data can be either public or private. If data is private, this Step I should be completed by authorized data providers. If data is public, anyone who wants to contribute to this framework can complete this step. The three sub-processes, data classification, annotation and labeling, and analysis, are discussed separately to simplify the process of the step. The two types

of research contributions can be identified in this step. They are publishing open access medical datasets with baseline experiments and performing benchmark experiments of medical data.

3.1.1 Collecting Real Data

Medical datasets are the key to initiate the DeepSynthBody framework. Hospital and medical research institutions are the sources for collecting real medical data. These medical data come from different sources, such as ECG machines [166], X-ray machines [167], endoscopy machines [168], MRI machines [169], and various other advanced types of machinery collecting human body data. In this thesis, the medical data collection process was performed continuously to achieve Sub-objective II, which also contributes to the data collection process of DeepSynthBody. As a result, seven open datasets were published. The datasets collected in this thesis are tabulated in Table 3.1 with additional two datasets used as case studies. These additional two datasets were not published as dataset papers of this thesis, but we have used them to have different case studies in the later stages of this thesis.

The first three datasets presented using bolded text in Table 3.1 were the selected three datasets. The first dataset is an ECG dataset, but it is restricted for public use because of privacy restrictions. HyperKvasir [23] is the largest public GI-tract dataset consisting of images and videos collected from real endoscopy examinations. This GI-tract dataset consists of polyp images with the corresponding annotations done by experts, unlabelled images, and a set of images belongs to 23 classes. VISEM [69] (sperm video data) was not collected as a part of the thesis, but the dataset is considered as one of the case studies. We have selected this dataset to represent the video data type in our experiments.

The fifth and the sixth in Table 3.1 are two other GI-tract datasets related to Hyper-Kvasir. These are the Kvasir-Capsule [27] and Kvasir-instruments [29] datasets. Kvasir-Capsule consists of images and video data collected from capsule endoscopy. This dataset has $47,238$ labeled images, 43 labeled videos, $4,694,266$ unlabeled images and, 74 unlabeled videos. By comparing the number of labeled and unlabeled images and videos, we understand the capabilities of this dataset for supervised and unsupervised machine learning techniques. However, this dataset does not have any segmented GI-tract images. In contrast to this, Kvasir-instruments is a segmentation dataset with manually annotated

segmentation masks of endoscopic tools. The dataset has 590 images with corresponding mask images of the segmented tools. Providing fewer images with this dataset indicates how hard it is to prepare this type of segmentation dataset with the help of medical experts. So, finding an alternative way to prepare segmentation datasets with medical datasets is important.

The PMData [25] dataset contains general life-logging data and sport activity data. Fitbit versa 2 fitness smartwatch was used to collect sensory data for this dataset. Therefore, the participants of this data collection process were encouraged to wear the watch as much as possible. In addition to this sensor data, all the participants were asked to record their daily activities and fitness levels, such as sleep hours, the mood of the person, etc., in PMSys sports logging app[1]. Furthermore, a Google form was used to collect another set of data: demographic data, food images including drinking, and weights. While this type of data collection is not directly connected with any pure medical data, such as collecting images and signals of the human body using medical instruments, these data are important to know the relationship between daily life and health problems. However, collecting these types of daily activities is challenging, and careful de-identification is needed before publishing data to the public.

PSYKOSE [26] is a motor activity dataset collected from 22 schizophrenia patients and 32 healthy control persons. All the motor activities were collected for an average of 12.7 days using a wrist-worn actigraph device[2]. In addition to the motor activity data, demographic data and the data about medical assessments are given. This kind of detests is essential for predicting the health states and performance outcome of a person. However, motor activity data and the corresponding demographic data are susceptible to privacy. Additionally, collecting health data with multiple sources for the same person is important because finding correlations among health data and other factors such as motor activity can lead researchers to discover hidden behaviors of our human body.

HTAD [28] presents a dataset with wrist-accelerometer data and sound data for the four most common daily activities of human life. These activities are sweeping, brushing teeth, washing hands, and watching TV. Fining the pattern of these kinds of activities can lead to finding new research directions such as assistive technology for older people. Not only as assistive technology, identifying unique patterns of sensor data corresponding

---

[1]https://forzasys.com/pmsys.html
[2]Actiwatch, Cambridge Neurotechnology Ltd, England, model AW4

to specific health conditions such as mental disorders can lead to treat such patients. However, collecting data about daily routines has a significant impact on privacy concerns. Therefore, these kinds of datasets are scarce, and publishing them needs a careful de-identification process. Otherwise, reaching a way to produce similar synthetic data can lead to share data without privacy concerns.

Toadstool [24] dataset has sensor data collected through an Empatica E4 wristband while a set of people are playing Super Mario Bros. In addition to the sensor data, videos captured during the playtime of the game were included. The data was collected from 10 participants of different ages, sex, and different experience levels. Toadstool looks like a non-medical dataset. However, finding correlations between sensor data and game playing patterns will encourage researchers for new areas like health conditions and game playing. Monitored heart rate and facial expression captured during the playtime can be used to find hidden correlations. While many people can collect this data type, data sharing is not as straightforward as a lack of privacy-preserving data sharing mechanisms. Therefore, we made this dataset to perform research to find suitable data sharing techniques and find a way to produce synthetic data alternatives to replace these advanced data collection processes.

The raw medical data should be classified using data classification methods introduced in DeepSynthBody. First, we have to identify the organ systems which we use as a biological classification method.

Table 3.1.: Datasets discussed in this study and corresponding DeepSynthBody Categories. **Bolded** text lists datasets discussing thoroughly in this study as a proof of concepts for the framwork. *Italic* text shows a dataset which was taken from open access datasets and it is not a dataset paper contributed under this study. All other datasets are published as the contributions from this research and they are not analysed.

Dataset	DeepSynthBody category (biological)	DeepSynthBody category (data format)	Availability	Description
ECG data	Cardiovascular	1-D	Restricted	An ECG dataset consists of 15606 samples with eight-leads readings for 10s long duration.
HyperKvasir [23]	Digestive	2-D, 3-D	Public	A dataset with images and videos collected from endoscopy examinations of GI tract.
VISEM (sperm dataset) [69]	Reproductive	3-D	Public	A video dataset consists of 85 video samples collected from microscopic examinations.
Kvasir-capsule [27]	Digestive	2-D, 3-D	Public	A dataset contains 117 videos captured from video capsule endoscopy (VCE) devices.
Kvasir-intruments [29]	Digestive	2-D	Public	An image dataset of 590 annotated frames containing GI-tract procedure tools.
PMData [25]	Cardiovascular, Digestive	1-D, 2-D	Public	The dataset consist of sensor data collected from smart watches and photos taken from smart phones.
PSYKOSE [26]	Muscular, Nervous	1-D	Public	A dataset collected from actigraph devices from 22 patients with schizophrenia and 32 healthy control persons.
HTAD [28]	Muscular	1-D	Public	A dataset consist of home task activities measured using smart wrist-bands and microphones.
Toadstool [24]	Cardiovascular, Muscular	1-D, 3-D	Public	A dataset collected from sensors of Empatica E4 wristband wore to game players and corresponding video frames of the game.

Biological Data Classification

The second row and the third row of Figure 3.1 represent the data classification methods. First, all medical data are classified into 11 categories [85] based on the anatomy of the human body, as presented in the second row of the figure. Then, all data are classified using data formats as represented in the third row. This biological classification was introduced to sort the data in a biological way to identify data using the organ systems of the human body. Then, the data format classification is applied as a supporting classification layer for developers who contribute to developing GANs to generate synthetic data.

The biological categories are cardiovascular, digestive, endocrine, integumentary, lymphatic, muscular, nervous, urinary, reproductive, respiratory, and skeletal. All the input medical data from various sources are considered through one of these categories (see the second column of Table 3.1). For example, ECG data, GI-tract data, and sperm data can be classified under the cardiovascular, digestive, and reproductive categories, respectively (the first three datasets in Table 3.1). If data cannot be considered for only one category, then the data can be classified under several categories. For example, PMData [25], PSYKOSE [26], and Toadstool [24] are classified as multi-classes according to biological categories in Table 3.1. It is essential to identify the correct biological class for data coming from various data sources to find the final categories in DeepSynthBody.

Data Dimension Classification

In addition to the biological classification, the medical data can be further classified using data dimensionality [170, 171]. In this classification, all data formats are classified into four classes, 1-D, 2-D, 3-D, and N-dimensional (N-D), for where $N > 3$. In the DeepSynthBody framework, data dimensionality means data dimensions coming through data sources (medical devices), but not the data dimensions used in data processing techniques. The third column of Table 3.1 presents this classification for our dataset contributions. Considering the dimensionality of real data is important because the dimensions of the real data increase the complexity of generative models (GANs) implementing in later sections (Step II) to generate synthetic representations for the real data. Additionally, data dimensionality decides which GAN architectures to use in Step II: developing generative models.

For the 1-D data format, biosignals (biomedical signals) collected from the human body are considered in this framework. Well-known biosignals are Electroencephalogram (EEG), Electrocardiogram (ECG), Electromyogram (EMG), Mechanomyogram (MMG), Electrooculography (EOG), Galvanic skin response (GSR), and Magnetoencephalogram (MEG). The ECG dataset, PSYKOSE [26] dataset, and HTAD dataset [28] are identified as the datasets with 1-D data format in our dataset contributions in Table 3.1.

On the other hand, medical imaging techniques [172, 173, 174] are commonly used to visualize human body organs, functions, and states for assisted diagnosis and treatment suggestions. Radiography, magnetic resonance imaging, nuclear medicine, ultrasound, elastography, photoacoustic imaging, tomography, functional near-infrared spectroscopy, and magnetic particle imaging are few examples of medical imaging data. Various technologies produce different types of medical images. In DeepSynthBody, medical imaging data is considered under three data format categories: 2-D, 3-D, and N-D, based on the dimensionality of the data obtained. For example, images collected from video cameras can be considered under the 2-D data type. Similarly, videos can be identified as a 3-D data type when the time (represented as consecutive video frames) is considered as the third dimension. However, some data sources produce 3-D data in a spatial domain, e.g., MRI data. However, this type of 3-D data can be classified into 4-D (into N-D because $N > 3$) when the source produces a series of 3-D data points along the time. In addition to 4-D data, some data sources have 5-D data [175], which are considered under the N-D data category. For example, dynamic MRI data with additional information layers such as tracking information has a 5-D data format. Under this definition, all real data sources are identified through 1-D, 2-D, 3-D, or N-D classes.

The data format classifications for the datasets collected under this thesis are presented in the third column of Table 3.1. In this table, multiple data format classifications can be seen for some datasets when the datasets have different types of data. The ECG dataset, which is not public, has the 1-D data format per channel as they received from the data source, and one sample has eight channels in total. While the original data format is 1-D, these ECG samples can be processed as 2-D as well by combining multiple channels together. However, we consider the data format of the original data source as the data format classification to simply this classification. In contrast to this ECG dataset, HyperKvasir [23], Kvasir-Capsule [27] have two different types of data formats. They are

2-D and 3-D. The images collected from endoscopy or capsule endoscopy are considered as 2-D data format. The videos collected from the same instruments are classified as 3-D. These data formats are important to process the data in later steps.

For example, designing image generators are easier than designing video generative models because video generators should consider temporal features compared to considering spatial features of images in the image generators. VISEM [69] dataset has only video data as the main data format, while ground truth data is presented using tabular data. On the other hand, PMData [26] and HTAD [28] data were considered as 1-D data because the main data format coming from the data collection instruments are signals. Toadstool dataset [24] has signals and videos, which means 1-D and 3-D data. Data coming from the Empatica E4 wristband, which was used to collect the players' physiological data streams, is considered 1-D data. The videos recorded from the computer which was used to play the game are considered as the 3-D data format. However, these are the format of raw data. In contrast to raw data formats, one can process these data with a different format; for example, video data can be processed as images if temporal information is unimportant.

The data format classification is done for only the development purpose. This format classification is important only for developers to find proper ML models such as classification, detection, segmentation, and generative models, which are compatible with the dataset.

Data Annotation Classification

After collecting medical data and classifying them according to DeepSynthBody classification, the data can be further categorized into two categories: (i) data without annotations (or labels) and (ii) data with annotations. This classification is represented in the fourth row of Figure 3.1. In this step, whether the data was labeled by experts or not is considered. Generally, most of the data coming from medical systems do not have expert annotations or labels, which are essential to training supervised ML algorithms. Advanced deep generative models such as conditional generative models [175] can be developed if the medical datasets have ground-truth data annotated by medical experts. The conditional generative models take labels (or other kinds of annotations such as pixel-wise classification) as input parameters and produce synthetic data conditioning on the input

annotations. While one of the primary objectives of DeepSynthBody is to reduce annotation cost and time required from experts, conditional GANs should be investigated. Therefore, producing annotated medical data by experts in this stage can help to train deep generative models to overcome the problem of medical data annotations.

Annotations or labels of medical data are different from dataset to dataset. Generally, medical datasets have continuous numerical values, discrete numerical values, class labels, coordinates such as bounding boxes or pixel-wise classifications (e.g., segmented mask). Medical experts can use different kinds of tools for annotating different types of ground truths. These tools may vary from simple image viewers to advanced AI-aided image mask generation tools or expensive medical data analysis tools [176, 177, 178]. However, an expert in the medical domain must operate these tools. While some tools can suggest or predict similar types of annotations, the experts should confirm the final annotations, which will be used as ground truth data for ML algorithms. This expert annotation process needs the medical experts' valuable time, which is costly. Therefore, the DeepSynthBody framework targets handling this problem also.

As explained above, if experts annotations are available, the annotations can be used to train advanced generative models such as conditional GANs. Therefore, experts' annotations were collected for most of the data sets tabulated in Table 3.1. The HyperKvasir dataset [23] consists of image labels and pixels-wise annotations (segmentation masks) for a part of this big dataset. Providing image labels is easier than providing segmentation masks, which represent pixel-wise annotations. Experts' knowledge was used in both annotation processes, but the segmentation annotation process took more time as expected than classifying into the labels. The HyperKvasir dataset consists of unlabeled data, images and videos also. In this context, this dataset can be classified as a dataset with and without data annotations.

The Kvasir-Capsule dataset [27] has labels for the images and the videos. However, in the current version of this dataset, there is not data with pixel-wise annotations. However, classification labels assigned by experts are used to prepare the labeled data. In addition to these labeled images and videos, the rest of the unlabeled images and videos were included without ground truth data because labeling them all is the costly and time-consuming task. If an alternative way to prepare labeled or annotated datasets automatically can be researched, then the expensive and time-consuming medical data annotation process can

be avoided.

In addition to the above GI-tract datasets, the Kvasir-instrument [29] dataset consists of only pixel-wise segmented images, which include instruments used in the colonoscopy examinations and operations. Therefore, this dataset can be identified as a dataset with annotated data. On the other hand, datasets [24, 25, 26, 28] collected through smart watch sensors or special wearable sensors can be considered datasets with annotations because manually identified events were reported in these datasets.

Selecting Case Studies

From the datasets presented in Table 3.1, only three different medical datasets were selected for case studies in this thesis, i.e., representing the various data types supported by our framework. They are an ECG signal dataset, a GI-tract image dataset, and a sperm video dataset. The ECG dataset is not published as a dataset paper. Therefore, this restricted ECG dataset is a perfect example for our Sub-objective IV, which focuses on generating synthetic data instead of the real dataset. On the other hand, the GI-tract [23] dataset is the largest image dataset published under this thesis, and this dataset represents biomedical images. The third dataset is an open-access video dataset [69]. This sperm dataset was selected because of the video data format, and the dataset represents another organ of the human body, while this dataset was not published as a contribution of this thesis. In this section, we discuss the three case studies with comprehensive details.

The ECG dataset is restricted, and only authorized people can access it. As a result, a dataset paper cannot be published. This dataset represents the biomedical signal data format which is considered under cardiovascular class and 1-D data format in DeepSynthBody. In this dataset, each ECG signal consists of readings from eight channels called in the cardiovascular context as channels `I, II, V1, V2, V3, V4, V5, V6` for 10-sec long duration. The eight readings can be converted to 12-leads ECGs mathematically by calculating missing leads `III, aVR, aVL, and aVF` using the following equations 3.1. The sample rate is 500 per ECG sample. Then, there are 5000 data points per lead. A
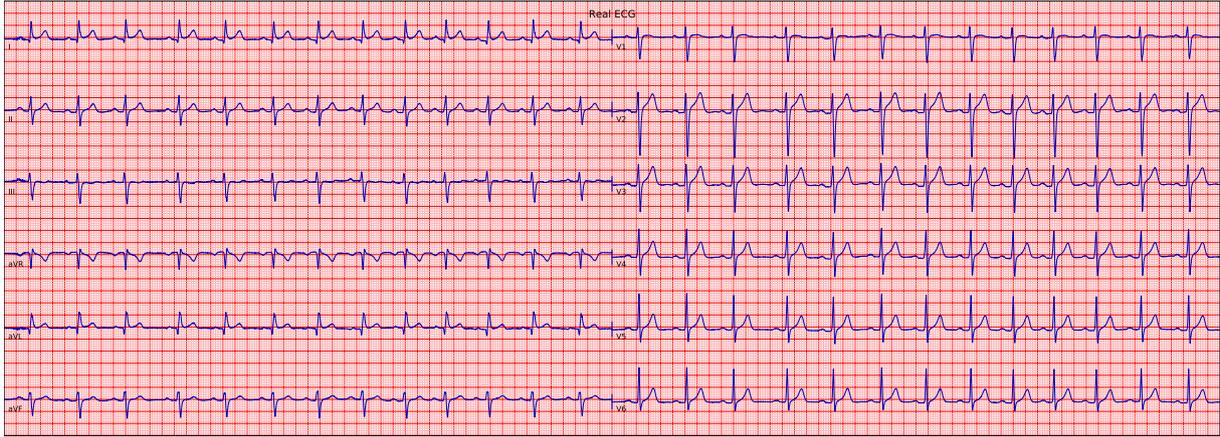
Figure 3.2: A sample of 12-leads 10-sec real ECG. Figure reference: [41]

sample from this dataset is depicted in Figure 3.2.

$$III = II - I$$
$$aVR = -0.5 \times (I + II)$$
$$aVL = I - 0.5 \times II \tag{3.1}$$
$$aVF = II - 0.5 \times I$$

These ECG signals have been collected from two populations. One population is the Danish General Suburban Population Study (GESUS) [179] which consists of $8,939$ samples, and the other one is the Inter99 study [180] (CT00289237, ClinicalTrials.gov) consists of $6,667$ samples. In total, there are $15,606$ ECG samples. All the collected ECGs were analyzed using a well know ECG analysis system named MUSE [181]. These MUSE reports are used as ground truth for this ECG dataset, and the reports contain important characteristics of ECG signals. The important characteristics of a single ECG pulse are depicted in Figure 3.3. According to the MUSE reports, all the ECGs are classified under four main classes as tabulated in Table 3.2. Other important ECG properties collected from the MUSE system are discussed in the benchmark paper [41].

The HyperKvasir dataset [23] consists of labeled images, segmented polyp images, and unlabelled images and videos. The labeled images consist of $10,662$ images under 23 classes. In the segmented polyp images, there are 1000 polyp images and corresponding ground truth masks annotated by experts. The unlabelled images have $99,417$ images, and there are 374 videos with 30 different classes. This dataset represents the biomedical
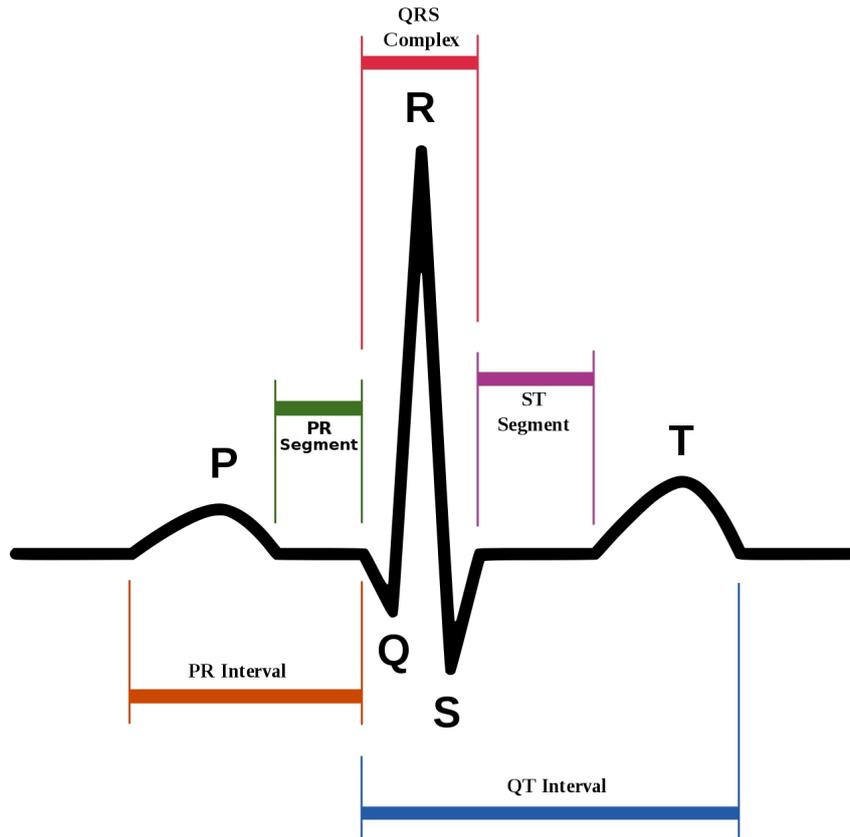
Figure 3.3: The common ECG characteristics. Reference for the image: [182]

Table 3.2: Different classes identified using the MUSE system analysis. Bold numbers represent "Normal" category ECGs which are going to be used as training data for GAN models used in later stages of DeepSynthBody. Reference for the table: [70]

Category	GESUS dataset	int99 dataset	Total
Normal	**3558**	**3675**	**7233**
Otherwise Normal	2370	1536	3906
Abnormal ECG	2118	905	3023
Borderline ECG	893	526	1419
Total	8939	6642	15581

imaging data format considered under digestive class and 2-D and 3-D data formats in DeepSynthBody. However, the labeled images, the segmented images, and the unlabelled images are used as case studies in this thesis, and it means, only 2-D data format is considered.

The labeled 23 classes and the number of images per class are illustrated in the graph in Figure 3.4. These images and corresponding class labels were used in baseline experiments performed for the dataset paper [23]. Then, unlabelled GI-tract images of the HyperKvasir dataset, as depicted in Figure 3.5, were used to train a GAN in developing generative

#images per class for the labeled images

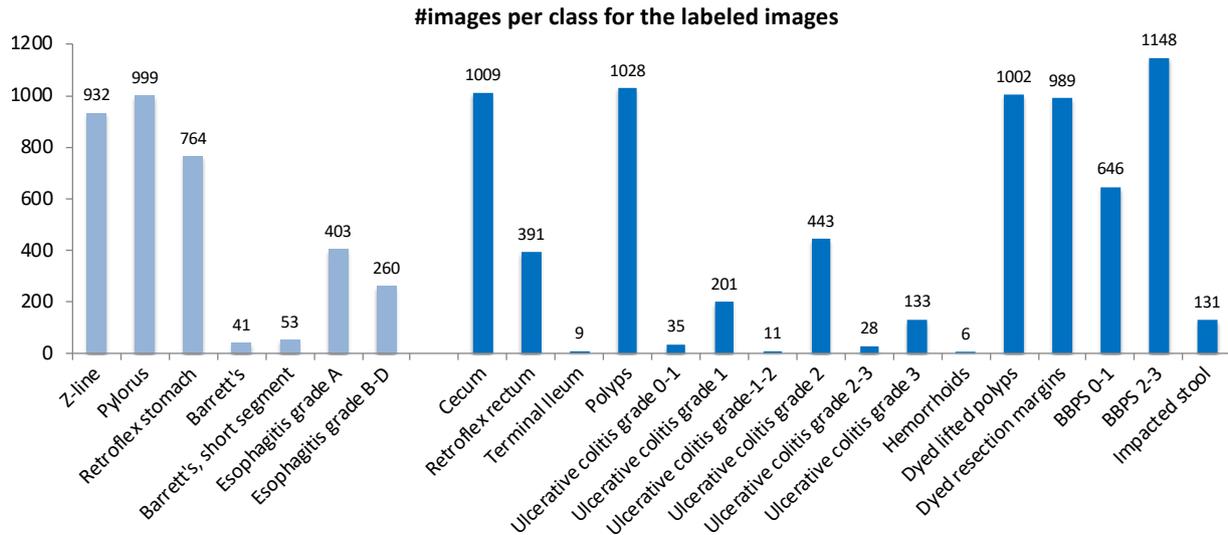Figure 3.4: The 23 classes of the HyperKvasir dataset and the number of iamges per class. The light blue bars represent classes under upper GI-tract and the dark blue bars represent lower GI-tract images. Reference for the plot: [23]

Figure 3.5: Sample images from unlabelled folder from HyperKvasir dataset. Reference for the image: [23].

models of DeepSynthBody. Polyp images and corresponding masks from the segmentation data folder are illustrated in Figure 3.6. The polyp data was used to train a GAN model, which was developed to show the possibility of using GANs as an alternative method for the costly and time-consuming data annotation process performed by domain experts. More details about the whole HyperKvasir dataset are presented in our dataset paper [23].

The VISEM dataset introduced by Haugen et al. [69] has 85 sperm videos recorded from sperm samples collected from different participants. The sperm video dataset consists of analysis data reports produced by experts in the domain of sperm analysis. The sperm dataset is classified under the reproductive system, and it covers the 3-D data format. Example frames extracted from the videos of this dataset are illustrated in Figure 3.7. Different density amounts of sperm counts are shown in this figure from left to right with low-density to high-density, respectively. The collected analysis reports attached with the sperm dataset give information about the morphology and motility level
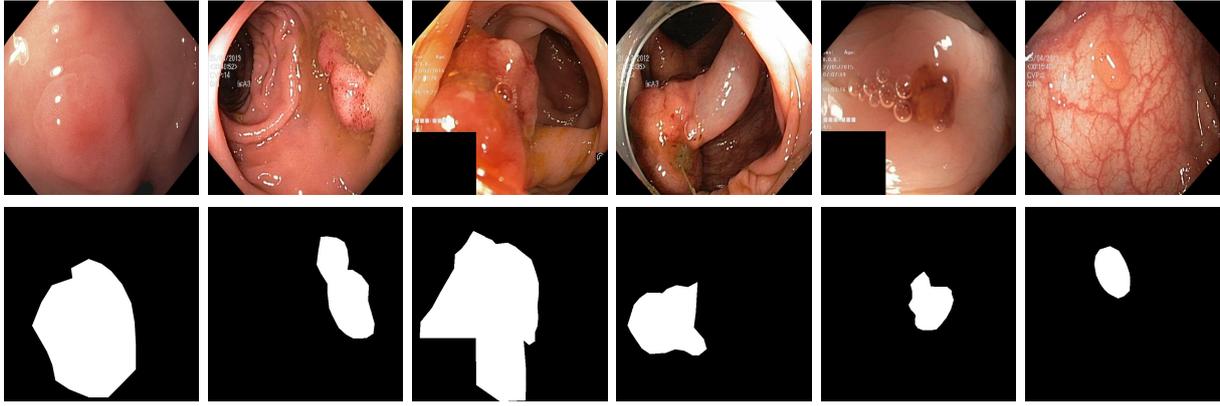
Figure 3.6: Sample images and corresponding masks from HyperKvasir dataset. Reference for the image: [23]
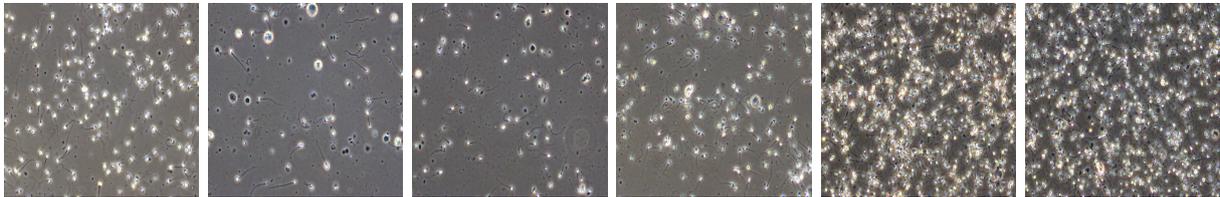
Figure 3.7: Sample frames extracted from different sperm videos from the sperm dataset (VISEM) [69].
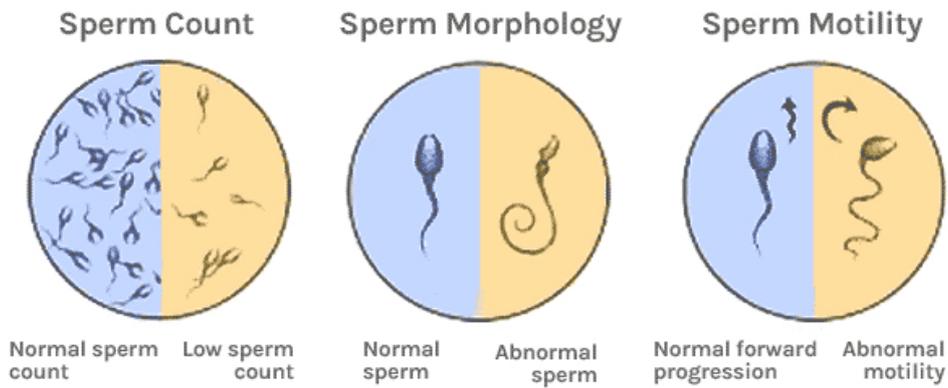
Figure 3.8: An illustration showing important sperm quality measurements. Reference for the figure: [183].

of the 85 sperm samples. Figure 3.8 shows the common quality measurements performing in sperm analysis. They are counting sperms, finding abnormal sperms (sperm morphology level), and finding abnormal movements of sperms (motility level), as illustrated in Figure 3.8 from left to right. Then, the main goal of this dataset is to predict the values in the analysis report automatically using ML techniques. More details about the sperm dataset can be found in the original dataset paper [69].

3.1.2   Analysis of Real Data

Performing benchmark analysis of real medical data is an important step in the Deep-SynthBody framework because it gives the initial understanding and inherited challenges about the datasets incoming to use the framework. Generally, baseline experiments and corresponding results are presented through dataset papers. However, benchmark experiments are the only source to know statistics about the private medical datasets when publishing dataset papers are not allowed because of privacy restrictions. Moreover, advanced analyses are performed in benchmark studies that focus on developing ML solutions rather than publishing datasets. Therefore, this section discusses benchmark analysis details of the selected datasets. The selected datasets are the ECG dataset [41], HyperKvasir dataset [23] and sperm dataset [69].

Electrocardiogram (ECG) Signal Analysis

The ECG benchmark analysis study [41] conducted under this thesis has two objectives. One is to predict ECG properties (see Figure 3.3), namely QT-interval, PR-interval, QRS-duration, heart-rate, J-point elevation, T-wave amplitude, and R-peak amplitude using regression ML methods. The second objective is to predict a person's sex (gender) from ECG signals using ML methods used for classification.

Using 12-leads 10-sec format or median ECGs produced from 12-leads ECGs can be used to predict regression values of the ECGs. The median ECG is a normalized single beat version of the long ECGs. Therefore, both types of input formats, 12-leads 10-sec, and the median format were evaluated as inputs to our ML models used to predict the properties of the ECGs. For each property of ECGs, separate ML models were implemented using convolutional neural network (CNN) techniques. On the other hand, to predict the sex, only the median ECGs were used because we needed to find the correlation between interval-specific features. Medical people are not interested in rhythm-based sex prediction.

All the CNNs were trained and evaluated using five-fold cross-validation to perform a better generalizable evaluation. Quantitative evaluations have been done using MAE and RMSE. In addition to evaluating models' predictions, the GradCAM [42] approach was applied to explain the predictions from CNNs. More details about this ECG analysis and benchmark results can be found in the full article [41]. Referring to this benchmark

analysis is the only way to understand this dataset because of the restrictions on sharing the real dataset. However, the methods are not reproducible because the dataset is restricted. The capabilities of DeepSynthBody to solve such privacy issues are discussed in later sections.

Gastrointestinal-tract Image Analysis

For GI-tract benchmark analysis, several experiments were performed for two different types of tasks, classification and segmentation. Under the initial objectives, we performed these experiments to develop ML models for CAD systems to assist doctors. However, under DeepSynthBody, the main goal of these experiments was changed to benchmark analysis. The summary of all the GI-tract analyses performed for the thesis is tabulated in Table 3.3. Some of the GI-tract analyses [30, 36, 35] have been performed as a part of competitions such as MedicoTask [184] and EndoCV-2021[3] grand challenge, which has used similar GI-tract data to HyperKvasir data [23] used in this thesis. Participating in competitions and solving the tasks given by the organizers helps to make benchmarks and analyze them globally with other participants of the competitions. Our initial objective was to produce well-performing ML models for CAD systems to assist doctors. However, the participating competitions and providing well-performing solutions such as the winning solution [35] provided to the EndoCV-2021 make them popular and get researchers' awareness to enhance them.

Moreover, the cross-data evaluations performed in our paper [31] show the data-bias problem occurred due to training ML models using a single training dataset. This generalizability issue occurs due to the lack of diverse medical datasets. This medical data shortage is identified as the main research question of this thesis. Additionally, in this study, the requirement of fair evaluations using multiple metrics such as accuracy, recall, precision, F1, MCC, and specificity were discussed when the cross dataset evaluations are performed as proof of generalizability.

Not only producing benchmark results, but proper evaluation criteria used to analyze them are essential. Therefore, an online calculator[4] [33] to calculate proper evaluation metrics for binary classification models was implemented with given proper guidelines using the GI-tract images classification as a case study. Using this tool, researchers (or

---

[3]https://endocv2021.grand-challenge.org/
[4]https://medimetrics.no/medimetrics/

other users of this tool) can calculate evaluation metrics for their studies and calculate missing metrics of other relevant studies that need to be compared. This tool makes a common platform for comparing studies fairly.

The performance of an ML model can depend on the resolution of input images to CNNs. Therefore, another investigation [32] was conducted to find the correlations between input resolutions and output performance using GI-tract images. Two different CNNs models (ResNet-152 [185] and Densenet-161 [186]) to classify 23 classes of the labeled folder in the HyperKvasir dataset [23] were investigated and presented the importance of having high resolutions images for CNNs.

A total of six benchmark analyses have been performed in this thesis to achieve Sub-objective III using the GI-tract datasets. The six models consists of four classifications [30, 31, 36, 32, 33], and two segmentation tasks [36, 35]. However, in this section, we considered all these implementations as benchmark analyses because our Sub-objective III is not producing ML models for CAD systems but investigate the data-related problems. These benchmark evaluations and corresponding results using the GI-tract data can be found in original articles tabulated in Table 3.3.

Sperm Video Analysis

According to the data and ground truth provided in the sperm dataset [69], the intended research work is to predict the morphology and motility level of the sperm samples in the dataset. The prediction of morphology and motility levels were identified as regression problems. The summary of all the studies conducted using this sperm dataset for this thesis is tabulated in Table 3.4.

We have performed three studies [38, 39, 40] using different pre-processing techniques and various types of CNNs. The main objective was to predict the morphology and motility levels of the sperm videos, which contain recorded videos of microscopic sperm analyses. Dense-optical flow [189] and Lucas-Kanade's algorithm [191] to predict sparse optical flow were investigated as pre-processing techniques for the study [38]. In addition to the optical flow extractions, stacked gray-scale video frames as input were tested. Moreover, video frames were reshaped to vertical frames and stacked to prepare new data structures to compress multiple video frames into one. This new structured data were also investigated in the study [38].

Table 3.3: GI-tract analysis done for producing baseline results and benchmark results. Two-type of analysis and different type of ways to produce baseline and benchmark results are tabulated here. These analysis results are relevant to the layer of collecting real data and analysis of DeepSynthBody.

Study	Analysis type	Description
[30]	Classification	Two CNNs are presented in this study to classify 16 classes of GI-tract finding given in the dataset of MedicoTask 2018 [184].
[31]	Classification	This study shows the importance of performing cross-dataset evaluations because training ML models using small datasets shows the data-bias behaviours [187].
[33]	Classification	Importance of fair evaluations of the predictions from ML solutions is discussed in this study. Therefore, an online tool to produce proper evaluation results is presented and published with this paper. The tool can help researchers to evaluate classification models. The study was validated using a review of studies of GI-tract analysis.
[32]	Classification	These studies show the effect of the resolutions of the input images using with CNNs. The importance of having high-resolution medical images is emphasised in these studing using GI-tract images as case study data.
[36]	Segmentation	The data augmentation method (PYRA) introduced in this study discuss how grid-like augmentation can improve the generalizability of polyp segmentation. This the segmentation solution proposed to the benchmark challenge in the Medico task at MediaEval 2020 [188].
[35]	Segmentation	The winning solution of EndoCV2021 is presented in this paper. Participating competitions and producing ML solutions for them help to figure out limitations and challenges of real medical data sources.

The dense-optical flow extractions with different amounts of frame strides were investigated in our study [39] for the sperm benchmark analysis problem in MedicoTask 2019 [190]. The stride amount is the gap between video frames extracted to calculate the dense-optical flow. The three-fold cross-validation with ResNet-34 [185] was performed to evaluate the models. For the same task, an auto-encoder-based solution was presented as a new submission [40]. In the second solution, auto-encoders were used to extract temporal information from stacked input video frames. The extracted temporal features act as images to CNNs to predict morphology and motility levels of the sperm videos. The extracted features are not readable to humans. However, CNNs trained using these features could learn to predict the morphology and the motility levels of the sperm samples.

Table 3.4: Summary of real sperm data analysis. Predicting motility and morphology is the main research problem with this dataset. The analysis type of this dataset is regression.

Study	Analysis type	Description
[38]	Regression	Four type of pre-processing techniques were experimented to predict morphology and motility level of the sperm videos.
[39]	Regression	Using the Dense-optical flow [189] algorithm, the videos were pre-processed before passing them into CNN architectures to predict morphology and motility levels. This implementation was submitted to MedicoTask-2019 [190].
[40]	Regression	Auto-encoders were used to extract temporal features into $2D$ spatial domain and the featured were analysed using CNNs tp predict morphology and motility levels of perm samples. The solution was proposed for MedicoTask-2019 [190].
[68]	Regression	A challenge named BioMedia organised for the ACM Multimedia grand challenge 2020. Participants were asked to develop ML solutions to predict morphology and motility levels automatically.

These benchmark results show how difficult to predict the motility and morphology levels only using a small dataset. The results of these experiments reflect the quality of the dataset and also the requirements to improve it. More details about these benchmark analyses performed for the sperm dataset can be accessed from the original papers [38, 39, 40].

3.2 Step II: Developing Generative Models

Step II is the core step of the DeepSynthBody framework. This step is two folds. First one is designing generative models and finding the best models using evaluation processes. The second is publishing the best generative models to the end-users who need synthetic data. Different GAN types and the evaluation methods used to evaluate deep generative models are discussed followed by the methods for publishing GAN models in the DeepSynthBody framework for the end-users.

3.2.1 Generative Model Design and Evaluation

Designing and evaluating GANs for generating synthetic data is the first process in Step II, developing generative models. After collecting and analyzing real medical datasets in Step I, GANs should be investigated to generate synthetic data to achieve the sub-objective III. Sub-objective III focuses on generating synthetic data to overcome the medical data deficiency problem which is the major obstacle for developing AI-based solutions in the medical domain.

The three datasets, analyzed in the data analysis stage, the ECG dataset, the GI-tract dataset, and the sperm dataset, were used as case studies. Comprehensive details of the designing GANs are discussed in this section because the GAN designing and getting state-of-the-art performances are essential for DeepSynthBody as it is the core of this framework. In addition to the GANs design methodology, a novel tool named "GANEx", a graphical user interface (GUI)-based GAN training tool, was introduced. A summary of all GAN-related studies performed for this thesis is summarized in Table 3.5.

Generating Synthetic electrocardiogram Signals

The ECG dataset discussed in our benchmark paper [41] would be a popular dataset for the people doing ECG analysis if it is not a private dataset. Unfortunately, many datasets like this are hidden from researchers as a result of privacy concerns. Therefore, GANs for generating synthetic ECGs were developed in this thesis to generate synthetic ECG data to share public instead of the restricted real dataset.

The first GAN architecture to generate synthetic ECG data was inspired by the Wave-GAN [192] architecture introduced by Donahue, McAuley, and Puckette. The original WaveGAN was developed to generate synthetic music. Therefore, in the first stage, the WaveGAN architecture was modified to generate ECG signals having a shape of 8×5000, which is the shape of eight-leads $10s$ long ECG samples of the dataset, and it was named WaveGAN*. Then, generated samples from WaveGAN* were analyzed qualitatively and quantitatively. The qualitative analysis was done by inspecting 12-leads plots, and for quantitative analysis, the evaluation reports collected from the MUSE system were used. According to the results, WaveGAN* had to be improved further to get better synthetic ECGs. Therefore, a novel architecture named Pulse2pulse [70], inspired by the UNet architecture [193], was introduced for the DeepSynthBody framework in this thesis.

Study	Task of GANs	Description
[70]	Generate synthetic ECG	A novel GAN architecture called Pulse2pulse was introduced to generate synthetic 10s long ECGs with eight-leads to overcome privacy issues of the real dataset.
[72, 73]	Pre-process input data	GAN architectures were experimented to fill a part of GI-tract images, which is the green box appeared at the bottom right corner of the images in HyperKvasir dataset [23].
[74]	Generate synthetic video frames	A GAN architecture named *Vid2pix* with a 3D CNN were investigated to generate synthetic Pilcam video frames [27] for time step $t+1$ conditioning on time steps $t, t-1, t-2$.
[67, 75]	Generate synthetic images with corresponding ground truth mask	GAN architectures were experimented to generate synthetic polyp images and corresponding ground truth mask as proof of concepts to solve privacy issues and medical data annotation cost problem.
[76]	Generate synthetic painting to sperm video frames	A GAN model was experimented to generate a painting like spots instead of sperms in a sperm video frame. This study was focused to generate sperm locations in a synthetic paintings for simple sperm analysis.
[77]	A tool to prefrom GAN experiment	GANEx is a tool with a GUI to perform series of GAN experiments for non-computer science people who want to produce data to DeepSynthBody.

Table 3.5: Summary of GAN-related experiments preformed under this thesis.

ECGs from the *Normal ECG* category of the dataset were used to train both GAN architectures because the *Normal ECG* category is the biggest population of the dataset (refer the Table 3.2). The discriminator used for both GAN architectures was adapted from the discriminator introduced in WaveGAN [192]. The modified WaveGAN generator, Pulse2pulse generator, and discriminator used for both GAN networks are illustrated in Figure 3.9. The complete architecture details are discussed in the full paper [70].

For both models, WaveGAN* and Pulse2Pulse, the best checkpoints were found using MUSE analysis reports collected from generated $10,000$ ECGs per checkpoint from every 500 epochs. Then, the two best checkpoints of WaveGAN* and Pluse2pluse were evaluated further for better understanding before publishing them to the end-users of DeepSynthBody. Five main properties of an ECG, namely RR, P duration, QT interval, QRS duration, and PR interval, were selected to compare the distributions of the selected best checkpoints. The distribution plots are illustrated in Figure 3.10. The blue color dots represent real normal ECG samples, and orange color dots represent generated ECG

(a) WaveGAN* generator.

(b) Pulse2pulse generator.

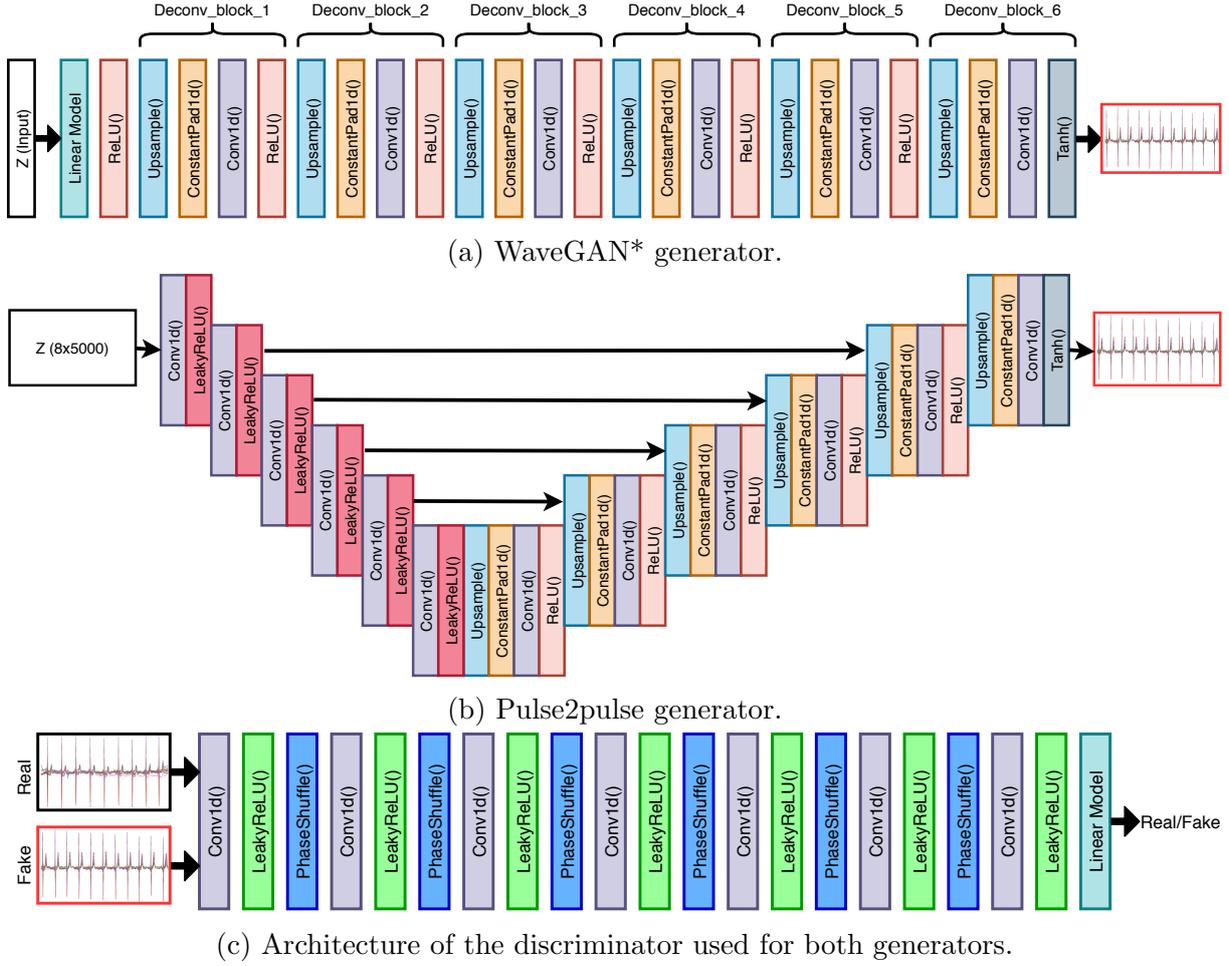(c) Architecture of the discriminator used for both generators.

Figure 3.9: Model architectures of the generators and the discriminator used to generate synthetic ECGs. WaveGAN* uses a 1D noise vector with 100 points. Pulse2Pulse uses a 2D noise vector with size of 8×5000. Reference for the figure: [70].

samples from WaveGAN* and Pulse2pulse.

Comparing distributions of ECG properties, WaveGAN* shows less accurate distribution overlaps with the distributions of real data compared to Pulse2Pulse. This difference can easily be noticed from the row presenting correlations between PR interval and other properties. Also, WavGAN* generated faulty synthetic ECG samples making more "nan" values in the MUSE analysis report than Pulse2Pulse. The MUSE algorithms give "nan" values when the algorithm cannot predict the specific property of an ECG. These statistical comparisons are discussed in our full paper [70].

After finding that the novel Pulse2Pulse architecture can generate better quality synthetic ECGs than WaveGAN*, a large synthetic ECG dataset with 150,000 samples was generated using the best checkpoint of Pulse2Pulse. Then, the synthetic dataset was analyzed using the MUSE system to predict the properties of the ECGs. From the MUSE
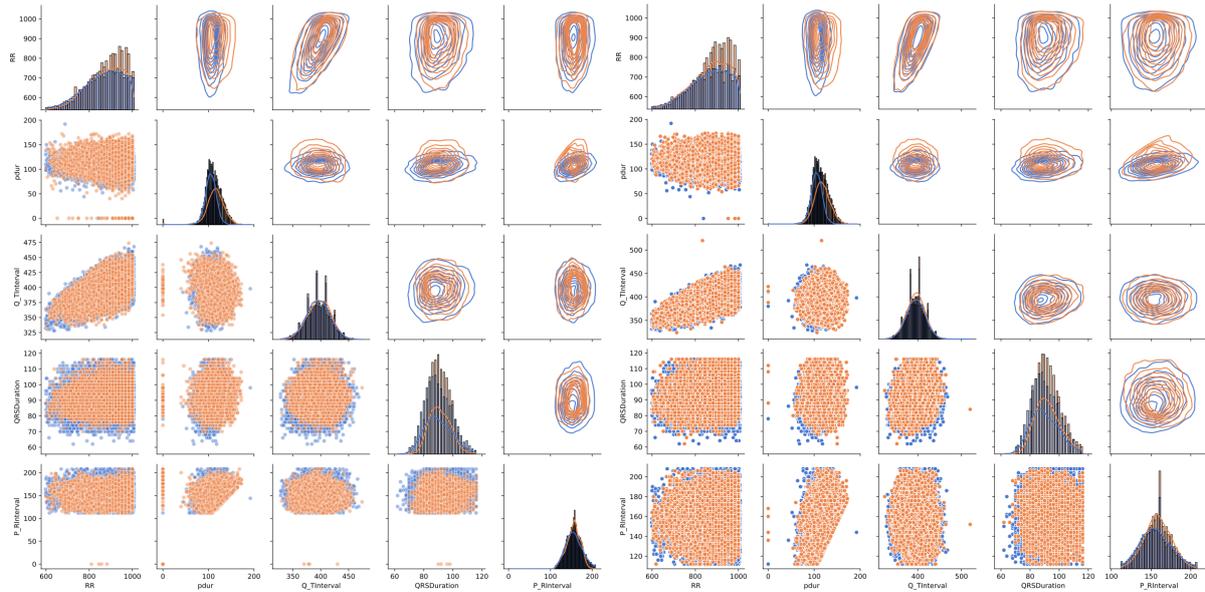
(a) From WaveGAN*.   (b) From Pulse2pulse.

Figure 3.10: Comparisons of MUSE predictions using characteristic distribution plots. Blue color plots represent real normal ECG distributions. Orange color plots represents distribution of fake ECGs generated by WaveGAN* and Pulse2pulse respectively in Figure 3.10a and Figure 3.10b. The "nan" values of the selected five features of "Normal ECG"s were converted into 0 to identify predicted "nan" values by the MUSE system.

analysis report, the most important nine properties, namely *RR, Ventricular Rate, pdur, QT interval, QRS duration, PR interval STJ V5, RPeakAmp V5, and TPeakAmp V5*, were further analyzed statistically, and the collected results are tabulated in Table 3.6 to compare with the real Normal ECG data statistics.

Table 3.6 presents statistics collected from three datasets for the selected parameters. First, statistics about the real ECG data (filtered "Normal" ECGs), used to train the GAN models are tabulated. Then, statistics about all the generated 150, 000 ECGs and statistics about filtered "Normal" ECGs (121977) from 150, 000 ECGs were tabulated. To achieve sub-objectives II and IV, collecting and developing medical data and developing generative models to generate synthetic data, the synthetic ECGs should have similar characteristics as real ECGs. According to the results presented in Table 3.6, the synthetic ECGs show similar statistical properties to real ECGs, such as equal or very close mean and std values for ventricular rate and QT interval. To present the qualitative properties of synthetic ECGs generated from Pulse2pulse, Figure 3.11 shows two synthetic ECG samples identified as "Normal" according to the MUSE report. Additionally, the 150, 000 synthetic ECG dataset and the filtered 121977 "Normal" ECGs can be downloaded with

Table 3.6: Comparison of MUSE analysis reports' statistics for selected ECG properties.

	Real				150k				All Normal (121977)			
	Mean	Std	2.5%	97.5%	Mean	Std	2.5%	97.5%	Mean	Std	2.5%	97.5%
RR	866	90	670	1000	870	91	667	1000	870	87	682	1000
VentricularRate	70	8	60	90	70	8	60	90	70	8	60	88
pdur	105	12	82	130	118	17	84	152	117	17	86	152
Q_TInterval	395	21	352	436	395	21	354	436	395	20	354	434
QRSDuration	90	9	74	110	93	10	78	114	92	9	78	112
P_RInterval	156	19	120	198	159	18	124	194	158	17	124	192
STJ_V5	2	27	-44	58	16	36	-54	92	18	33	-44	87
RPeakAmp_V5	1287	402	600	2163	1272	404	561	2114	1276	370	615	2031
TPeakAmp_V5	343	137	126	664	360	141	141	678	364	134	151	668

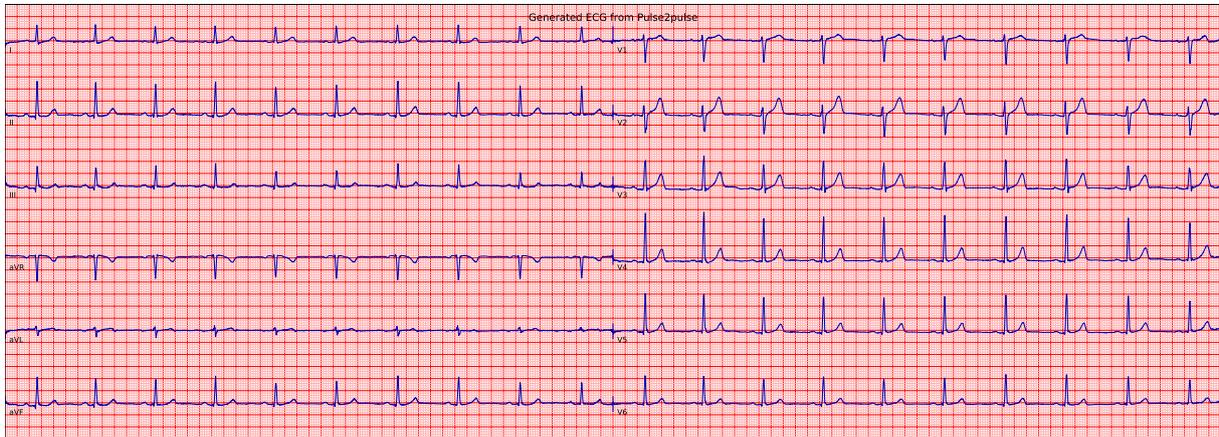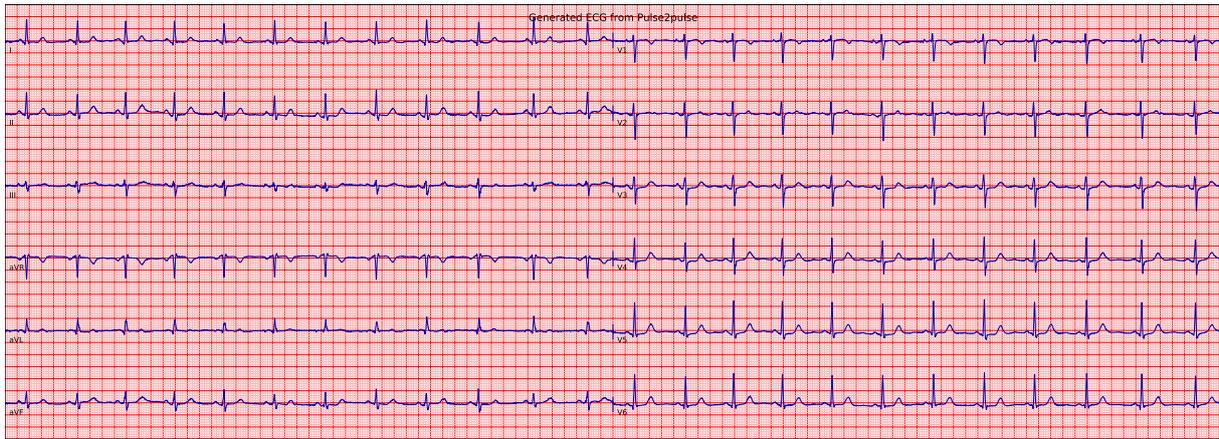Figure 3.11: 12-leads plots of fake ECG samples from the novel ECG generator introduced in this study: Pulse2pulse.

the corresponding MUSE reports from `https://osf.io/6hved/`.

In summary, we could see that our Pulse2Pulse generates realistic synthetic data with very close properties to the real dataset. Then, these generated synthetic ECGs can be used to share to the public instead of the real dataset with privacy concerns.

Generating Synthetic gastrointestinal-tract Images

The HyperKvasir dataset [23] is used as the main case study to experiment with GANs for GI-tract data. Additionally, the Kvasir-Capsule [27] dataset is used. Using these datasets, several GANs were developed to investigate how GANs can generate synthetic medical image data, in this case, GI-tract images.

Several preliminary experiments were performed to use GANs to fill missing parts of GI-tract images [72, 73] and predict future frames of the Pilcam videos of GI-tract videos [74]. The studies [72, 73] focused on removing green boxes that appeared in GI-tract images by generating synthetic filling using GANs. Sample GI-tract images with green boxes are presented in Figure 3.16. The green box removing process is a preprocessing step to prepare GI-tract images for other ML models. Then, the main goal of this study is to find the effect of removing green boxes that appeared in the GI-tract images on the HyperKvasir dataset by replacing the green box with a generated realistic replacement.

In the preliminary experiment [74], a GAN was researched and developed to generate synthetic video frames for capsule endoscopy (pill cam) videos [27]. The GAN architecture experimented for the video generations process has used $3D$ CNN to predict future frames of the videos to extend the available real dataset to improve the dataset. Then, the goal of improving data is to improve the performance of other machine learning algorithms which use extended synthetic videos.

The generative models discussed with the preliminary experiments have given the foundation to build other GANs discussing in this section. However, quantitative and qualitative analyses show that the performance of these preliminary experiments was not enough for solving Sub-objective II by generating synthetic medical data. Still, experiments discussed in studies [72, 73] are contributed to Sub-objective III of this thesis. Therefore, those GAN architectures were excluded from the final DeepSynthBody platform until improving these using future research works.

Another three advanced GAN architectures were investigated with the HyperKvasir dataset after the foundation analysis from preliminary studies [72, 73, 74]. These three studies, namely GI-StyleGAN [71], SinGAN-polyp-augmentation [67], and Polyp-inpainting [75], were conducted as proof of concepts to mainly address Sub-objective IV, which focus on generating synthetic medical data to solve the data deficiency problem in the medical domain. These three studies and corresponding contributions to the sub-objectives are
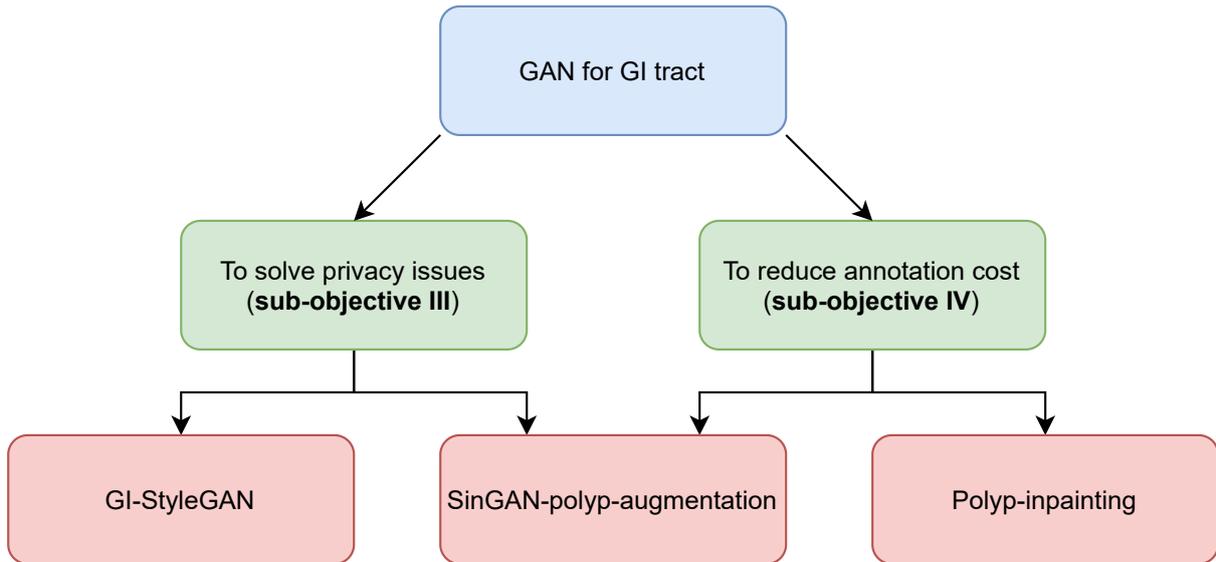
Figure 3.12: Different type of GANs for generating synthetic GI-tract findings for different purpose.

depicted in Figure 3.12.

The GI-StyleGAN experiment presented in the concept paper of this thesis [71] used StyleGAN-v2 introduced by Karras et al. [194] with the unlabelled data folder of the HyperKvasir dataset to generate synthetic GI-tract images. The main objective of this experiment was to achieve Sub-objective II and IV, which are collecting and developing medical datasets and researching and developing GANs to generate synthetic data. All the unlabelled images (around $100,000$) from HyperKvasir were used to train the StyleGAN-v2 model because the model is prone to a large training dataset. Pytorch implementation of StyleGAN-v2[5] was trained $10,000,000$ steps for more than eight days to get good output. In this training process, checkpoints were saved after every $1000,000$ steps (not using epochs) to check the progress of the quality of generated GI images and Frechet inception distance (FID) values introduced by Heusel et al. [195] were calculated to find the best checkpoint. The calculated FID values from different feature layers, namely 64: first max-pooling features, 192: second max-pooling features, 768: preaux classifier features, and 2048: final average pooling features, are tabulated in Table 3.7. Randomly picked synthetic colon images are presented in Figure 3.13. The presented images show that the StyleGAN implementation is capable of generating realistic synthetic colon images. This colon StyleGAN is not only for generating random images, but it can generate interpolated images between two randomly generated images, as depicted in Figure 3.14. This

---

[5]`https://github.com/lucidrains/stylegan2-pytorch`

Table 3.7: FID scores calculated from different checkpoints of StyleGAN trained for generating GI-tract findings.

chk_point	FID_64	FID_192	FID_768	FID_2048
0	39.1090	189.4938	2.6159	342.0751
100	1.7710	8.3480	0.3030	58.9490
200	1.6616	8.0271	0.2977	59.7215
300	1.6575	7.8310	0.2671	52.6597
400	1.2801	6.1183	0.2429	48.5694
500	1.2262	5.8759	0.2372	49.3512
600	1.5974	7.4586	0.2626	52.9441
700	1.3826	6.5063	0.2363	46.2668
800	1.1938	5.9112	0.2312	46.7931
900	**0.6537**	**3.0260**	0.2017	44.3310
1000	0.8736	4.2926	**0.1980**	**41.2039**

Figure 3.13: Style-GAN generated random gastrointestinal-tract findings.

functionality introduced in the vanilla implementation of StyleGAN [196] can generate synthetic data as needed for end-users. This particular generative model can be used to achieve the **sub-objective II and IV**, aiming to develop medical datasets and solve privacy concerns by generating synthetic medical data.

Generating synthetic data with corresponding ground truth is challenging than generating random synthetic data samples solely. However, generating both synthetic data and ground truth is essential to overcome the data deficiency problem to achieve sub-objectives II and IV. We can use synthetic data to replace the costly and time-consuming medical data annotation process, which is identified as one of the reasons causing the data deficiency problem. We can generate both synthetic data and the corresponding ground

Figure 3.14: First five samples generated with 200 interpolation steps for two different random seeds. First and second row represent the two different random seeds. [71]

truth using GANs to solve the problem.

The polyp inpainting GAN [75], capable of generating synthetic polyps on clean colon images, is another study performed with GI data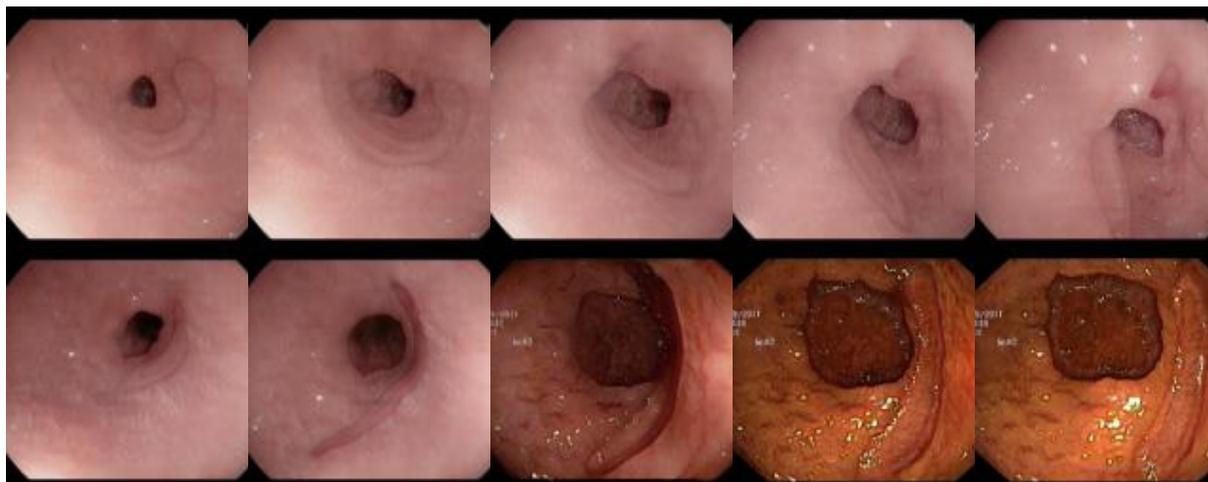. This gan was researched and developed as the first solution to overcome the data annotation problem, as presented using the third leaf node of Figure 3.12. In this experiment, image inpainting using generative multi-column CNN presented by Wang et al. [197] was studied, researched, and developed to do polyps inpainting for non-polyp images using given masks that represent regions of interest to have polyps. However, the available polyp data in the polyp datasets are not enough to train the GAN from scratch. Therefore, the inpainting GAN model was trained from clean colon image folders as the first step. The clean colon image folders have enough images identified as non-disease images by experts to train a DL model. After training with the clean colon data, the model was retrained using polyp data and corresponding masks using the transfer learning mechanism [198] to generate polyps on clean colon images for given masks. This training process is illustrated in Figure 3.15 according to the steps discussed in the original publication [75].

After the training process, the polyp inpainting GAN can convert clean colon images into corresponding polyp images using given masks. Therefore, this inpainting GAN can generate synthetic polyp datasets with the masks of the polyp regions. Then, the inpainting GAN can be used as a solution to achieve Sub-objective IV by producing synthetic data as alternatives to the resource-consuming medical data annotation process. The inpainting GAN can generate synthetic polyps for given random polyp masks without any aid from experts. Therefore, we can use this type of GANs to generate synthetic true
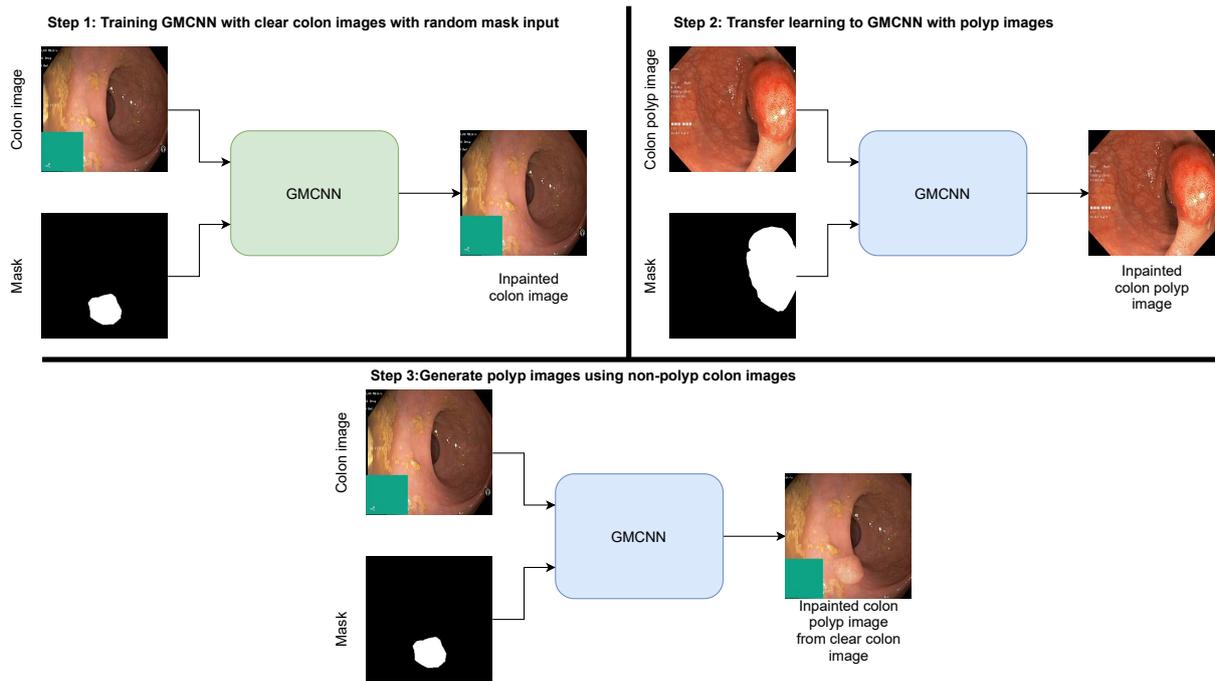
Figure 3.15: Steps of the polyp inpainting training process as discussed in [75]. Generative multi-column convolutional neural networks (GMCNN) [197] is the core network in this process.

positive data from true negative data, which are common and easy to find. We showed that synthetic polyps show visual properties also indistinguishable from real samples for the domain experts.

A qualitative analysis was done using a survey with medical experts to evaluate the quality of the synthetic polyps generated from the polyp inpainting GAN. Using the polyp inpainting GAN, synthetic polyps were generated and analyzed by domain experts. The experts analyzed five synthetic and five real polyps samples. The samples used for a questionnaire are presented in Figure 3.16. In this questionnaire, experts were asked to discriminate synthetic polyps from real polyps and give a confidence score for the particular selection. Two experts, three non-experts and three internal medicine residents (total is eight) have participated in this questionnaire. The summary of the results collected from this questionnaire is presented in Table 3.8. Finally, the proposed GAN architecture can generate synthetic polyp image conditioned on a clean colon image and a random mask representing a polyp region. The polyp inpainting GAN shows that modified GAN architectures can generate synthetic data with corresponding masks, usually prepared by experts manually, which is a costly and time-consuming task. More details about this polyp inpainting GAN can be found in our original paper [75]. However, this inpainting
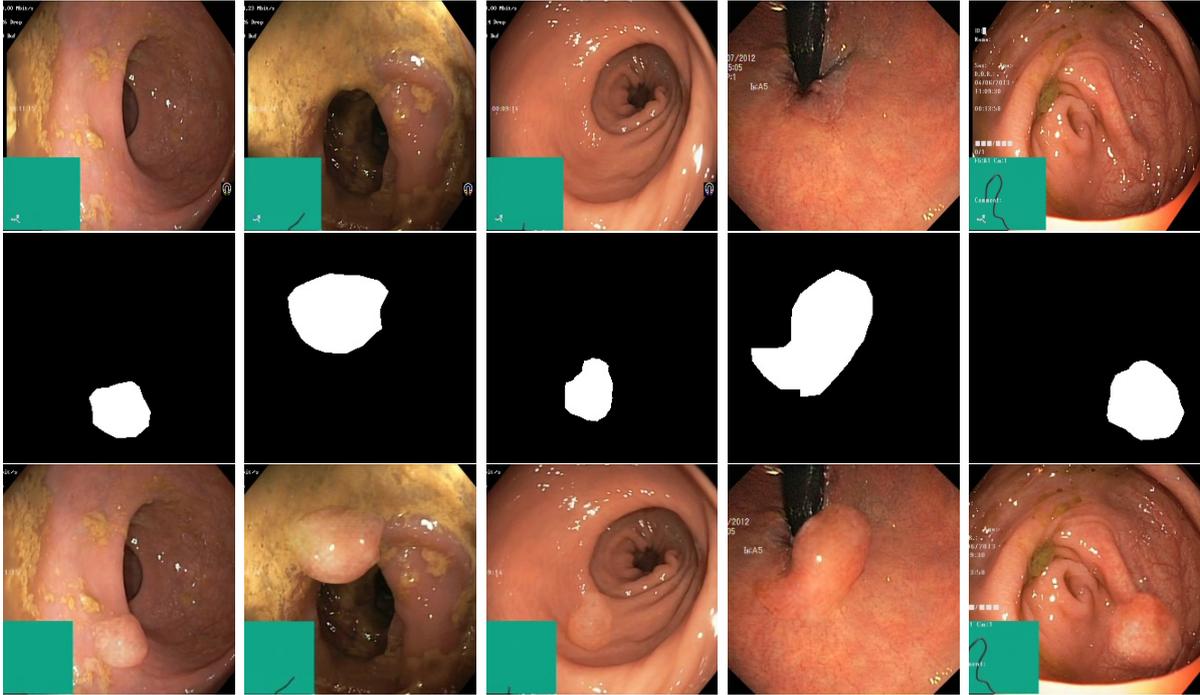
Figure 3.16: Polyp inpainted samples from polyp inpainting gan. The first row illustrates input images. The images in the second row represent input masks used with input images. The third row represents the output images from the polyp inpanting GAN.

Table 3.8: Overview of obtained results from all 8 readers (2 experts - EE and 3 non-experts - NE, 3 internal medicine residents - IM) for discriminating real and inpainted polyps.

Reader	TP	FN	FP	TN	Accuracy
EE1	3	4	2	1	0.4
EE2	3	5	2	0	0.3
NE1	2	1	3	4	0.6
NE2	3	2	2	3	0.6
NE3	4	2	1	3	0.7
IM1	4	2	1	3	0.7
IM2	4	3	1	2	0.6
IM3	4	1	1	4	0.8

GAN is not suitable for a privacy-preserving data sharing technique because the non-polyp regions are identical to the real clean colon images.

SinGAN-Seg [67] was investigated in this thesis to achieve sub-objectives III and IV. The SinGAN-Seg implementation was inspired by the original SinGAN introduced by Rott Shaham, Dekel, and Michaeli [149]. The vanilla SinGAN learns from a single image and generates synthetic samples similar to the pixel distribution of the image used to train it. The original paper presents different applications such as paint to image,

Figure 3.17: A representation of the four-channels SinGAN training step.

super-resolution, editing images, harmonization, and generating animations using a single image. In our SinGAN study [67], the original SinGAN was changed to input four channels containing the input image and its ground truth mask. Then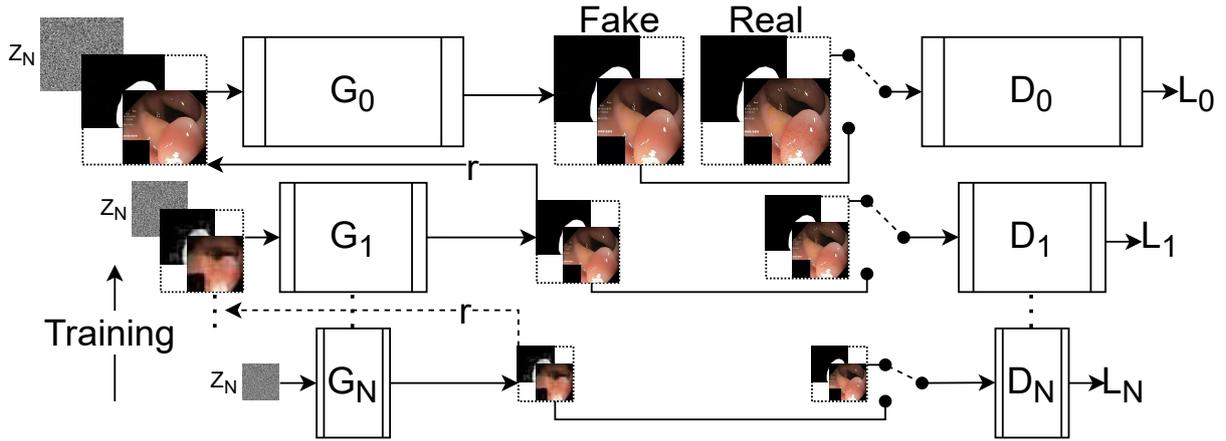, the modified SinGAN was named SinGAN-Seg because it has a generated synthetic image and its ground truth mask (segmentation mask). So, SinGAN-Seg is a modified version of SinGAN to perform the novel application that generates random images and the corresponding segmentation masks. This SinGAN-Seg was introduced in this thesis to address the sub-objectives III and IV. The complete training process of SinGAN-Seg is depicted in Figure 3.17.

The SinGAN-Seg architectures were trained using the 1000 polyps images of the HyperKvasir dataset. Then, 1000 different checkpoints were generated to replace the 1000 polyp images to demonstrate the capabilities of novel sinGAN-Seg to solve privacy concerns and resource-consuming medical data augmentation process. Synthetic polyp images and corresponding ground truth masks generated automatically using SinGAN-Seg are depicted in Figure 3.18. The first column of the figure presents real images and corresponding masks of polyp regions, annotated by experts manually. Other columns present generated synthetic polyps and generated masks from SinGAN-Seg learned from the input image of the first column. While the training data consists of only polyp images, SinGAN-Seg can generate non-polyp images as presented in the 3rd and 4th rows in Figure 3.18. This novel SinGAN-Seg implementation contributed to sub-objectives I, II, III, and IV by presenting a well-performing polyp segmentation model, generating realistic synthetic polyps and corresponding ground truth masks to replace private medical data, and tackling costly and time-consuming medical data annotation process.
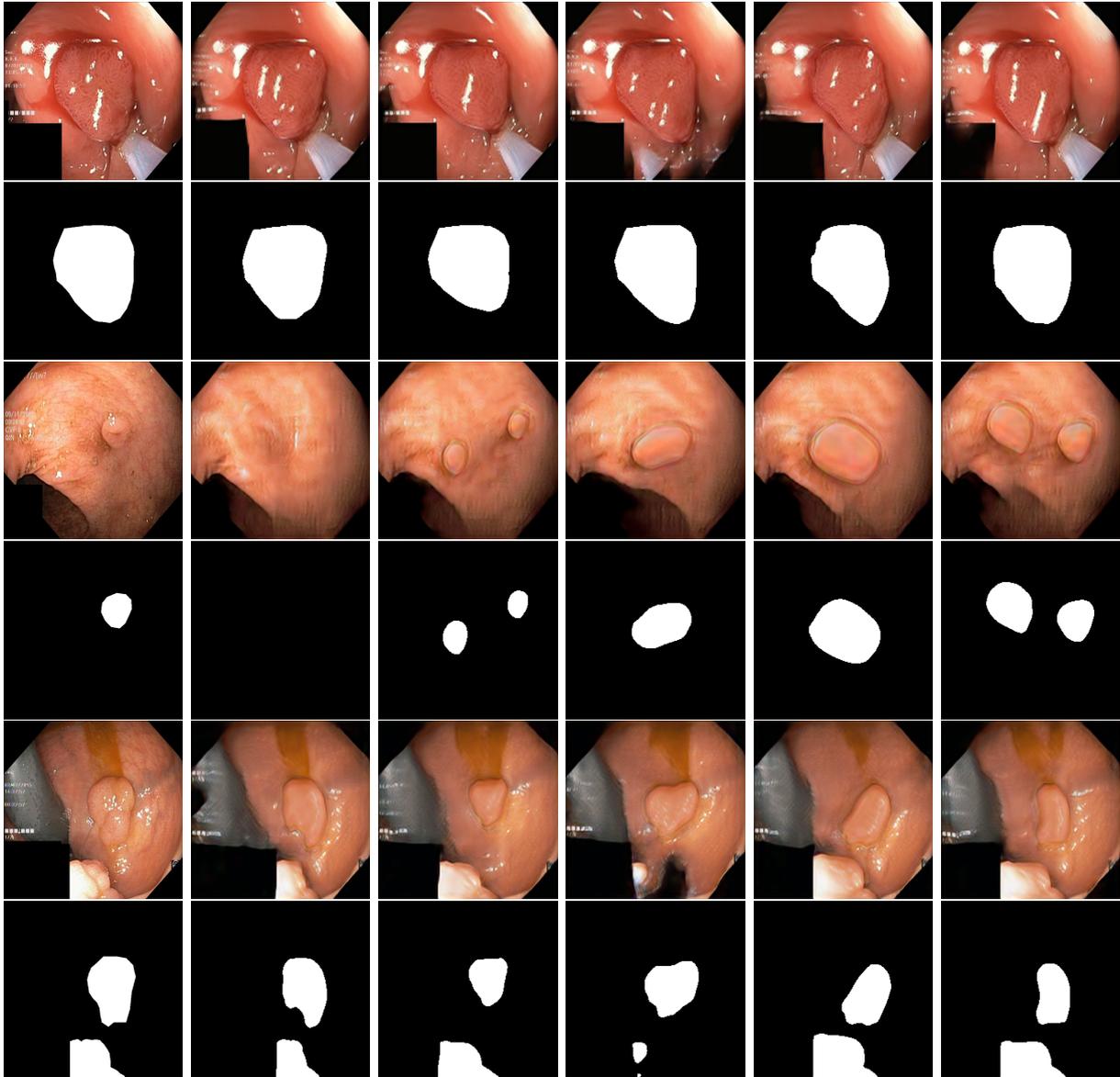
Figure 3.18: Sample real images and corresponding SinGAN generated synthetic GI-tract images with corresponding masks. The first column is illustrated with real images and masks. All other columns represent randomly generated synthetic data from SinGANs which were trained from the image on the first column.

After generating synthetic polyp images and corresponding maks using our SinGAN-seg, the global features of the synthetic images look awkward because of the unrealistic texture of synthetic images (see Figure 3.18). As a solution to this, the style-transfer algorithm [199] was used to transfer styles from the training image to generated synthetic images. More details about this style-transfer method can be seen in our paper [67].

In summary, we could generate realistic synthetic GI tract images in three ways. The StyleGAN model can generate random synthetic GI-tract landmarks that are indistinguishable from real samples. The polyp inpainting GAN can generate synthetic polyp images by converting a true-negative sample into a true positive sample. The qualitative analysis shows that domain experts also cannot differentiate between real and synthetic samples generated from this polyp inpainting GAN. Synthetic data generated from polyp inpainting GAN addresses data imbalance problems. SinGAN-Seg is another GAN architecture that is capable of generating synthetic polyps and ground truth masks. This GAN can be used to overcome the costly and time-consuming medical image annotation process, which experts usually do.

Generating Synthetic Sperm Video

Synthetic sperm data generation is another area considered as a case study. However, limited data and time constraints were barriers to producing successful GAN architectures that can be plugged in to the DeepSynthBody framework. However, we performed several experiments using SinGAN to generate painting-like sperm video frames [76] to represent the real sperm data because SinGAN learns from a single image it does not need large datasets.

We used vanilla SinGAN [149] to experiment with the sperm dataset to perform unsupervised sperm segmentation to achieve Sub-objective IV. In this case, the data deficiency problem will be solved by reducing the annotation cost of medical data. In this task, SinGAN was used to track the locations of sperms in an unsupervised way. The complement operation of the paint-to-image operation introduced in the original SinGAN, image-to-paint, was investigated to generate sperm sample-like paintings to represent sperm locations with a clear background. To achieve this, the SinGAN model was trained from a sperm-like picture. Sample training images investigated to train SinGANs are depicted in Figure 3.19. Then, video frames were input into the pre-trained SinGAN using different
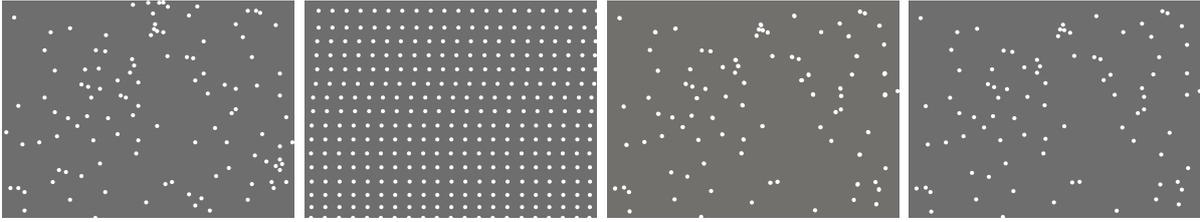
Figure 3.19: Sperm like paintings used to train SinGANs to generate sperm tracking. The last two images have same dot patterns except the background colour.

scale levels as introduced in the original SinGAN implementation. Results were analyzed qualitatively with different input scales. Generated sperm-like paintings from real sperm images can be used to identify sperm locations using this method.

Sample synthetic sperm paintings to represent real sperm sample images are depicted in Figure 3.20. However, the quality of synthetic sperm video frames generated from our SinGAN is not enough for publishing in DeepSynthBody. The results implies that future experiments are required with different GAN architectures and high-quality sperm datasets. A successful GAN architecture to produce sperm like painting can be used to overcome the Sub-objective IV because synthetic sperm like painting can be used for sharing data when privacy concerns are there and, the synthetic sperm like painting is an alternative representation for real sperm video frames which are hard to analyze by experts.
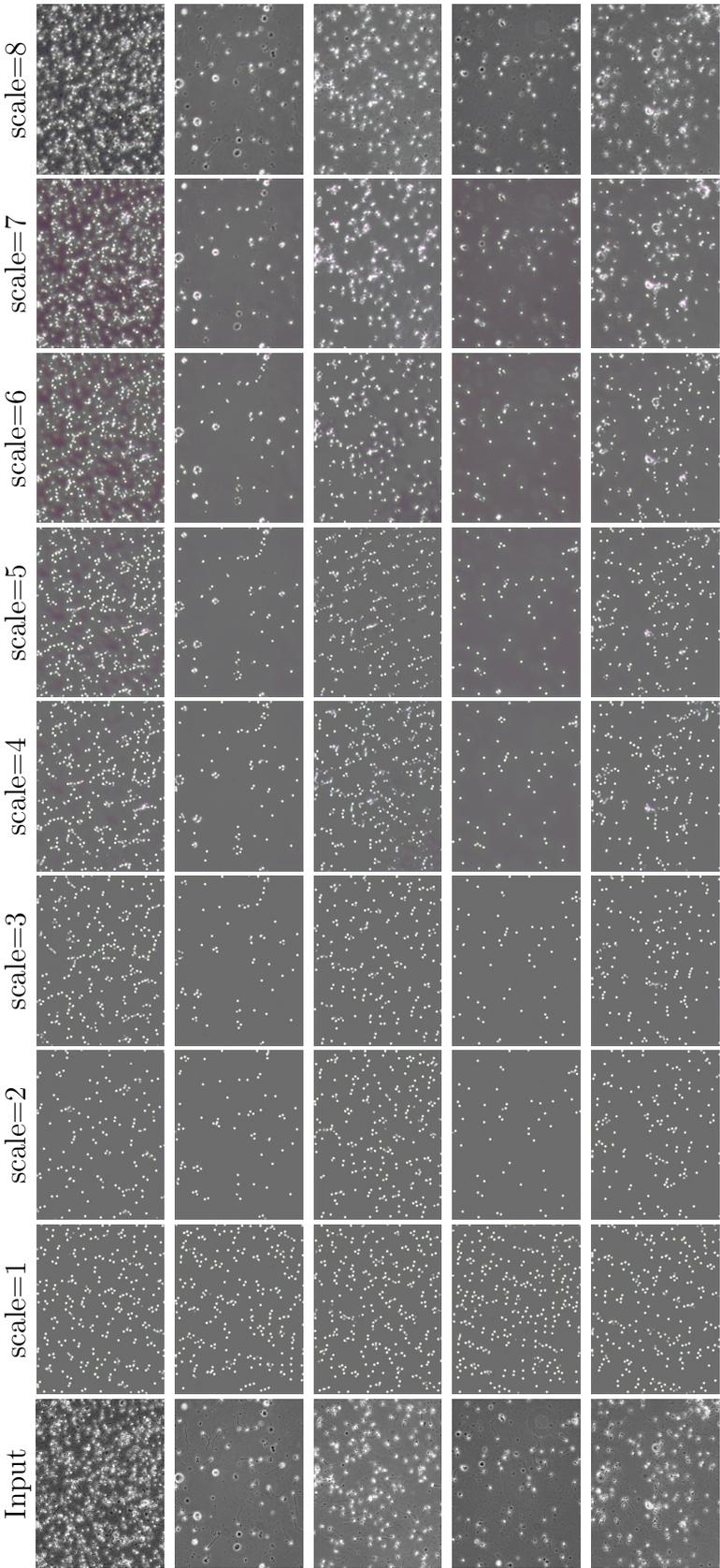
Figure 3.20: Predicted sperm lacations using the SinGAN trained with synthetic sperm samples

In summary, we developed a GAN architecture, based on the SinGAN architecture to generate synthetic sperm data to replace real data. We have generated painting-like sperm images that can measure the quality of the sperm sample. Moreover, this GAN could tract sperm locations using white dots in an unsupervised way. Therefore, usual image processing techniques (without DL) can be used to analyze the sperm samples easily.

3.2.2 Publishing Deep Generative Models

After researching and developing GANs which can generate synthetic data to overcome privacy issues and the costly and time-consuming medical data annotation process, these deep generative models should be published to the end-users. Therefore, the contributors who are developing GANs in DeepSynthBody should have a common platform to share them. As a platform to share the final GAN models with the end-user in this initial stage, the PyPI was selected. Therefore, all the developments were done in the most popular programming language, Python [200], because PyPI is for Python.

> The joy of coding Python should be in seeing short, concise, readable classes that express a lot of action in a small amount of clear code, not in reams of trivial code that bores the reader to death. – Guido van Rossum (creator of Python)

First, the contributors who develop GANs can publish their work as an individual package in PyPI. Then, the PyPI package can be included as a sub-module in the main PyPI called `deepsynthbody`. In cases where PyPI does not work, authors of GAN models, which will be connected with our framework, can share the checkpoints of their deep generative models with corresponding source codes with the main contributors of the framework. If any of these options do not work, researchers can publish only synthetic data in any public data repository, and the corresponding links can be connected to the DeepSynthBody. However, in the latter case, the end-users cannot control the synthetic data generation process.

The flow of PyPI packages is depicted in Figure 3.21. The figure shows how individual PyPI packages are contributing to the main Python package, `deepsynthbody`. First, GAN developers should produce python packages for individual GAN trained for a specific real
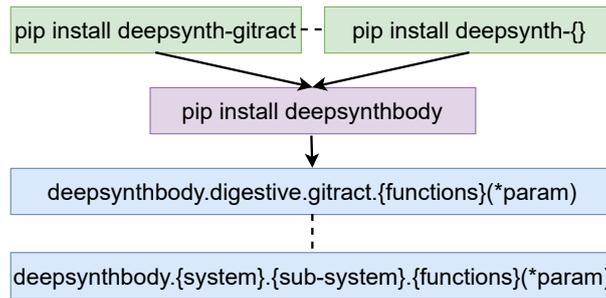
Figure 3.21: The flow of python packages which act as sub-modules of DeepSynthBody framework. The figure reference: [71]

dataset. After training a GAN to produce realistic synthetic data, the GAN can be used as a replacement to the real dataset used to train the GAN. Then, a python package with functionalities to generate synthetic data and the best checkpoints of the GAN model should be packaged into a python package independently. This individual independent python package development process was introduced to reduce the development overhead of the main python package. Finally, these individual packages are connected to the main `deepsynthbody` package according to the human body categorization introduced in Step IV of the framework (see Figure 3.1).

As a proof of concept, two Python packages were developed following the above criteria. First, a python package named deepfake-ecg[6] (`pip install deepfake-ecg`) was published to generate synthetic data from the best checkpoint of the pre-trained Pulse2Pulse [70] ECG GAN. Second, for generating synthetic GI-tract images using the StyleGAN implementation introduced in the paper [71], a python package called `deepsynth-gitract`[7] was published. These packages were developed independently from the `deepsynthbody` package. After publishing the individual packages, they have been connected to the `deepsynthbody`[8] main package.

3.2.3 A Tool to Experiment with Generative Adversarial Networks: GANEx

The DeepSynthBody framework should interact with medical data providers to collect deep generative models,. However, the main challenge is all medical data providers do not have ML programmers who can perform GAN experiments to produce generative models.

---

[6] https://pypi.org/project/deepfake-ecg/
[7] https://pypi.org/project/deepsynth-gitract/
[8] https://pypi.org/project/deepsynthbody/
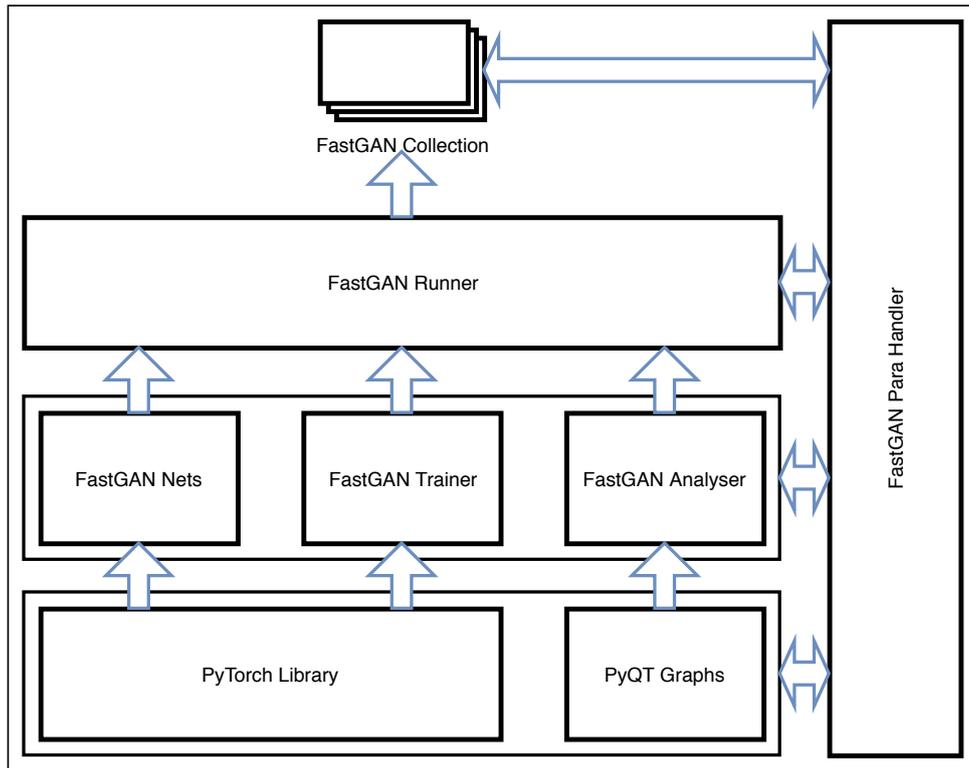
Figure 3.22: The FastGAN library [77] introduced to connect multi-disciplinary user to DeepSynthBody framework.

Additionally, data providers may not have the authority to share the data with intermediate partners to develop GANs. In this context, GANEx (GAN Experimenter) [77] is a tool introduced in this thesis to overcome the barrier of performing GAN experiments by one who does not have a deep understanding of ML or DL. This tool makes a bridge between DeepSyntBody and multi-disciplinary medical data providers.

GANEx consists of two main components: a FastGAN library and a GUI. The Fast-GAN library is a high-level GAN library, which provides functionalities to create pre-defined GANs, train GANs and analyze them through a high-end abstract layer called FastGAN Runner, as depicted in Figure 3.22. Using this FastGAN library as the backend, the GUI has been developed to interact with the backend. The GUI of GANEx provides functionalities to create GAN projects, experiments using a predefined collection of GANs provided from the FastGAN library. Then, using the same GUI, users can run and analyze series of GANs using their datasets without writing a single line of code. The whole process of the GUI is illustrated as a flow diagram in Figure 3.23. After completing the GAN training process, the users have GAN checkpoints, which can be shared to generate synthetic data without any privacy concerns.
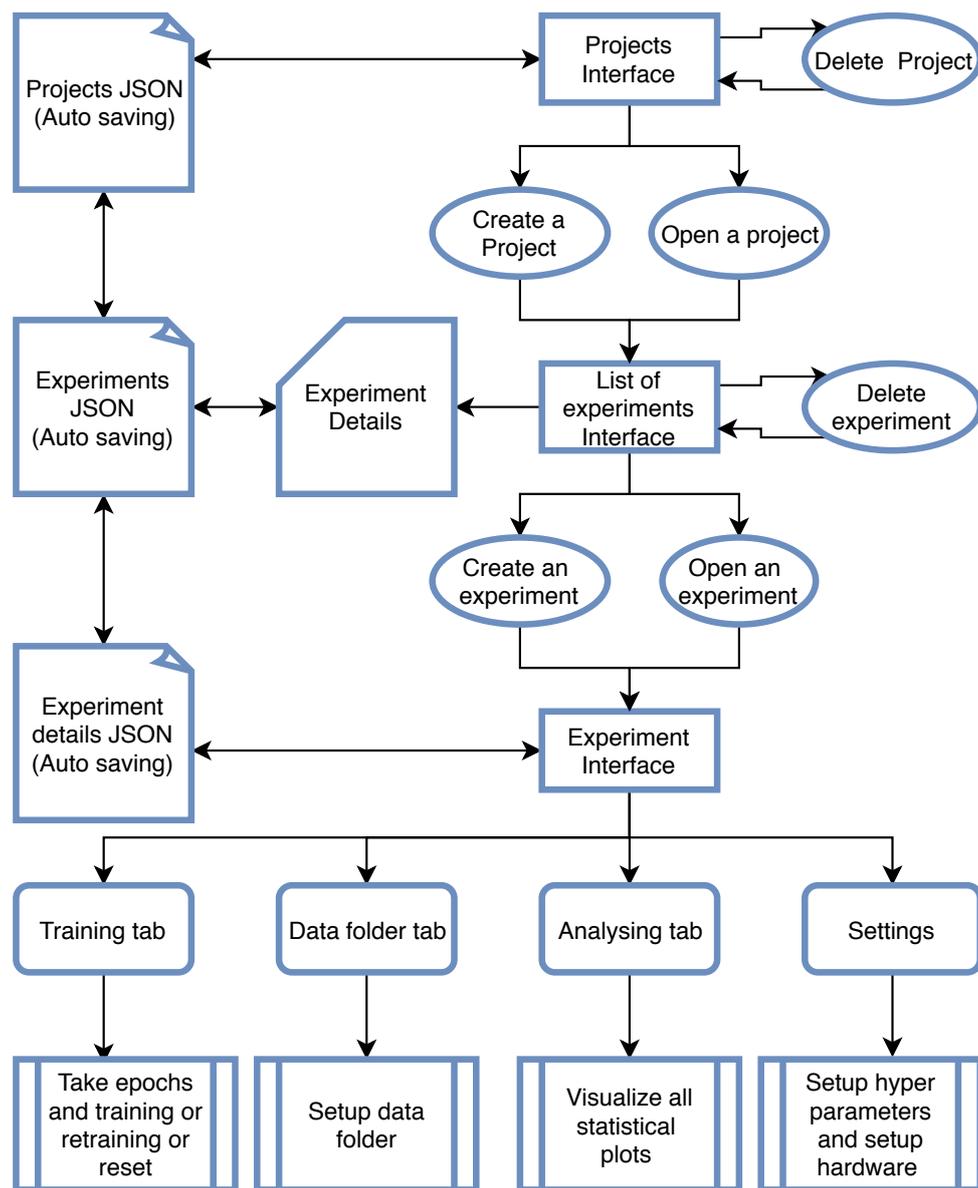
Figure 3.23: GUI flow of GANEx which is a tool to handle GAN experiments for non computer science users of DeepSynthBody.

The sample screenshots of the tool are presented in Figure 3.24. Training progress, a setting page of hyperparameters, and generated sample synthetic data from the CelebA dataset [201] are given in the figure. The given screenshots show how the tool manages every GAN training step without programming (coding). GANEx was developed as a supporting tool to achieve the main objective, which focusing on combing all sub-objectives together to make the functional full framework DeepSynthBody. In addition to this GAN tool, `deepsynthbody.org` is hosted as the main website to achieve the main objective. The website is for both contributors and end-users of the DeepSynthBody framework.
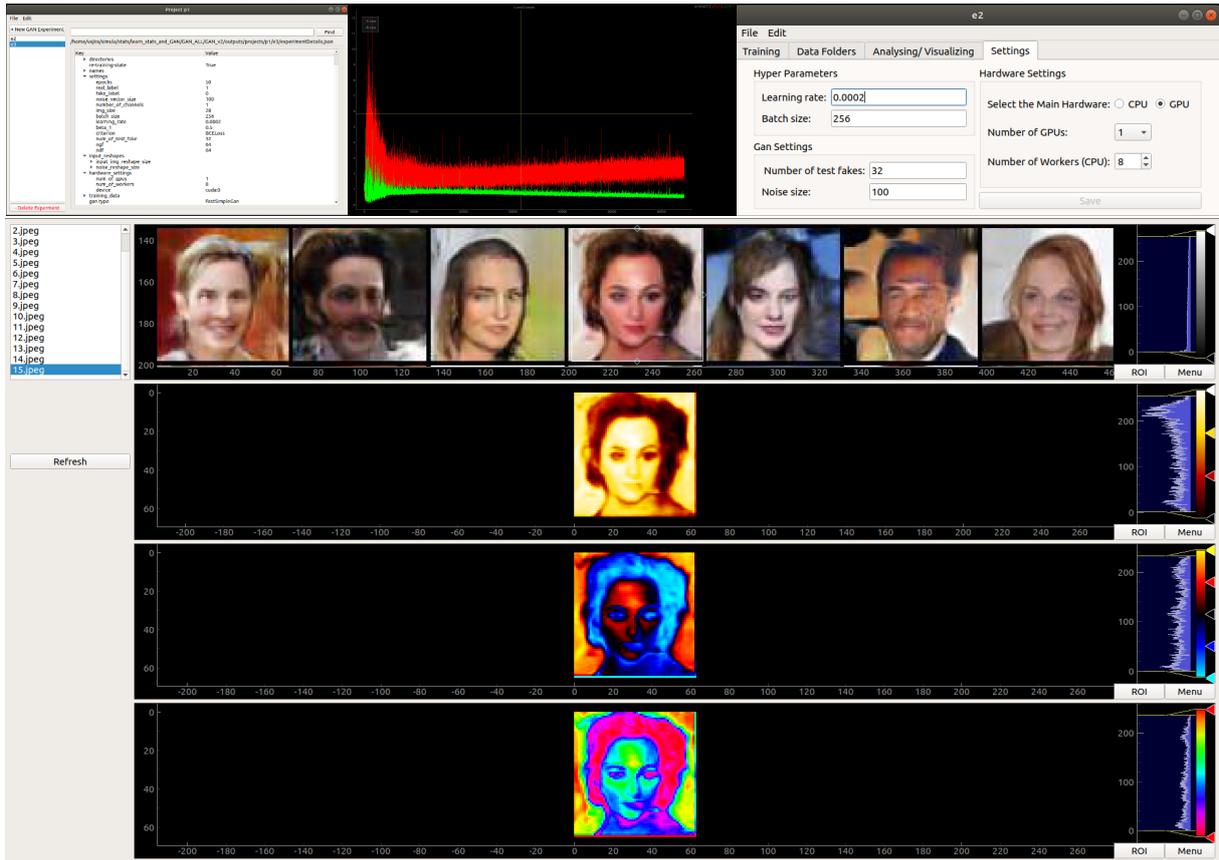
Figure 3.24: Sample screenshots of the GANEx GUI showing user friendly GUI design which can be handled by non computer science multi-disciplinary people. **Top-left**: is showing a screenshot of GAN project management window which shows the summary of all experiments saved in GANEx. **Top-middle**: is showing a screen shot taken from real-time analysis of a GAN experiment using generator loss and discriminator loss. **Top-right**: is showing the window of GANEx which gives functionalities to users to change configuraions of GANs. **Bottom**: is showing GAN generated sample analyser which has functionalities to produce histogram and heat maps of images.

3.3   Step III: Producing DeepSynth Data

Producing DeepSynth data in Step III is presented in the big picture of DeepSynthBody in Figure 3.1. In other words, this is the layer for the end-users who want to generate synthetic data. This Step III has a flow similar to Step I, but the objectives are slightly different. In Step I, the categorization is used to classify input data, while Step III uses the same categorization to generate synthetic data. The data annotation layer presented in Step I was replaced with two new data generation processes: unconditional and conditional. As the final layer of Step III, synthetic data generation functionalities are used instead of the real data analysis in Step I.

We use the same 11 categories in Step III as used in Step I to generate synthetic data for the end-users. Step III is the output layer of the DeepSynthBody pipeline. We further split the 11 categories into four categories based on the data dimensionality (1-D, 2-D, 3-D, and N-D) as discussed in Section 3.1.1. The data dimension layer decides the data output format when there are multiple data formats to generate synthetic data. For example, MRI data can be generated as images (2-D) or volume data (3-D) if both formats are available at DeepSynthBody. In addition, the end-users can decide that the generation process is either unconditional or conditional if both options are available. Several generative models can exist in this framework for a specific generative task (e.g., two different conditional GAN models to generate synthetic ECGs). If more than one model exists, the end-users can choose one for their specific application based on the benchmark reports or using their own qualitative and quantitative comparisons. Similarly, multiple GANs can be used together to generate diverse data distributions because different GAN models may have different data distributions based on the training data used to train them. The website named `deepyynthbody.org` is an online platform for providing all the information about functionalities and their usage to the end-users of the DeepSynthBody concept [71].

The website `deepsynthbody.org` links the researchers and the end-users. The main purpose of this online platform is to connect everything to achieve the main objective. Sample screenshots of the current website are given in Figure 3.25. This site provides the necessary information to contributing to DeepSynthBody and the end-users of this concept. However, the content of this site is subject to change based on new contributions and user experiences. Like the contents, the functional flow of the site is also subject to
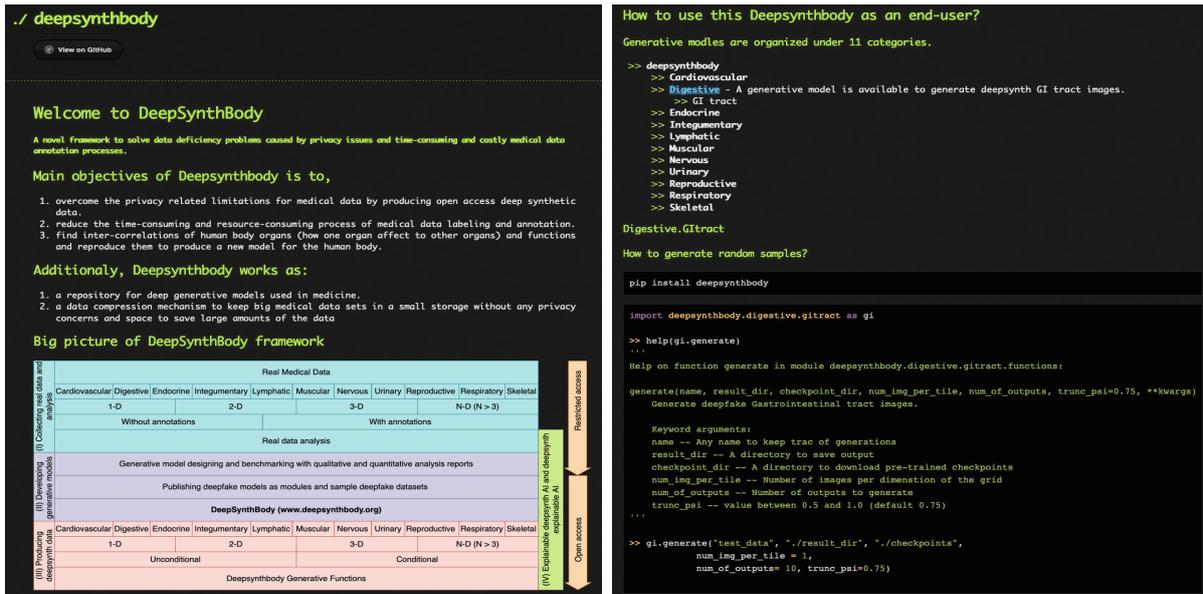
Figure 3.25: Sample screenshots of **deepsynthbody.org**.

```
import deepsynthbody
deepsynthbody.cardiovascular.ecg.generate
            ("number of ECG to generate",
             "Path to generate",
             "start file ids from this number",
             "device to run")
```

Listing 1: The generative function to generate synthetic ECGs that are 10s long and having 8-leads.

change to give better user experiences in the future. At the moment, two functionalities to generate synthetic data are presented on the website. One is for generating synthetic ECGs, and others for generating synthetic GI-tract data.

Abstract functions to generate synthetic ECGs were implemented as presented in Listing 1. Using this generation function, the end-users of DeepSynthBody can generate an unlimited number of 8-leads 10-sec long ECGs, which are convertible to 12-leads ECGs. However, this ECG generative model does not generate ground truth properties such as *PR interval, QT interval, heart rate*, and other properties discussed in the ECG analysis paper [41]. Suppose the end-users are interested in pre-analyzed ECGs. In that case, the generated ECGs can be analyzed using the MUSE system or the pre-generated dataset from the best checkpoint of Pulse2Pulse, and the corresponding MUSE analysis report can be downloaded here: `https://osf.io/6hved/` as presented in our ECG GAN paper [70].

Similarly, the end-users can generate an unlimited number of GI-tract images us-

```python
import deepsynthbody.digestive.gitract as gi

>> help(gi.generate)
'''
Help on function generate in module
    deepsynthbody.digestive.gitract.functions:

generate(name, result_dir, checkpoint_dir, num_img_per_tile,
        num_of_outputs, trunc_psi=0.75, **kwargs)
    Generate deepfake Gastrointestinal tract images.

    Keyword arguments:
    name -- Any name to keep trac of generations
    result_dir -- A directory to save output
    checkpoint_dir -- A directory to download pre-trained checkpoints
    num_img_per_tile -- Number of images per dimenstion of the grid
    num_of_outputs -- Number of outputs to generate
    trunc_psi -- value between 0.5 and 1.0 (default 0.75)
'''

>> gi.generate("test_data", "./result_dir", "./checkpoints",
            num_img_per_tile = 1,
            num_of_outputs= 10, trunc_psi=0.75)
```

Listing 2: Random GI-tract image generation function using StyleGAN.

ing the function presented in Listing 2. In addition to the main generation function, an additional generation function, originally discussed in the vanilla implementation of StyleGANv2 [194], was presented to generate intermediate generations using interpolations between two random points of synthetic generations. This function is presented in Listing 3.

3.4   Step IV: Explainable DeepSynth AI and Deep-Synth Explainable AI

Step IV in the framework, namely explainable DeepSynth AI and DeepSynth XAI, is introduced to embed explainability and transparency into all other layers. This layer covers an essential concept to explain our deep generative models to increase trust and enable deeper failure analysis. Additionally, it allows another way to explain other ML methods using synthetic examples when the data restrictions are applied with real medical

```python
import deepsynthbody.digestive.gitract as gi
>> help(gi.generate_interpolation)
'''
Help on function generate_interpolation in module
    deepsynthbody.digestive.gitract.functions:

generate_interpolation(name, result_dir, checkpoint_dir,
    num_img_per_tile, num_of_outputs, num_of_steps_to_interpolate,
    save_frames, trunc_psi=0.75, **kwargs)

    Generate deepfake Gastrointestinal tract images.

    Keyword arguments:
    name -- Any name to keep trac of generations
    result_dir -- A directory to save output
    checkpoint_dir -- A directory to download pre-trained checkpoints
    num_img_per_tile -- Number of images per dimenstion of the grid
    num_of_outputs -- Number of outputs to generate
    num_of_steps_to_interpolate -- Number of step between
                    two random points
    save_frames -- True if you want frame by frame,
                    otherwise .gif will be generated
    trunc_psi -- value between 0.5 and 1.0 (default 0.75)
'''

>> gi.generate_interpolation("test_data", "./result_dir",
                        "./checkpoints",
                        num_img_per_tile=1,
                        num_of_outputs=1,
                        save_frames=True,
                        num_of_steps_to_interpolate=100,seed=100)
```

Listing 3: The interpolation function to generate random images between two points of generation.

data.

If additional explanations are available to explain the synthetic data generation process before using the synthetic data to replace real medical data, the trust of the end-users to use synthetic data can be improved. Therefore, DeepSynthBody introduces eXplainable DeepSynth Artificial Intelligence (XSAI). XSAI's primary goal is to explain deep generative models [202, 203] to increase understanding of the generative process and the quality of the generated data.

On the other hand, in the medical domain, XAI should be applied in increasing trust to accept solutions from ML models that generally perform classification, detection, and segmentation. While XSAI discusses the explainability of generative models, deep synthetic data can be used to support the XAI of other ML models. This functionality is discussed under DeepSynth XAI (SXAI). In this context, the main goal is not to explain the deep generative models but rather to explain other ML models used to classify, detect and segment medical data using synthetic data as examples. This DeepSynth XAI can overcome the privacy issues occurring when real data is used to explain ML models. For example, when researchers cannot explain their ML models by examples because the real data is restricted to share, they can use synthetic examples to explain their models with less concern about the privacy restrictions.

Both XSAI and SXAI concepts are discussed in the theoretical model. However, this explainable layer is a value-added layer to the DeepSynthBody framework. Therefore, Step IV is an optional step, and as a result, the DeepSynthBody framework functions without these XSAI and SXAI implementations. In this regard, we keep these options for future research works.

3.5 Summary

The DeepSynthBody concept was introduced in this thesis as the main solution to the data deficiency problem, which was identified during researching and developing ML models for CAD systems to assist doctors (Sub-objective I). The concept and the corresponding framework were discussed in four steps. These are, collecting real data and analysis, developing generative models, producing deep synthetic data, and explainable DeepSynth AI and DeepSynth explainable AI. In this chapter, these four steps were discussed one by

one with the corresponding contributions.

Medical data is the core of any ML solution. Therefore, we successfully collected and published seven dataset papers [23, 24, 25, 26, 27, 28, 29] to achieve Sub-objective II. Additionally, these datasets are required data to initiate DeepSynthBody. The datasets were classified according to the novel classification protocol introduced using the biological organ classification and the data dimension classification. Since analyzing the four steps with all types of medical data is impractical, an ECG signal dataset, a GI-tract image dataset, and a sperm video dataset were analyzed as case studies.

The ECG dataset is private, and it is not shareable. Therefore, this dataset was used as a case study to show how synthetic data is shared instead of a real dataset, which has privacy restrictions to share. A benchmark experiment was performed to understand the ECG dataset and implemented a novel GAN architecture, Pulse2Pluse, to generate realistic synthetic data. The Pulse2Pulse can generate synthetic 12-leads, 10-sec ECGs as alternative data to represent the restricted ECG data. The results show that synthetic ECGs generated from Pulse2Pulse are preserving the quality of the real dataset.

The GI-tract dataset was used as case studies to implement synthetic image generators to demonstrate synthetic medical image data sharing to avoid privacy concerns and present the capabilities of using synthetic data to solve the costly and time-consuming medical data annotation process. The deepsynth-gi generator using StyleGAN-v2 was implemented to generate synthetic GI-tract data. Additionally, the image inpainting GAN and SinGAN-Seg were demonstrated as solutions to the resource-consuming medical data annotation process.

The sperm dataset was analyzed, and SinGAN was investigated to perform an unsupervised medical video annotation process. The SinGAN functionality of converting paint-to-image was reversed and used as image-to-paint to accomplish this unsupervised sperm localization mechanism to use as another implementation to prove the capability of DeepSynthBody to use as an alternative data provider for the costly and time-consuming medical data annotation process. These sperm analysis experiments are in the early stage. Therefore final version of synthetic sperm generations will not be available at `deepsynthbody.org` until the GAN can produce quality output that can be published for the end-users of DeepSynthBody. Other than this synthetic sperm generator, the end-users of the DeepSynthBody can access both the synthetic ECG generator and the

GI-tract image generator via `deepsynthbody.org`.

Overall, we could generate synthetic data using three different case studies representing other data formats, such as signals, images, and videos. In most cases, we have generated realistic-looking synthetic data that can be replaced for real data with privacy restrictions. Moreover, we showed that GANs could generate synthetic data with ground truths to overcome the costly and time-consuming data annotation process. Furthermore, we presented how to convert true negative data into true positive data using GANs to address the data imbalance problem. Presented qualitative and quantitative analyses imply that synthetic data can overcome the data deficiency problem in the medical domain.

Explainable DeepSynth AI and DeepSynth explainable AI were introduced as an optional step in this framework, and therefore, contributors can decide that they are following this step or not. This functionality was kept for future research. However, adding explainability to generative models used in this framework can improve the trust of the end-users to use the synthetic data.

Chapter 4

Discussion and Conclusion

The main objective of this thesis is to research and develop generalizable, accurate and well-performing ML models which can be used in CAD systems to aid doctors by detecting more anomalies to save lives ultimately. However, we identified that the lack of medical data is a major problem in the current pipeline of applying ML methods in the medical domain. Therefore, we have defined several objectives to find a way to overcome the data deficiency problem in applying ML solutions in the medical domain. As a result, we introduced a novel concept and the corresponding framework, DeepSynthBody, to bypass the data deficiency problem.

In this thesis, the main research question stated was "**What are the problems that emerge from data in computer-aided diagnosis systems, and how can these problems be tackled?**". To address the research question, we have researched and analyzed ML models used in CAD systems. To support it, we collected and investigated the real medical datasets, researched and developed benchmark analysis to identify the data problems to be addressed. We could identify that data deficiency is the main problem in the medical domain. This problem has occurred due to privacy concerns, the time-consuming and costly data annotation, and the data imbalance problem in the medical domain. To overcome these problems, we researched and developed a GAN-based concept and a framework to tackle the data deficiency problem in the medical domain, namely DeepSynthBody. In the DeepSynthBody solution, the main focus is to overcome the data deficiency problems using synthetic medical data. We show that synthetic data can overcome the data deficiency problem by omitting privacy concerns, generating synthetic data with ground truth and generating synthetic data to overcome data imbalance problems

by converting true negatives to true positives.

To achieve the main objective, **seven datasets** [23, 24, 25, 26, 27, 28, 29], 12 **benchmark analysis studies and ML models to use with CAD systems** [30, 31, 38, 39, 40, 68, 41, 36, 32, 33, 35, 34] and **eight GAN studies** [72, 73, 77, 74, 70, 67, 75, 76, 71] were published to cover all the sub-objectives and finally achieve the main objective and answer the research question. Some of these papers contribute to multiple objectives, while others contribute to only a single objective. These contribution overlaps are illustrated in Figure 1.5 in Section 1.5.

4.1 Contributions and Discussions

The main focus of our research, in general, is to find generalizable and well-performing ML models, which are the main component of CAD systems to assist doctors, and this thesis address several of the challenges arising in this context. In particular, we have focused on researching ML models for CAD systems with special attention to the challenges medical data scarcity introduces. To accomplish this, Sub-objective I was introduced. However, the data deficiency problem was identified as a significant barrier to achieve the sub-objective I. Therefore, this thesis also introduced sub-objectives II, III, and IV to research and develop medical datasets, research and establish benchmarks to identify the data problems, and research and develop GAN-based frameworks to generate synthetic data as the solution. Sub-objective I and Sub-objective III are overlapped greatly because designing ML models for CAD systems consists of implicit benchmark analysis and vice-versa. Finally, we achieved Sub-objective IV by introducing the novel DeepSynthBody concept and the corresponding framework. Three different medical branches, *gastroenterology*, *andrology*, and *cardiology*, were used as the case studies for sub-objectives I, II, III, and IV:

- **Sub-objective I**: The main focus of this sub-objective is to research and develop well-performing ML models for CAD systems to assist doctors. As case studies, we have selected three branches of medicine. In gastroenterology, images collected from colonoscopies were the main data stream to apply ML algorithms which are the core algorithms in CAD systems. Several classification models [30, 31] and segmentation models [35, 36] were researched and implemented for the gastroenterology branch

under this thesis in different stages of the timeline. Not only using real data, but also synthetic data was used with segmentation models [67] used to predict polyps in GI-tract data. Similarly, ML-based regression models were investigated and developed for the andrology branch [38, 39, 40, 68]. For the cardiology branch, an ML-based ECG analysis system [41] was researched and implemented. Moreover, all the dataset papers [23, 24, 25, 26, 27, 28, 29] introduced ML models as baseline experiments considered as initial models for developing CAD systems.

- **Sub-objective II**: The main task of this sub-objective is to collect and produce medical datasets. Collecting medical data and producing baseline results to understand the data is the first step of developing CAD systems. Therefore, different types of medical datasets [23, 24, 25, 26, 27, 28, 29] representing different types of human body organs were collected and published with the baseline experiments. While all the datasets contribute to the main objective, the GI-tract dataset [23] was selected to use as one of the case studies for other sub-objectives because of the data diversity and a large amount of data. Despite our dataset contributions, two additional datasets were used as the case studies. They are an ECG dataset, which is a private medical dataset representing biomedical signal, and a sperm dataset representing video data. The additional datasets were selected to research and develop ML models for CAD systems in the initial stage. Later, these additional dataset were used to maintain the diversity of the case studies used as proof of concepts.

  From the perspective of DeepSynthBody, which is the solution introduced in this thesis to overcome the data deficiency problem, this data processing step is an in-house step if the datasets are private. In this thesis, one private dataset and two public datasets were used to prove the concept of DeepSynthBody. For further investigating the concept's possibilities, experimenting with new medical data types can be started with public datasets with other data types, which were not covered in this thesis. At the end of the successful implementation of DeepSynthBody, we could introduce synthetic datasets, such as synthetic ECGs, synthetic polyps, and the corresponding ground truth masks, and randomly generated synthetic GI-tract landmarks to support the main objective.

- **Sub-objective III**: The selected datasets were used to design generalizable and

well-performing ML models for CAD systems in our Sub-objective I. However, after identifying the data problems of the ML-based CAD system designing process, we re-analyzed the process of designing ML models as benchmark analysis to investigate the data deficiency problem to be addressed in Sub-objective IV. Under Sub-objective I, different types of ML solutions for CAD systems were investigated under the three different selected medical branches, gastroenterology [30, 31, 36, 32, 33, 35], andrology [38, 39, 40, 68], and cardiology [41]. However, all findings were considered as benchmark articles under new Sub-objective III as well because these studies reflect the real problems associated with the medical data.

A set of benchmark articles for the selected datasets as case studies were published to achieve the benchmark analysis objective (Sub-objective III). While all the datasets should have benchmark analysis results, we chose the same three datasets selected in Sub-objective I, as case studies to achieving Sub-objective III. They are the ECG data, the GI-tract dataset, and the sperm dataset. Then, different types of benchmark analysis experiments done for developing ML models for CAD systems in Sub-objective I with the GI-tract data [30, 31, 36, 32, 33, 35] were re-considered to support this objective. Similarly, the ECG analysis [41] and sperm analysis [38, 39, 40, 68] experiments were investigated again as benchmark analyses for identifying the data-related problems to address experimenting GANs. Without having benchmark analysis, it is not recommended to research GANs under this DeepSynthBody framework because the end-user of the DeepSynthBody framework will not have results to compare the quality of synthetic data coming from this framework in addition to understanding the data-related problems. In these benchmark analyses, we contributed to organizing a competition, namely BioMedia 2020 [68], and participated in a competition, namely EndoCV 2021 [35], to maintain higher standards for the benchmark results. A detailed analysis of GI tract landmark classification was performed within the benchmark analyses to introduce proper generalizable analyses using cross datasets of GI data [31]. As a result of the cross dataset evaluation, we further discussed proper evaluation mechanisms and guidelines for binary classification of medical data in our Medimetrics[1] [33], an open-access tool for fair evaluations among different research findings. Furthermore, the effect of image resolution [32]

---

[1]https://medimetrics.no/medimetrics/

was investigated to show that high-resolution data can improve the performance of ML models using the GI-tract data as the case study.

- **Sub-objective IV**: In this sub-objective, the main purpose is to generate synthetic medical data to overcome privacy-related problems, the time-consuming and costly medical data annotation process, the data bias problem in the medical domain, and the medical data imbalance problem. Before studying synthetic data generation experiments, we investigated possible use cases of GANs with GI-tract data. One study has investigated to preprocessing GI tract images using a GAN [72, 73] to fill green regions of endoscopic images. Another study was performed to predicting blurry pill cam video frames using a GAN [74]. The later GAN experiment shows that the GAN can predict the fifth frame for the given four input frames of a pill cam endoscopic video. These experiments helped us to get a basic understanding of GANs in the medical domain.

  Then, advanced GAN experiments to overcome the privacy issues were researched and developed. The privacy concerns were identified as one of the major issues that caused the data deficiency problem in the medical domain. The private ECG dataset was investigated and successfully published a novel GAN architecture called Pulse2Pulse [70], which can generate synthetic 12-leads 10-seconds long ECGs indistinguishable from real ECGs. Not only this ECG generation GAN, we investigated a GI-tract image generation in the concept paper [71], which introduced the DeepSynthBody concept. The synthetic GI-tract data generator introduced in the concept paper showed how to generate controllable synthetic data as an alternative to real medical image data if the real datasets have privacy concerns.

  Not only privacy concerns, but the results collected from the ECG generation and GI-tract image generation experiments show that synthetic data can represent the real data distributions. Remarkably, the synthetic ECGs clearly show the exact distribution of the properties of the real dataset used to train the Pulse2Pulse GAN. Besides generating the synthetic samples within the distribution, the generated synthetic data can cover untouched regions of the real data distribution. For more information about the distribution overlap between the real and the synthetic data, refer to the original article of Pulse2Pulse [70]. Similar to the synthetic ECGs, the synthetic GI-tract images shows realistic GI-tract landmark within the generated

images. Then, these GANs are an indication that synthetic data can be used to generate uniform data distributions or missing data.

Furthermore, under this objective, to introduce an alternative method for the costly and time-consuming expert's data annotation process, we researched and developed novel pipelines of GAN architectures using two case studies, the GI-tract dataset [23] and the sperm dataset [69]. In one study, the GI-tract dataset was used to train a GAN to generate synthetic polyp data from clean colon images [75]. This study also contributed to the data imbalance problem in the medical domain because the pipeline introduced in this study converts a real clean colon image (true-negative sample) into a synthetic polyp image (true-positive sample). In another study, SinGAN-Seg, synthetic polyp data were generated with the corresponding mask from a single polyp image [67]. In the SinGAN-Seg study, an unlimited number of synthetic samples can be generated with the corresponding segmentation masks of polyps. This GAN can solve the time-consuming and costly data annotation process by generating synthetic data and the corresponding segmentation masks automatically. Moreover, we show that generated synthetic samples can improve the performance of polyp segmentation algorithms used in CAD systems when the manually annotated dataset is small. Additionally, we have investigated the usability of GANs to produce synthetic sperm data [76] instead of blurry-looking sperm video samples to have better quality assessments.

In this thesis, we researched an unsupervised way to segment sperms using a GAN-based model. The results showed promising directions of converting real sperm video frames into synthetic clear video frames with sperm locations, which can be used to analyze the sperm samples in future studies. This sperm study also a proof for using GANs to overcome the time-consuming and costly data annotation process in the medical domain.

In addition to generating synthetic data with segmentation masks representing the most advanced ground truth type, which is pixel-wise classification, all other ground truth generations, such as continuous values, class labels, and bounding boxes, can be explored and considered to overcome the data deficiency problem using GANs as explored under Sub-objective IV. For example, conditional GANs generating synthetic medical data using simple numerical values as input conditions can make

synthetic datasets with numerical ground truth data. Similarly, using class labels as input to GANs can produce synthetic datasets with the corresponding class labels. Moreover, bound box ground truth, one of the famous medical image analysis techniques, can be made using similar conditional GANs.

Finally, we formalized the GAN development process using the novel concept and the framework called DeepSynthBody to overcome the data deficiency problem. In this framework, we pipeline the synthetic data generation process in the medical domain using four steps. Developers who are researching GANs and end-users who need synthetic data can use our framework via `www.deepsynthbody.org`. In this framework, we encouraged to publish generative models as PyPI package instead of publishing pre-generated billions of synthetic data samples. This encouragement is a trade-off because it has advantages and disadvantages. Pre-trained GAN models need less space than publishing pre-generated data is an advantage. If pre-trained GANs are conditional GANs, then the end-users can generate synthetic data as they needed. This custom data generation is another advantage. The main disadvantage of using pre-trained models instead of pre-generated synthetic data is the reproducibility of research works performed using privately generate synthetic datasets. However, publishing the synthetic datasets used to perform the research in other public data repositories can solve this problem. Therefore, overall we recommend publishing pre-generated GAN models instead of pre-generated datasets.

- **Main-objective**: The final objective was to connect all sub-objectives to produce well-performing and more accurate ML models for CAD systems to assist doctors in efficient diagnoses by addressing the data deficiency problem. The initial ML models designed to achieve the Sub-objective I showed the effects of data deficiency problems in the medical domain. Then, we collected, researched, and developed datasets (real and synthetic) for developing ML models for CAD systems for biomedical applications. In Sub-objective III, benchmark analyses were performed to identify the data problem to be addressed using GANs. Then, we proposed DeepSynthBody, which is based on GANs to address the data deficiency problem in the medical domain (Sub-objective IV). Finally, we published our solution as an open-source project for getting more collaborations worldwide at `www.deepsynthbody.org`.

Generated synthetic ECG data show that our concept can avoid the privacy concerns

in the medical domain. We proved the usability of synthetic ECG data qualitatively and quantitatively in our DeepFake ECG paper [70]. Moreover, synthetic polyp generation studies [67, 75] showed that the data imbalance problem and the time-consuming and costly data annotation problem can be solved using synthetic data. Additionally, SinGAN-Seg [67] showed performance improvements when synthetic datasets are used instead of small real datasets. Ultimately, we could show that the main-objective is achievable using the novel concept and the corresponding framework, namely DeepSynthBody, introduced in Sub-objective IV and achieving other three sub-objectives I, II, and III.

By achieving the four sub-objectives, we reached our main objective: research and develop ML models for CAD systems for different medical applications focusing on the problems of limited availability of biomedical data. Finally, we showed that the research question, "**What are the problems that emerge from data in computer-aided diagnosis systems, and how can these problems be tackled?**" could be answered using our novel concept called DeepSynthBody. Now the concept is public. All the necessary infrastructure of the DeepSynthBody framework is ready for contributions from researchers who can provide deep generative models to this framework to make a fully functional open-source DeepSynthBody. This concept will open a new era for open science in the medical domain. For contributions, researchers can visit our online platform: `www.deepsynthbody.org`.

4.2 Ethical Consideration

Medical data collected from one patient, one hospital, one region, or one human race to train and develop ML models used in CAD systems can lead to ethical problems because the models based on this data can make biased predictions. Therefore, researchers should pay more attention to this problem in their research. For example, when an ML model is trained from a patient's data, then the model should consider the patient's anonymity and confidentiality. In this regard, we have maintained all the participants' anonymity and privacy for our data collections by de-identifying data samples, thus, making it impossible to connect the data to real persons. Furthermore, we combined data collected from several hospitals to avoid patient bias problems and hospital bias

problems. However, to avoid the race and country bias problems, a more extensive data collection should be performed. Collecting data in the medical domain is challenging due to privacy concerns, the costly and timely medical data annotation process, and data bias problems. The DeepSynthBody concept can address these problems by omitting anonymization and confidentiality concerns (privacy concerns) by generating synthetic data with ground truths, and generating synthetic data by converting true negatives to true positives.

Although DeepSynthBody is created to solve the data problems in the medical domain, the ethics of synthetic data, which is the core of the concept, is a critical topic. Deep Fakes [204], a popular topic in synthetic data generation, can fool people by generating realistic-looking face images and videos. In this context, Deep Fakes are sometimes used to make fake news about famous people. While some of these Deep Fakes are used to entertain society, others are purposely harming both people and the society.

The same problems may happen with synthetic data in the medical domain. Some possibilities are that someone can generate fake medical reports with generated realistic-looking medical images and videos, etc. People may use these fake reports to cheat their companies to get social benefits such as additional money. This kind of circumstance cannot be avoided, and making a fully secure link with hospitals to get approval can be a solution. Another ethical issue arises with converting true negatives into true positives. Somebody can argue that this is not an ethical procedure because one converts healthy medical data to unhealthy data. However, if true positives are not identified using a real name, we believe that this conversion is ethical.

In sum, we believe that the possible negative effects of synthetic data in the medical domain are outweighed by the positive aspects. We presented in this thesis how to use synthetic data to share private datasets in order to avoid privacy concerns. Furthermore, we showed that synthetic data is a possible solution to overcome the data bias problems. For example, we converted non-polyp images into polyp images. In other words, we converted true negative samples into true positives. A similar mechanism can overcome the data imbalance problems by converting data from one racial background to another racial background to avoid ethical issues related to imbalanced data. Moreover, ground truth data in the medical domain can raise ethical issues due to differences from an expert to another expert who performs the ground truth preparation process. These differences

affect the final performance of the ML models trained from the data. The ML models, in some way, reflect the skills of the person who prepared the ground truth data. In this context, we have proposed a way to generate synthetic data with the corresponding ground truth. Therefore, experts' knowledge can be used to verify the ground truth rather than preparing ground truths which have differences from one expert to another. Overall, we can see that synthetic data in medicine can rather help to solve ethical issues than producing new ones. Nevertheless, like for all research where humans are involved, one needs to be very careful and sensitive in addressing ethical questions for each specific medical application area where synthetic data might be used.

4.3   Future Works

Our solution, DeepSynthBody, which was introduced to tackle the data deficiency problem for developing ML models for CAD systems in the medical domain, can be improved in different ways from Step I to Step IV. In Step I, many datasets from different organ systems have been collected. However, in this thesis, only one dataset from the data collection was investigated with additional two datasets from the outside of our data collection. Therefore, benchmark studies and GAN experiments should be performed with the rest [24, 25, 26, 27, 28, 29] of our data collection. In addition to the collected datasets, other open-access datasets can be used as case studies, such as, MRI datasets representing 4-D datatype, which was not considered in this thesis. In Step II, the evaluation process of benchmark results can be improved by introducing a common guideline to measure the performance of detection and segmentation ML models such as MediMetrics [33], which was introduced to improve the quality of evaluations used with binary classifications.

The GAN models used for the three case studies, which used ECG dataset, GI-tract dataset, and sperm dataset as main data sources, can be further improved. For example, Pulse2Pulse [70] can be enhanced by adding conditional input such as ECG properties. Additionally, continuous ECG pulse generation can be researched with a modified version of the Pulse2Pulse generator by conditioning on the first half of ECGs as input to the generator. Moreover, GI-tract style GAN [71] can be improved using conditional-GANs of GI-tract images to generate specific landmarks of GI-tract. However, the main challenge for training conditional GANs for generating synthetic GI-tract images is a lack of

labeled GI-tract images. In this case, researchers can experiment with transfer learning mechanisms for GAN training [205, 206]. Further investigations with SinGAN-Seg and polyp inpainting GANs can improve the quality of the synthetic data generated from these GANs, i.e., adding super-resolution GAN [207] to the pipeline of synthetic polyp image generation. In the end, we have considered three branches of medicine, cardiology, gastroenterology, and andrology. Other branches of medicine should be considered in future research to build the complete DeepSynthBody, and ML models for CAD systems to assist doctors.

The GANEx [77] tool can be further improved with several functionalities. For GANEx, we can introduce functionalities to publish checkpoints of trained GAN architectures directly into the DeepSynthBody framework. In this functionality, the submitted checkpoint can be reviewed by computer science experts of the future community of the framework before merging them into the final `deepsynthbody` package. Adding these kinds of functionalities can help non-computer science people to publish their GAN modules without any coding burdens. Additionally, integrating interaction between GANEx and the online platform can introduce real-time performance comparisons, such as qualitative comparisons for synthetic images, if there are two or more models for the same purposes. Not only that, Federated learning techniques [208, 209] can be investigated for GANEx to enable distributed GAN learning to input bigger training data distribution to get better realistic synthetic data. However, some re-engineering, such as added web services for distributed computing, can be researched for online interactions and distributed computing.

We believe that DeepSynthBody will open new research directions and overcome the data deficiency problem in medicine. For example, DeepSynthBody can produce a new model for representing the human body and its intra-correlations of functionalities of the organs. These functionalities can be achieved by collecting multi-model datasets consisting of various types of medical data correlated with each other. Suppose we can investigate GAN models, which can condition on one datatype and generate synthetic data on another data type. In that case, those models can be used to find correlations among different medical data types. Finally, GANs can be trained to generate synthetic data conditioned on one organ's data and generate data for another organ system. Successful findings of these correlations can lead to finding correlations about organs' functions

because data coming from organ systems is inherited from their functions. Additionally, this platform will act as a large medical data repository without any privacy concerns and data storage shortages because successful GANs can act as a data compression method. For example, the size of the training dataset used in the Pusle2Pulse [70] implementation is around 3GB for around 15,000 ECGs. However, if we use the `deepfake-ecg` PyPI package, it takes around 50MB to store in cloud platforms. Still, it can generate an unlimited number of realistic synthetic ECGs from a similar distribution of the real data. In this thesis, proper evaluations were not done focusing on this data compression because it was not our main goal. Thus, future studies can be focused on evaluating this privacy-preserving data compression and storage.

4.4 Conclusion

In conclusion, ML-based CAD systems are a great value addition to medicine because these systems have the capabilities to assist doctors by performing automated diagnosis processes. However, we showed that a lack of medical data to train ML models causes generalizability and performance issues. Collecting and processing medical domain data is a basic solution to overcome this problem. However, collecting and processing data is not easy in the medical domain because of privacy restrictions and the costly and time-consuming data annotation process. Generating synthetic medical data to train ML models is an alternative solution to overcome this data deficiency problem.

Well-performing GAN architectures can generate realistic synthetic data. These synthetic data can represent real medical data when the real datasets are not permitted to share. Moreover, conditional GAN architectures can generate synthetic datasets with the corresponding ground truth data, which domain experts normally do. For example, we showed that how to generate synthetic polyps and the corresponding ground truth masks. Furthermore, GANs can generate synthetic medical data by converting true negative data samples into true positive data samples. Data conversion, such as true negatives to true positives, can solve the data imbalance problem in the medical domain.

DeepSynthBody framework, which was introduced as the main solution in this thesis to overcome the data deficiency problem, provides a complete framework to generate synthetic data and develop generative models. We published this concept and the framework

as an open-source project to get contributions worldwide. Getting more contributions, we hope to produce the largest synthetic data repository in the world. Ultimately, this DeepSynthBody concept can be improved to use as a model to represent the human body. Furthermore, the data compression ability of GANs is a solution for storing medical data in a limited space avoiding privacy concerns.

4.5 Final Remarks

In this thesis, we researched and developed ML-based components for CAD systems in three different branches, gastroenterology, andrology, and cardiology. All the data collected under these three branches were collected from hospitals in Norway and Denmark. In most of the cases, datasets were analyzed by experts in the domains. In the cases where we generated synthetic data, domain experts helped us to perform a qualitative analysis with their expertise. Furthermore, our solution proposed in the thesis, namely DeepSynthBody, shows a high potential to be an important part of the future of developing well-performing ML models for developing CAD systems. However, the success of the future directions of DeepSynthBody depends on the contributions from the research community of ML and the medical data providers. Therefore, the framework is available as an open-source project at `deepsynthbody.org` to get more contributions and to the end-users who want to generate synthetic medical data. Moreover, we showed advanced future directions of our DeepSynthBody, such as using the framework as a novel model to the human body and a novel way to store medical data.

Bibliography

[1] David Reinsel–John Gantz–John Rydning. "The digitization of the world from edge to core". In: *Framingham: International Data Corporation* (2018).

[2] Daniel E O'Leary. "AI in accounting, finance and management". In: *Intelligent Systems in Accounting, Finance and Management* 4.3 (1995), pp. 149–153.

[3] Henri Arslanian and Fabrice Fischer. *The Future of Finance: The Impact of Fin-Tech, AI, and Crypto on Financial Services*. Springer, 2019.

[4] Bo-hu Li, Bao-cun Hou, Wen-tao Yu, Xiao-bing Lu, and Chun-wei Yang. "Applications of artificial intelligence in intelligent manufacturing: a review". In: *Frontiers of Information Technology & Electronic Engineering* 18.1 (2017), pp. 86–96.

[5] Cihan H Dagli. *Artificial neural networks for intelligent manufacturing*. Springer Science & Business Media, 2012.

[6] Richard Lachman and Michael Joffe. "Applications of Artificial Intelligence in Media and Entertainment". In: *Analyzing Future Applications of AI, Sensors, and Robotics in Society*. IGI Global, 2021, pp. 201–220.

[7] Sylvia M Chan-Olmsted. "A Review of Artificial Intelligence Adoptions in the Media Industry". In: *International Journal on Media Management* 21.3-4 (2019), pp. 193–215.

[8] Fei-Yue Wang. "Toward a revolution in transportation operations: AI for complex systems". In: *IEEE Intelligent Systems* 23.6 (2008), pp. 8–13.

[9] Adel W Sadek. "Artificial intelligence applications in transportation". In: *Transportation Research Circular* (2007), pp. 1–7.

[10] Fei Wang and Anita Preininger. "AI in health: state of the art, challenges, and future directions". In: *Yearbook of medical informatics* 28.1 (2019), p. 16.

Bibliography

[11] Thomas Davenport and Ravi Kalakota. "The potential for artificial intelligence in healthcare". In: *Future healthcare journal* 6.2 (2019), p. 94.

[12] Susmita Ray. "A quick review of machine learning algorithms". In: *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE. 2019, pp. 35–39.

[13] Rocio Vargas, Amir Mosavi, and Ramon Ruiz. "Deep learning: a review". In: *Advances in intelligent systems and computing* (2017).

[14] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. "Overview and Importance of Data Quality for Machine Learning Tasks". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2020, pp. 3561–3562.

[15] Xue-Wen Chen and Xiaotong Lin. "Big data deep learning: challenges and perspectives". In: *IEEE access* 2 (2014), pp. 514–525.

[16] IBM. *IBM Cloud Learn Hub*. Accessed: 2021-04-25. URL: https://www.ibm.com/cloud/learn/artificial-intelligence.

[17] Kunio Doi. "Computer-aided diagnosis in medical imaging: historical review, current status and future potential". In: *Computerized medical imaging and graphics* 31.4-5 (2007), pp. 198–211.

[18] David E Newman-Toker, Zheyu Wang, Yuxin Zhu, Najlla Nassery, Ali S Saber Tehrani, Adam C Schaffer, Chihwen Winnie Yu-Moe, Gwendolyn D Clemens, Mehdi Fanai, and Dana Siegal. "Rate of diagnostic errors and serious misdiagnosis-related harms for major vascular events, infections, and cancers: toward a national incidence estimate using the "Big Three"". In: *Diagnosis* 1.ahead-of-print (2020).

[19] Eric J Chin, Andrew Bloom, and Andrew Thompson. "A comparison of perceived acceptable missed diagnosis rates for high-risk emergency medicine diagnoses: A brief report". In: *The American journal of emergency medicine* 35.12 (2017), pp. 1973–1977.

[20] Nam Hee Kim, Yoon Suk Jung, Woo Shin Jeong, Hyo-Joon Yang, Soo-Kyung Park, Kyuyong Choi, and Dong Il Park. "Miss rate of colorectal neoplastic polyps and risk factors for missed polyps in consecutive colonoscopies". In: *Intestinal research* 15.3 (2017), p. 411.

[21] LM Blendis, WJ McNeilly, Louise Sheppard, Roger Williams, and JW Laws. "Observer variation in the clinical and radiological assessment of hepatosplenomegaly". In: *Br Med J* 1.5698 (1970), pp. 727–730.

[22] Paul Brennan and Alan Silman. "Statistical methods for assessing observer variability in clinical measures." In: *BMJ: British Medical Journal* 304.6840 (1992), p. 1491.

[23] Hanna Borgli, Vajira Thambawita, Pia H Smedsrud, Steven Hicks, Debesh Jha, Sigrun L Eskeland, Kristin Ranheim Randel, Konstantin Pogorelov, Mathias Lux, Duc Tien Dang Nguyen, et al. "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy". In: *Scientific Data* 7.1 (2020), pp. 1–14.

[24] Henrik Svoren, Vajira Thambawita, Pål Halvorsen, Petter Jakobsen, Enrique Garcia-Ceja, Farzan Majeed Noori, Hugo L. Hammer, Mathias Lux, Michael Alexander Riegler, and Steven Alexander Hicks. "Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros". In: *Proceedings of the 11th ACM Multimedia Systems Conference*. MMSys '20. Istanbul, Turkey: Association for Computing Machinery, 2020, pp. 309–314. ISBN: 9781450368452. DOI: 10.1145/3339825.3394939. URL: https://doi.org/10.1145/3339825.3394939.

[25] Vajira Thambawita, Steven Alexander Hicks, Hanna Borgli, Håkon Kvale Stensland, Debesh Jha, Martin Kristoffer Svensen, Svein-Arne Pettersen, Dag Johansen, Håvard Dagenborg Johansen, Susann Dahl Pettersen, Simon Nordvang, Sigurd Pedersen, Anders Gjerdrum, Tor-Morten Grønli, Per Morten Fredriksen, Ragnhild Eg, Kjeld Hansen, Siri Fagernes, Christine Claudi, Andreas Biørn-Hansen, Duc Tien Dang Nguyen, Tomas Kupka, Hugo Lewi Hammer, Ramesh Jain, Michael Alexander Riegler, and Pål Halvorsen. "PMData: A Sports Logging Dataset". In: *Proceedings of the 11th ACM Multimedia Systems Conference*. MMSys '20. Istanbul, Turkey: Association for Computing Machinery, 2020, pp. 231–236. ISBN:

9781450368452. DOI: 10.1145/3339825.3394926. URL: https://doi.org/10.1145/3339825.3394926.

[26] P. Jakobsen, E. Garcia-Ceja, L. A. Stabell, K. J. Oedegaard, J. O. Berle, V. Thambawita, S. A. Hicks, P. Halvorsen, O. B. Fasmer, and M. A. Riegler. "PSYKOSE: A Motor Activity Database of Patients with Schizophrenia". In: *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*. 2020, pp. 303–308. DOI: 10.1109/CBMS49503.2020.00064.

[27] Pia H. Smedsrud, Vajira Thambawita, Steven A. Hicks, Henrik Gjestang, Oda Olsen Nedrejord, Espen Næss, Hanna Borgli, Debesh Jha, Tor Jan Derek Berstad, Sigrun L. Eskeland, Mathias Lux, Håvard Espeland, Andreas Petlund, Duc Tien Dang Nguyen, Enrique Garcia-Ceja, Dag Johansen, Peter T. Schmidt, Ervin Toth, Hugo L. Hammer, Thomas de Lange, Michael A. Riegler, and Pål Halvorsen. "Kvasir-Capsule, a video capsule endoscopy dataset". In: *Scientific Data* 8.1 (2021), p. 142. DOI: 10.1038/s41597-021-00920-z. URL: https://doi.org/10.1038/s41597-021-00920-z.

[28] Enrique Garcia-Ceja, Vajira Thambawita, Steven A. Hicks, Debesh Jha, Petter Jakobsen, Hugo L. Hammer, Pål Halvorsen, and Michael A. Riegler. "HTAD: A Home-Tasks Activities Dataset with Wrist-Accelerometer and Audio Features". In: *MultiMedia Modeling*. Ed. by Jakub Lokoč, Tomáš Skopal, Klaus Schoeffmann, Vasileios Mezaris, Xirong Li, Stefanos Vrochidis, and Ioannis Patras. Cham: Springer International Publishing, 2021, pp. 196–205. ISBN: 9783030678357.

[29] Debesh Jha, Sharib Ali, Krister Emanuelsen, Steven Hicks, Vajira Thambawita, Enrique Garcia-Ceja, Michael Riegler, Thomas de Lange, Peter T Schmidt, Håvard Johansen, et al. "Kvasir-Instrument: Diagnostic and Therapeutic Tool Segmentation Dataset in Gastrointestinal Endoscopy". In: ().

[30] Vajira Thambawita, Debesh Jha, Michael Riegler, Pål Halvorsen, Hugo Lewi Hammer, Håvard D Johansen, and Dag Johansen. "The Medico-Task 2018: Disease Detection in the Gastrointestinal Tract using Global Features and Deep Learning". In: *Proc. of MediaEval* (2018).

[31] Vajira Thambawita, Debesh Jha, Hugo Lewi Hammer, Håvard D. Johansen, Dag Johansen, Pål Halvorsen, and Michael A. Riegler. "An Extensive Study on Cross-

Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification". In: *ACM Trans. Comput. Healthcare* 1.3 (June 2020). ISSN: 2691-1957. DOI: 10.1145/3386295. URL: https://doi.org/10.1145/3386295.

[32]   Vajira Thambawita, Steven Hicks, Strümke Inga, Michael A. Riegler, Pål Halvorsen, and Sravanthi Parasa. "Impact of image resolution on convolutional neural networks performance in gastrointestinal endoscopy". In: *Gastrointestinal Endoscopy* (2021). DDW 2021 AGA Program and Abstracts.

[33]   Steven A. Hicks, Inga Strümke, Vajira Thambawita, Malek Hammou, Michael A. Riegler, Pål Halvorsen, and Sravanthi Parasa. "On evaluation metrics for medical applications of artificial intelligence". In: *medRxiv* (2021). DOI: 10.1101/2021.04.07.21254975. eprint: https://www.medrxiv.org/content/early/2021/04/09/2021.04.07.21254975.full.pdf. URL: https://www.medrxiv.org/content/early/2021/04/09/2021.04.07.21254975.

[34]   Henrik L. Gjestang, Steven A. Hicks, Vajira Thambawita, Pål Halvorsen, and Michael A. Riegler. "A self-learning teacher-student framework for gastrointestinal image classification". In: *Proceedings of International Symposium on Computer-Based Medical Systems (CBMS)*. 2021.

[35]   Vajira Thambawita, Steven A. Hicks, Pål Halvorsen, and Michael A. Riegler. "DivergentNets: Medical Image Segmentation by Network Ensemble". In: *Proceedings of the 3rd International Workshop and Challenge on Computer Vision in Endoscopy (EndoCV 2021) co-located with with the 17th IEEE International Symposium on Biomedical Imaging (ISBI 2021)*. 2021.

[36]   Vajira Thambawita, Steven Hicks, Pål Halvorsen, and Michael A Riegler. "Pyramid-Focus-Augmentation: Medical Image Segmentation with Step-Wise Focus". In: *arXiv preprint arXiv:2012.07430* (2020).

[37]   B. Mac Namee, P. Cunningham, S. Byrne, and O.I. Corrigan. "The problem of bias in training data in regression problems in medical decision support". In: *Artificial Intelligence in Medicine* 24.1 (2002), pp. 51–70. ISSN: 0933-3657. DOI: https://doi.org/10.1016/S0933-3657(01)00092-6. URL: https://www.sciencedirect.com/science/article/pii/S0933365701000926.

[38] Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, Vajira Thambawita, Pål Halvorsen, Hugo L. Hammer, Trine B. Haugen, and Michael A. Riegler. "Machine Learning-Based Analysis of Sperm Videos and Participant Data for Male Fertility Prediction". In: *Scientific Reports* 9.1 (2019). DOI: 10.1038/s41598-019-53217-y.

[39] Vajira Thambawita, Pål Halvorsen, Hugo Hammer, Michael Riegler, and Trine B Haugen. "Stacked Dense Optical Flows and Dropout Layers to Predict Sperm Motility and Morphology". In: *Proc. of MediaEval* (2019).

[40] Vajira Thambawita, Pål Halvorsen, Hugo Hammer, Michael Riegler, and Trine B Haugen. "Extracting Temporal Features into a Spatial Domain Using Autoencoders for Sperm Video Analysis". In: *Proceedings of MediaEval* (2019).

[41] Steven A. Hicks, Jonas L. Isaksen, Vajira Thambawita, Jonas Ghouse, Gustav Ahlberg, Allan Linneberg, Niels Grarup, Inga Strümke, Christina Ellervik, Morten Salling Olesen, Torben Hansen, Claus Graff, Niels-Henrik Holstein-Rathlou, Pål Halvorsen, Mary M. Maleckar, Michael A. Riegler, and Jørgen K. Kanters. "Explaining deep neural networks for knowledge discovery in electrocardiogram analysis". In: *Scientific Reports* 11.1 (2021), p. 10949. DOI: 10.1038/s41598-021-90285-5. URL: https://doi.org/10.1038/s41598-021-90285-5.

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[43] Filip Karlo Došilović, Mario Brčić, and Nikica Hlupić. "Explainable artificial intelligence: A survey". In: *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE. 2018, pp. 0210–0215.

[44] Martin J Willemink, Wojciech A Koszek, Cailin Hardell, Jie Wu, Dominik Fleischmann, Hugh Harvey, Les R Folio, Ronald M Summers, Daniel L Rubin, and Matthew P Lungren. "Preparing medical imaging data for machine learning". In: *Radiology* 295.1 (2020), pp. 4–15.

[45]   Deven McGraw and Kenneth D. Mandl. "Privacy protections to encourage use of health-relevant digital data in a learning health system". In: *npj Digital Medicine* 4.1 (2021), p. 2. DOI: 10.1038/s41746-020-00362-8. URL: https://doi.org/10.1038/s41746-020-00362-8.

[46]   2nd Price W Nicholson and I Glenn Cohen. "Privacy in the age of medical big data". In: *Nature medicine* 1 (Jan.), pp. 37–43. DOI: 10.1038/s41591-018-0272-7.

[47]   Marcello Ienca, Agata Ferretti, Samia Hurst, Milo Puhan, Christian Lovis, and Effy Vayena. "Considerations for ethics review of big data health research: A scoping review". In: *PloS one* 13.10 (2018), e0204937.

[48]   Bartha Maria Knoppers and Adrian Mark Thorogood. "Ethics and big data in health". In: *Current Opinion in Systems Biology* 4 (2017), pp. 53–57.

[49]   Julia Lane and Claudia Schur. "Balancing access to health data and privacy: a review of the issues and approaches for the future". In: *Health services research* 45.5p2 (2010), pp. 1456–1467.

[50]   *The Norwegian Data Protection Authority.* Accessed: 2021-04-25. URL: https://www.datatilsynet.no/en/.

[51]   *The Personal Data Act.* Accessed: 2021-04-25. URL: https://www.forskningsetikk.no/en/resources/the-research-ethics-library/legal-statutes-and-guidelines/the-personal-data-act/.

[52]   Paul Voigt and Axel Von dem Bussche. "The eu general data protection regulation (gdpr)". In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10 (2017), p. 3152676.

[53]   Peter Edemekong, Pavan Annamaraju, and Micelle Haydel. "Health Insurance Portability and Accountability Act". In: *StatPearls* (2020).

[54]   *California Consumer Privacy Act.* 2018. URL: https://oag.ca.gov/privacy/ccpa.

[55]   *Act on the Protection of Personal Information.* 2003. URL: https://www.cas.go.jp/jp/seisaku/hourei/data/APPI.pdf.

[56] *Personal Information Protection Commission.* 2011. URL: `http://www.pipc.go.kr/cmt/main/english.do`.

[57] *THE PERSONAL DATA PROTECTION BILL.* 2018. URL: `https://www.meity.gov.in/writereaddata/files/Personal_Data_Protection_Bill,2018.pdf`.

[58] *Anolytics - Making AI Possible with Right Data and Image Annotation Services.* Accessed: 2021-04-25. URL: `https://www.anolytics.ai/solutions/healthcare/`.

[59] *Mindy Support.* Accessed: 2021-04-25. URL: `https://mindy-support.com/industries-posts/healthcare/`.

[60] Shaode Yu, Mingli Chen, Erlei Zhang, Junjie Wu, Hang Yu, Zi Yang, Lin Ma, Xuejun Gu, and Weiguo Lu. "Robustness study of noisy annotation in deep learning based medical image segmentation". In: *Physics in Medicine & Biology* 65.17 (2020), p. 175007. DOI: `10.1088/1361-6560/ab99e5`. URL: `https://doi.org/10.1088/1361-6560/ab99e5`.

[61] Google. *Labeling costs.* `https://cloud.google.com/ai-platform/data-labeling/pricing`. Apr. 2021.

[62] Christopher J Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. "Key challenges for delivering clinical impact with artificial intelligence". In: *BMC medicine* 17.1 (2019), pp. 1–9.

[63] Jeya Vikranth Jeyakumar, Joseph Noor, Yu-Hsi Cheng, Luis Garcia, and Mani Srivastava. "How Can I Explain This to You? An Empirical Study of Deep Neural Network Explanation Methods". In: *Advances in Neural Information Processing Systems* 33 (2020).

[64] Jasper van der Waa, Elisabeth Nieuwburg, Anita Cremers, and Mark Neerincx. "Evaluating XAI: A comparison of rule-based and example-based explanations". In: *Artificial Intelligence* 291 (2021), p. 103404. ISSN: 0004-3702. DOI: `https://doi.org/10.1016/j.artint.2020.103404`. URL: `http://www.sciencedirect.com/science/article/pii/S0004370220301533`.

[65] Gordana Dodig-Crnkovic. "Scientific methods in computer science". In: *Proceedings of the Conference for the Promotion of Research in IT at New Universities and at University Colleges in Sweden, Skövde, Suecia.* 2002, pp. 126–130.

[66] D. E. Comer, David Gries, Michael C. Mulder, Allen Tucker, A. Joe Turner, Paul R. Young, and Peter J. Denning. "Computing as a Discipline". In: *Commun. ACM* 32.1 (Jan. 1989), pp. 9–23. ISSN: 0001-0782. DOI: 10.1145/63238.63239. URL: https://doi.org/10.1145/63238.63239.

[67] Vajira Thambawita, Pegah Salehi, Sajad Amouei Sheshkal, Steven Hicks, Hugo L. Hammer, Sravanthi Parasa, Thomas de Lange, Pål Halvorsen, and Michael A Riegler. "SinGAN-Seg: Synthetic Training Data Generation for Medical Image Segmentation". In: *arXiv preprint* (2021).

[68] Steven A. Hicks, Vajira Thambawita, Hugo L. Hammer, Trine B. Haugen, Jorunn M. Andersen, Oliwia Witczak, Pål Halvorsen, and Michael A. Riegler. "ACM Multimedia BioMedia 2020 Grand Challenge Overview". In: New York, NY, USA: Association for Computing Machinery, 2020, pp. 4655–4658. ISBN: 9781450379885. URL: https://doi.org/10.1145/3394171.3416287.

[69] Trine B. Haugen, Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, Hugo L. Hammer, Rune Borgli, Pål Halvorsen, and Michael A. Riegler. "VISEM: A Multimodal Video Dataset of Human Spermatozoa". In: *Proceedings of the 10th ACM on Multimedia Systems Conference*. MMSys'19. Amherst, MA, USA: ACM, 2019. DOI: 10.1145/3304109.3325814. URL: http://doi.acm.org/10.1145/3304109.3325814.

[70] Vajira Thambawita, Jonas L Isaksen, Steven Hicks, Jonas Ghouse, Gustav Ahlberg, Allan Linneberg, Niels Grarup, Christina Ellervik, Morten Salling Olesen, Torben Hansen, Claus Graff, Niels-Henrik Holstein-Rathlou, Inga Strümke, Hugo L. Hammer, Mary M Maleckar, Pål Halvorsen, Michael A. Riegler, and Jørgen K. Kanters. "DeepFake electrocardiograms: the beginning of the end for privacy issues in medicine". In: *medRxiv* (2021). DOI: 10.1101/2021.04.27.21256189. eprint: https://www.medrxiv.org/content/early/2021/04/30/2021.04.27.21256189.full.pdf. URL: https://www.medrxiv.org/content/early/2021/04/30/2021.04.27.21256189.

[71] Vajira Thambawita, Steven A. Hicks, Jonas Isaksen, Mette Haug Stensen, Trine B. Haugen, JØrgen Kanters, Sravanthi Parasa, Thomas de Lange, Håvard D. Johansen, Dag Johansen, Hugo L. Hammer, Pål Halvorsen, and Michael A. Riegler.

"DeepSynthBody: the beginning of the end for data deficiency in medicine". In: *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*. 2021, pp. 1–8. DOI: 10.1109/ICAPAI49758.2021.9462062.

[72] Mathias Kirkerød, Vajira Thambawita, Michael Riegler, and Pål Halvorsen. "Using Preprocessing as a Tool in Medical Image Detection." In: *Proceedings of MediaEval*. 2018.

[73] Mathias Kirkerød, Rune Johan Borgli, Vajira Thambawita, Steven Hicks, Michael Alexander Riegler, and Pål Halvorsen. "Unsupervised preprocessing to improve generalisation for medical image classification". In: *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*. 2019, pp. 1–6. DOI: 10.1109/ISMICT.2019.8743979.

[74] Oda O. Nedrejord, Vajira Thambawita, Steven A. Hicks, Pål Halvorsen, and Michael A. Riegler. "Vid2Pix - A Framework for Generating High-Quality Synthetic Videos". In: *2020 IEEE International Symposium on Multimedia (ISM)*. 2020, pp. 25–26. DOI: 10.1109/ISM.2020.00010.

[75] Vajira Thambawita, Steven Hicks, Strümke Inga, Michael A. Riegler, Pål Halvorsen, and Sravanthi Parasa. "Generative Adversarial Networks For Creating Realistic Artificial Colon Polyp Images". In: *Gastrointestinal Endoscopy* (2021). DDW 2021 ASGE Program and Abstracts.

[76] Vajira Thambawita, Trine B. Haugen, Mette Haug Stensen, Oliwia Witczak, Hugo L. Hammer, Pål Halvorsen, and Michael A. Riegler. "Identification of spermatozoa by unsupervised learning from video data". In: *Proceedings of ESHRE*. 2021.

[77] Vajira Thambawita, Hugo Lewi Hammer, Michael Riegler, and Pål Halvorsen. "GANEx: A complete pipeline of training, inference and benchmarking GAN experiments". In: *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE. 2019, pp. 1–4.

[78] Olav A. Norgård Rongved, Steven A. Hicks, Vajira Thambawita, Håkon K. Stensland, Evi Zouganeli, Dag Johansen, Michael A. Riegler, and Pål Halvorsen. "Real-Time Detection of Events in Soccer Videos using 3D Convolutional Neural Networks". In: *2020 IEEE International Symposium on Multimedia (ISM)*. 2020, pp. 135–144. DOI: 10.1109/ISM.2020.00030.

[79] M. C. Elish and danah boyd. "Situating methods in the magic of Big Data and AI". In: *Communication Monographs* 85.1 (2018), pp. 57–80. DOI: 10.1080/03637751. 2017.1375130. eprint: https://doi.org/10.1080/03637751.2017.1375130. URL: https://doi.org/10.1080/03637751.2017.1375130.

[80] Kristian Kersting and Ulrich Meyer. "From Big Data to Big Artificial Intelligence?" In: *KI - Künstliche Intelligenz* 32.1 (2018), pp. 3–8. DOI: 10.1007/s13218-017-0523-7. URL: https://doi.org/10.1007/s13218-017-0523-7.

[81] Kurt Benke and Geza Benke. "Artificial Intelligence and Big Data in Public Health". In: *International journal of environmental research and public health* 15.12 (Dec. 2018), p. 2796. DOI: 10.3390/ijerph15122796. URL: https://pubmed.ncbi.nlm.nih.gov/30544648.

[82] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[83] *A sample MRI - 7-Tesla MRI scanner*. Accessed: 2021-04-25. URL: https://openneuro.org/datasets/ds003642/versions/1.0.0/file-display/sub-075:ses-001:anat:sub-075_ses-001_INV2.nii.gz.

[84] *A sample MRI - 7-Tesla MRI scanner*. Accessed: 2021-04-25. URL: http://neuromorpho.org/.

[85] *Organ System Definition*. https://biologydictionary.net/organ-system/. Accessed: 2021-01-25.

[86] Kim D Pruitt, Tatiana Tatusova, Garth R Brown, and Donna R Maglott. "NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy". In: *Nucleic acids research* 40.D1 (2012), pp. D130–D135.

[87] Stephen T Sherry, M-H Ward, M Kholodov, J Baker, Lon Phan, Elizabeth M Smigielski, and Karl Sirotkin. "dbSNP: the NCBI database of genetic variation". In: *Nucleic acids research* 29.1 (2001), pp. 308–311.

[88] Alexander Andreopoulos and John K Tsotsos. "Efficient and generalizable statistical models of shape and appearance for analysis of cardiac MRI". In: *Medical Image Analysis* 12.3 (2008), pp. 335–357.

[89] Jianwei Zheng, Jianming Zhang, Sidy Danioko, Hai Yao, Hangyuan Guo, and Cyril Rakovski. "A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients". In: *Scientific Data* 7.1 (2020), p. 48. DOI: `10.1038/s41597-020-0386-x`. URL: `https://doi.org/10.1038/s41597-020-0386-x`.

[90] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. "KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection". In: *Proceedings of the 8th ACM on Multimedia Systems Conference.* MMSys'17. Taipei, Taiwan: ACM, 2017, pp. 164–169. ISBN: 9781450350020. DOI: `10.1145/3083187.3083212`. URL: `http://doi.acm.org/10.1145/3083187.3083212`.

[91] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, Mario Taschwer, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. "Nerthus: A Bowel Preparation Quality Video Dataset". In: *Proceedings of the 8th ACM on Multimedia Systems Conference.* MMSys'17. Taipei, Taiwan: ACM, 2017, pp. 170–174. ISBN: 9781450350020. DOI: `10.1145/3083187.3083216`. URL: `http://doi.acm.org/10.1145/3083187.3083216`.

[92] Romain Leenhardt, Cynthia Li, Jean-Philippe Le Mouel, Gabriel Rahmi, Jean Christophe Saurin, Franck Cholet, Arnaud Boureille, Xavier Amiot, Michel Delvaux, Clotilde Duburque, Chloé Leandri, Romain Gérard, Stéphane Lecleire, Farida Mesli, Isabelle Nion-Larmurier, Olivier Romain, Sylvie Sacher-Huvelin, Camille Simon-Shane, Geoffroy Vanbiervliet, Philippe Marteau, Aymeric Histace, and Xavier Dray. "CAD-CAP: a 25,000-image database serving the development of artificial intelligence for capsule endoscopy". In: *Endoscopy international open* 8.3 (Mar. 2020), E415–E420. DOI: `10.1055/a-1035-9088`. URL: `https://pubmed.ncbi.nlm.nih.gov/32118115`.

[93] Manuel Barberio, Marianne Maktabi, Ines Gockel, Nada Rayes, Boris Jansen-Winkeln, Hannes Köhler, Sebastian M. Rabe, Lena Seidemann, Jonathan P. Takoh, Michele Diana, Thomas Neumuth, and Claire Chalopin. "Hyperspectral based discrimination of thyroid and parathyroid during surgery". In: *Current Directions in*

Biomedical Engineering 4.1 (2018), pp. 399–402. DOI: `doi:10.1515/cdbme-2018-0095`. URL: `https://doi.org/10.1515/cdbme-2018-0095`.

[94] Lina Pedraza, Carlos Vargas, Fabián Narváez, Oscar Durán, Emma Muñoz, and Eduardo Romero. "An open access thyroid ultrasound image database". In: *10th International Symposium on Medical Information Processing and Analysis.* Vol. 9287. International Society for Optics and Photonics. 2015, 92870W.

[95] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific Data* 5.1 (2018), p. 180161. DOI: `10.1038/sdata.2018.161`. URL: `https://doi.org/10.1038/sdata.2018.161`.

[96] Veronica Rotemberg, Nicholas Kurtansky, Brigid Betz-Stablein, Liam Caffery, Emmanouil Chousakos, Noel Codella, Marc Combalia, Stephen Dusza, Pascale Guitera, David Gutman, et al. "A patient-centric dataset of images and metadata for identifying melanomas using clinical context". In: *Scientific data* 8.1 (2021), pp. 1–8.

[97] Holger R Roth, Le Lu, Ari Seff, Kevin M Cherry, Joanne Hoffman, Shijun Wang, Jiamin Liu, Evrim Turkbey, and Ronald M Summers. "A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations". In: *International conference on medical image computing and computer-assisted intervention.* Springer. 2014, pp. 520–527.

[98] Kemal Polat and Salih Güneş. "Automatic determination of diseases related to lymph system from lymphography data using principles component analysis (PCA), fuzzy weighting pre-processing and ANFIS". In: *Expert Systems with Applications* 33.3 (2007), pp. 636–641. ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2006.06.004`. URL: `https://www.sciencedirect.com/science/article/pii/S0957417406001898`.

[99] G. Andreisek, Benedikt Kislinger, R. Dessouky, and A. Chhabra. "MRI of the Intrinsic Muscles of the Hand". In: *Seminars in Musculoskeletal Radiology* 21 (2017), pp. 392–402.

Bibliography

[100] F. S. Gayzik, D. P. Moreno, C. P. Geer, S. D. Wuertzer, R. S. Martin, and J. D. Stitzel. "Development of a Full Body CAD Dataset for Computational Modeling: A Multi-modality Approach". In: *Annals of Biomedical Engineering* 39.10 (2011), p. 2568. DOI: `10.1007/s10439-011-0359-5`. URL: `https://doi.org/10.1007/s10439-011-0359-5`.

[101] Nadine Chang, John A. Pyles, Austin Marcus, Abhinav Gupta, Michael J. Tarr, and Elissa M. Aminoff. "BOLD5000, a public fMRI dataset while viewing 5000 visual images". In: *Scientific Data* 6.1 (2019), p. 49. DOI: `10.1038/s41597-019-0052-3`. URL: `https://doi.org/10.1038/s41597-019-0052-3`.

[102] Florian Knoll, Martin Holler, Thomas Koesters, Ricardo Otazo, Kristian Bredies, and Daniel K Sodickson. "Joint MR-PET Reconstruction Using a Multi-Channel Image Regularizer". In: *IEEE Transactions on Medical Imaging* 36.1 (2017), pp. 1–16. DOI: `10.1109/TMI.2016.2564989`.

[103] Konrad S Famulski, Declan G de Freitas, Chatchai Kreepala, Jessica Chang, Joana Sellares, Banu Sis, Gunilla Einecke, Michael Mengel, Jeff Reeve, and Philip F Halloran. "Molecular phenotypes of acute kidney injury in kidney transplants". In: *Journal of the American Society of Nephrology* 23.5 (2012), pp. 948–958.

[104] Mehmet Sarier, Ibrahim Duman, Mehmet Callioglu, Ahmet Soylu, Sabri Tekin, Emrah Turan, Hasan Celep, Asuman Havva Yavuz, Alper Demirbas, and Erdal Kukul. "Outcomes of conservative management of asymptomatic live donor kidney stones". In: *Urology* 118 (2018), pp. 43–46.

[105] Soroush Javadi and Seyed Abolghasem Mirroshandel. "A novel deep learning method for automatic assessment of human sperm images". In: *Computers in Biology and Medicine* 109 (2019), pp. 182–194. ISSN: 0010-4825. DOI: `https://doi.org/10.1016/j.compbiomed.2019.04.030`. URL: `https://www.sciencedirect.com/science/article/pii/S0010482519301386`.

[106] Fernando H Biase, Xiaoyi Cao, and Sheng Zhong. "Cell fate inclination within 2-cell and 4-cell mouse embryos revealed by single-cell RNA sequencing". In: *Genome research* 24.11 (2014), pp. 1787–1796.

[107] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2097–2106.

[108] SP Morozov, AE Andreychenko, NA Pavlov, AV Vladzymyrskyy, NV Ledikhova, VA Gombolevskiy, Ivan A Blokhin, PB Gelezhe, AV Gonchar, and V Yu Chernina. "Mosmeddata: Chest ct scans with covid-19 related findings dataset". In: *arXiv preprint arXiv:2005.06465* (2020).

[109] Pranav Rajpurkar, Jeremy Irvin, Aarti Bagul, Daisy Ding, Tony Duan, Hershel Mehta, Brandon Yang, Kaylie Zhu, Dillon Laird, Robyn L Ball, et al. "Mura: Large dataset for abnormality detection in musculoskeletal radiographs". In: *arXiv preprint arXiv:1712.06957* (2017).

[110] Nicholas Bien, Pranav Rajpurkar, Robyn L Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, Bhavik N Patel, Kristen W Yeom, Katie Shpanskaya, et al. "Deep-learning-assisted diagnosis for knee magnetic resonance imaging: development and retrospective validation of MRNet". In: *PLoS medicine* 15.11 (2018), e1002699.

[111] Alhanoof Althnian, Duaa AlSaeed, Heyam Al-Baity, Amani Samha, Alanoud Bin Dris, Najla Alzakari, Afnan Abou Elwafa, and Heba Kurdi. "Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain". In: *Applied Sciences* 11.2 (2021), p. 796.

[112] Sandeep Dutta and Eric Gros. "Evaluation of the impact of deep learning architectural components selection and dataset size on a medical imaging task". In: *Medical Imaging 2018: Imaging Informatics for Healthcare, Research, and Applications*. Vol. 10579. International Society for Optics and Photonics. 2018, p. 1057911.

[113] Maciej A Mazurowski, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker, and Georgia D Tourassi. "Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance". In: *Neural networks* 21.2-3 (2008), pp. 427–436.

[114] Bartosz Krawczyk. "Learning from imbalanced data: open challenges and future directions". In: *Progress in Artificial Intelligence* 5.4 (2016), pp. 221–232.

[115] Edward S Dove and Mark Phillips. "Privacy law, data sharing policies, and medical data: a comparative perspective". In: *Medical data privacy handbook*. Springer, 2015, pp. 639–678.

[116] Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen. "Kvasir-SEG: A Segmented Polyp Dataset". In: *MultiMedia Modeling*. Ed. by Yong Man Ro, Wen-Huang Cheng, Junmo Kim, Wei-Ta Chu, Peng Cui, Jung-Woo Choi, Min-Chun Hu, and Wesley De Neve. Cham: Springer International Publishing, 2020, pp. 451–462. ISBN: 9783030377342.

[117] Jorge Bernal, F Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodrıguez, and Fernando Vilariño. "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians". In: *Computerized Medical Imaging and Graphics* 43 (2015), pp. 99–111.

[118] Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer". In: *International journal of computer assisted radiology and surgery* 9.2 (2014), pp. 283–293.

[119] D.F. Specht. "A general regression neural network". In: *IEEE Transactions on Neural Networks* 2.6 (1991), pp. 568–576. DOI: 10.1109/72.97934.

[120] Xueheng Qiu, Le Zhang, Ye Ren, P. N. Suganthan, and Gehan Amaratunga. "Ensemble deep learning for regression and time series forecasting". In: *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*. 2014, pp. 1–6. DOI: 10.1109/CIEL.2014.7015739.

[121] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. "Accurate Uncertainties for Deep Learning Using Calibrated Regression". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, July 2018, pp. 2796–2804. URL: http://proceedings.mlr.press/v80/kuleshov18a.html.

[122] Heba Mohsen, El-Sayed A El-Dahshan, El-Sayed M El-Horbaty, and Abdel-Badeeh M Salem. "Classification using deep learning neural networks for brain tumors". In: *Future Computing and Informatics Journal* 3.1 (2018), pp. 68–71.

[123] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9 (2017), pp. 2352–2449.

[124] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 2503–2510.

[125] Zhuoling Li, Minghui Dong, Shiping Wen, Xiang Hu, Pan Zhou, and Zhigang Zeng. "CLU-CNNs: Object detection for medical images". In: *Neurocomputing* 350 (2019), pp. 53–59.

[126] Mohammad Hesam Hesamian, Wenjing Jia, Xiangjian He, and Paul Kennedy. "Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges". In: *Journal of Digital Imaging* 32.4 (2019), pp. 582–596. DOI: `10.1007/s10278-019-00227-x`. URL: `https://doi.org/10.1007/s10278-019-00227-x`.

[127] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N Chiang, Zhihao Wu, and Xiaowei Ding. "Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation". In: *Medical Image Analysis* 63 (2020), p. 101693.

[128] Ravi K Samala, Heang-Ping Chan, Lubomir Hadjiiski, and Sathvik Koneru. "Hazards of data leakage in machine learning: a study on classification of breast cancer using deep neural networks". In: *Medical Imaging 2020: Computer-Aided Diagnosis*. Vol. 11314. International Society for Optics and Photonics. 2020, p. 1131416.

[129] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[130] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[131] Lu Mi, Macheng Shen, and Jingzhao Zhang. "A probe towards understanding gan and vae models". In: *arXiv preprint arXiv:1812.05676* (2018).

[132] Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. "Variational autoencoder for semi-supervised text classification". In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 31. 1. 2017.

[133] Xin Yi, Ekta Walia, and Paul Babyn. "Generative adversarial network in medical imaging: A review". In: *Medical image analysis* 58 (2019), p. 101552.

[134] Shengjia Zhao, Jiaming Song, and Stefano Ermon. "Towards deeper understanding of variational autoencoding models". In: *arXiv preprint arXiv:1702.08658* (2017).

[135] T. Jaydeep. "Comparative Study of GAN and VAE". In: *International Journal of Computer Applications* 182 (2018), pp. 1–5.

[136] Farzan Farnia and Asuman Ozdaglar. "Do GANs always have Nash equilibria?" In: *Proceedings of the 37th International Conference on Machine Learning.* Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 3029–3039. URL: http://proceedings.mlr.press/v119/farnia20a.html.

[137] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. "Are GANs Created Equal? A Large-Scale Study". In: *Advances in Neural Information Processing Systems.* Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.

[138] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. "Seeing what a gan cannot generate". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019, pp. 4502–4511.

[139] Zhaoyu Zhang, Mengyan Li, and Jun Yu. "On the convergence and mode collapse of gan". In: *SIGGRAPH Asia 2018 Technical Briefs.* 2018, pp. 1–4.

[140] Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. "Generative adversarial network: An overview of theory and applications". In: *International Journal of Information Management Data Insights* (2021), p. 100004.

[141] Lan Lan, Lei You, Zeyang Zhang, Zhiwei Fan, Weiling Zhao, Nianyin Zeng, Yidong Chen, and Xiaobo Zhou. "Generative Adversarial Networks and Its Applications in Biomedical Informatics". In: *Frontiers in Public Health* 8 (2020), p. 164.

[142]   Zhengwei Wang, Qi She, and Tomas E Ward. "Generative adversarial networks in computer vision: A survey and taxonomy". In: *arXiv preprint arXiv:1906.01529* (2019).

[143]   Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., 2014. URL: `https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf`.

[144]   Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *CVPR* (2017).

[145]   Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In: *Computer Vision (ICCV), 2017 IEEE International Conference on*. 2017.

[146]   Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and Improving the Image Quality of StyleGAN". In: *CoRR* abs/1912.04958 (2019).

[147]   Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Training generative adversarial networks with limited data". In: *arXiv preprint arXiv:2006.06676* (2020).

[148]   Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large Scale GAN Training for High Fidelity Natural Image Synthesis". In: *International Conference on Learning Representations*. 2019. URL: `https://openreview.net/forum?id=B1xsqj09Fm`.

[149]   Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. "SinGAN: Learning a Generative Model from a Single Natural Image". In: *Computer Vision (ICCV), IEEE International Conference on*. 2019.

[150]   Ali Borji. "Pros and cons of gan evaluation measures". In: *Computer Vision and Image Understanding* 179 (2019), pp. 41–65.

[151] Maayan Frid-Adar, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. "Synthetic data augmentation using GAN for improved liver lesion classification". In: *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*. IEEE. 2018, pp. 289–293.

[152] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Fadi Al-Turjman, and Plácido Rogerio Pinheiro. "Covidgan: data augmentation using auxiliary classifier gan for improved covid-19 detection". In: *Ieee Access* 8 (2020), pp. 91916–91923.

[153] Talha Iqbal and Hazrat Ali. "Generative adversarial network for medical images (MI-GAN)". In: *Journal of medical systems* 42.11 (2018), pp. 1–11.

[154] Hitesh Tekchandani, Shrish Verma, and Narendra Londhe. "Performance improvement of mediastinal lymph node severity detection using GAN and Inception network". In: *Computer Methods and Programs in Biomedicine* 194 (2020), p. 105478.

[155] Avi Ben-Cohen, Eyal Klang, Stephen P Raskin, Shelly Soffer, Simona Ben-Haim, Eli Konen, Michal Marianne Amitai, and Hayit Greenspan. "Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection". In: *Engineering Applications of Artificial Intelligence* 78 (2019), pp. 186–194.

[156] Shuangting Liu, Jiaqi Zhang, Yuxin Chen, Yifan Liu, Zengchang Qin, and Tao Wan. "Pixel level data augmentation for semantic image segmentation using generative adversarial networks". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 1902–1906.

[157] Marco Domenico Cirillo, David Abramian, and Anders Eklund. "Vox2Vox: 3D-GAN for brain tumour segmentation". In: *arXiv preprint arXiv:2003.13653* (2020).

[158] Zhongyi Han, Benzheng Wei, Ashley Mercado, Stephanie Leung, and Shuo Li. "Spine-GAN: Semantic segmentation of multiple spinal structures". In: *Medical image analysis* 50 (2018), pp. 23–35.

[159] Jin Zhu, Guang Yang, and Pietro Lio. "How can we make gan perform better in single medical image super-resolution? A lesion focused multi-scale approach". In:

2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019). IEEE. 2019, pp. 1669–1673.

[160]   Qi Chang, Hui Qu, Yikai Zhang, Mert Sabuncu, Chao Chen, Tong Zhang, and Dimitris N Metaxas. "Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13856–13866.

[161]   Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. "Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record". In: *Journal of the American Medical Informatics Association* 25.3 (2018), pp. 230–238.

[162]   Jinsung Yoon, Lydia N Drumright, and Mihaela Van Der Schaar. "Anonymization through data synthesis using generative adversarial networks (ads-gan)". In: *IEEE journal of biomedical and health informatics* 24.8 (2020), pp. 2378–2388.

[163]   Debapriya Hazra and Yung-Cheol Byun. "SynSigGAN: Generative Adversarial Networks for Synthetic Biomedical Signal Generation". In: *Biology* 9.12 (2020), p. 441.

[164]   Darius Dirvanauskas, Rytis Maskeliūnas, Vidas Raudonis, Robertas Damaševičius, and Rafal Scherer. "Hemigen: human embryo image generator based on generative adversarial networks". In: *Sensors* 19.16 (2019), p. 3578.

[165]   Hasib Zunair and A Ben Hamza. "Synthesis of COVID-19 chest X-rays using unpaired image-to-image translation". In: *Social Network Analysis and Mining* 11.1 (2021), pp. 1–12.

[166]   Anthony Dupre, Sarah Vincent, and Paul A Iaizzo. "Basic ECG theory, recordings, and interpretation". In: *Handbook of cardiac anatomy, physiology, and devices*. Springer, 2005, pp. 191–201.

[167]   Clive Whiston and Florence Elizabeth Prichard. "X-ray Methods". In: (1987).

[168]   Jyoti Shah. "Endoscopy through the ages". In: *BJU international* 89.7 (2002), pp. 645–652.

Bibliography

[169] AM Blamire. "The technology of MRI—the next 10 years?" In: *The British journal of radiology* 81.968 (2008), pp. 601–617.

[170] Michele Larobina and Loredana Murino. "Medical image file formats". In: *Journal of digital imaging* 27.2 (2014), pp. 200–206.

[171] Alois Schlögl. "An overview on data formats for biomedical signals". In: *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany.* Springer. 2009, pp. 1557–1560.

[172] FJ Murphy. "The paradox of imaging technology: a review of the literature". In: *Radiography* 12.2 (2006), pp. 169–174.

[173] Zachary Munn and Zoe Jordan. "The patient experience of high technology medical imaging: a systematic review of the qualitative evidence". In: *Radiography* 17.4 (2011), pp. 323–331.

[174] Jacob Beutel, Harold L Kundel, and Richard L Van Metter. *Handbook of medical imaging.* Vol. 1. Spie Press, 2000.

[175] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[176] *Get to production AI faster.* `https://labelbox.com/`. Accessed: 2021-04-17.

[177] *Basic AI.* `https://www.basic.ai/`. Accessed: 2021-04-17.

[178] *Awesome data annotation.* https://github.com/taivop/awesome-data-annotation. Accessed: 2021-04-17.

[179] CR Juhl, IM Miller, GB Jemec, JK Kanters, and C Ellervik. "Hidradenitis suppurativa and electrocardiographic changes: a cross-sectional population study". In: *British Journal of Dermatology* 178.1 (2018), pp. 222–228.

[180] Jonas Ghouse, Christian Theil Have, Peter Weeke, Jonas Bille Nielsen, Gustav Ahlberg, Marie Balslev-Harder, Emil Vincent Appel, Tea Skaaby, Søren-Peter Olesen, Niels Grarup, et al. "Rare genetic variants previously associated with congenital forms of long QT syndrome have little or no effect on the QT interval". In: *European heart journal* 36.37 (2015), pp. 2523–2529.

[181] *MUSE v9 Cardiology Information System.* Accessed: 2021-03-26. URL: https://www.gehealthcare.com.au/products/diagnostic-ecg/cardio-data-management/muse-v9.

[182] *Electrocardiography-Wikipedia.* Accessed: 2021-03-26. URL: https://en.wikipedia.org/wiki/Electrocardiography.

[183] *Sperm and Semen Testing and Evaluation.* Accessed: 2021-03-26. URL: https://www.fertility-docs.com/programs-and-services/sperm-evaluation/sperm-and-semen-testing.php.

[184] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Steven Hicks, Kristin Ranheim Randel, Duc Tien Dang Nguyen, Mathias Lux, Olga Ostroukhova, and Thomas de Lange. "Medico multimedia task at mediaeval 2018". In: *CEUR Workshop Proceedings.* Vol. 2283. Technical University of Aachen. 2018, pp. 1–4.

[185] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016, pp. 770–778.

[186] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 4700–4708.

[187] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. "A Deeper Look at Dataset Bias". In: *Domain Adaptation in Computer Vision Applications.* Ed. by Gabriela Csurka. Cham: Springer International Publishing, 2017, pp. 37–55. ISBN: 9783319583471. DOI: 10.1007/978-3-319-58347-1_2. URL: https://doi.org/10.1007/978-3-319-58347-1_2.

[188] Debesh Jha, Steven A Hicks, Krister Emanuelsen, Håvard Johansen, Dag Johansen, Thomas de Lange, Michael A Riegler, and Pål Halvorsen. "Medico Multimedia Task at MediaEval 2020: Automatic Polyp Segmentation". In: *arXiv preprint arXiv:2012.15244* (2020).

[189] Gunnar Farnebäck. "Two-frame motion estimation based on polynomial expansion". In: *Scandinavian conference on Image analysis.* Springer. 2003, pp. 363–370.

Bibliography

[190] Steven Hicks, Pål Halvorsen, Trine B Haugen, Jorunn M Andersen, Oliwia Witczak, Konstantin Pogorelov, Hugo L Hammer, Duc-Tien Dang-Nguyen, Mathias Lux, and Michael Riegler. "Medico Multimedia Task at MediaEval 2019". In: *CEUR Workshop Proceedings-Multimedia Benchmark Workshop (MediaEval)*. 2019.

[191] Bruce D Lucas, Takeo Kanade, et al. "An iterative image registration technique with an application to stereo vision". In: Vancouver, British Columbia. 1981.

[192] Chris Donahue, Julian McAuley, and Miller Puckette. "Adversarial Audio Synthesis". In: *International Conference on Learning Representations*. 2019. URL: `https://openreview.net/forum?id=ByMVTsR5KQ`.

[193] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[194] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. "Analyzing and improving the image quality of stylegan". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8110–8119.

[195] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6629–6640. ISBN: 9781510860964.

[196] Tero Karras, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[197] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. "Image Inpainting via Generative Multi-column Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. 2018, pp. 331–340.

[198] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. "A survey on deep transfer learning". In: *International conference on artificial neural networks*. Springer. 2018, pp. 270–279.

[199] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "A neural algorithm of artistic style". In: *arXiv preprint arXiv:1508.06576* (2015).

[200] KR Srinath. "Python–the fastest growing programming language". In: *International Research Journal of Engineering and Technology* 4.12 (2017), pp. 354–357.

[201] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. Dec. 2015.

[202] Vineel Nagisetty, Laura Graves, Joseph Scott, and Vijay Ganesh. "xAI-GAN: Enhancing Generative Adversarial Networks via Explainable AI Systems". In: *arXiv preprint arXiv:2002.10438* (2020).

[203] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. "GAN Dissection: Visualizing and Understanding Generative Adversarial Networks". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2019.

[204] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. "Deepfakes and beyond: A survey of face manipulation and fake detection". In: *Information Fusion* 64 (2020), pp. 131–148.

[205] Yaël Frégier and Jean-Baptiste Gouray. "Mind2Mind: transfer learning for GANs". In: *arXiv preprint arXiv:1906.11613* (2019).

[206] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. "Transferring gans: generating images from limited data". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 218–234.

[207] Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. "To learn image super-resolution, use a GAN to learn how to do image degradation first". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Sept. 2018.

[208] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. "Federated learning: Strategies for improving communication efficiency". In: *arXiv preprint arXiv:1610.05492* (2016).

Bibliography

[209] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečný, Stefano Mazzocchi, H Brendan McMahan, et al. "Towards federated learning at scale: System design". In: *arXiv preprint arXiv:1902.01046* (2019).

Appendix A

Published Articles

A.1   Paper I - HyperKvasir, a Comprehensive Multi-class Image and Video Dataset for Gastrointestinal Endoscopy

Authors: Hanna Borgli, **Vajira Thambawita**, Pia H. Smedsrud, Steven Hicks, Debesh Jha, Sigrun L. Eskeland, Kristin Ranheim Randel, Konstantin Pogorelov, Mathias Lux, Duc Tien Dang Nguyen, Dag Johansen, Carsten Griwodz, Håkon K. Stensland, Enrique Garcia-Ceja, Peter T. Schmidt, Hugo L. Hammer, Michael A. Riegler, Pål Halvorsen, Thomas de Lange

Abstract: Artificial intelligence is currently a hot topic in medicine. However, medical data is often sparse and hard to obtain due to legal restrictions and lack of medical personnel for the cumbersome and tedious process to manually label training data. These constraints make it difficult to develop systems for automatic analysis, like detecting disease or other lesions. In this respect, this article presents HyperKvasir, the largest image and video dataset of the gastrointestinal tract available today. The data is collected during real gastro- and colonoscopy examinations at Bærum Hospital in Norway and partly labeled by experienced gastrointestinal endoscopists. The dataset contains 110,079 images and 374 videos, and represents anatomical landmarks as well as pathological and normal findings. The total number of images and video frames together is around 1 million. Initial experiments demonstrate the potential benefits of artificial intelligence-based computer-assisted diagnosis systems. The HyperKvasir dataset can play a valuable role in developing better algorithms and computer-assisted examination systems not only for gastro- and colonoscopy, but also for other fields in medicine.

Candidate contributions: Vajira contributed (as one of the main authors) to the conception and design of the paper and doing the deep learning baseline experiments in this manuscript for the classification task by critically analyzing the data. He developed and analyzed four different deep learning methods (ResNeet-152, DenseNet-161, averaged ResNet-152, DenseNet-161, and combined ResNet-152 and DenseNet-161 through an MLP) using two folds cross-validation and Pytorch deep learning

framework. These deep learning experiments show the best baseline performance of this paper. He contributed to drafting the manuscript and revising it.

Thesis objectives: Sub-objective I, Sub-objective II

Appendix A. Published Articles

SCIENTIFIC DATA

Check for updates

OPEN

DATA DESCRIPTOR

HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy

Hanna Borgli[1,3,15], Vajira Thambawita [1,2,15], Pia H. Smedsrud[1,3,6,15], Steven Hicks[1,2,15], Debesh Jha[1,7,15], Sigrun L. Eskeland[4], Kristin Ranheim Randel[3,10], Konstantin Pogorelov[8], Mathias Lux[11], Duc Tien Dang Nguyen[5], Dag Johansen[7], Carsten Griwodz[3], Håkon K. Stensland[3,8], Enrique Garcia-Ceja [13], Peter T. Schmidt[9,14], Hugo L. Hammer[1,2,15], Michael A. Riegler[1,15,16], Pål Halvorsen [1,2,15,16] & Thomas de Lange [4,6,12,15,16]

Artificial intelligence is currently a hot topic in medicine. However, medical data is often sparse and hard to obtain due to legal restrictions and lack of medical personnel for the cumbersome and tedious process to manually label training data. These constraints make it difficult to develop systems for automatic analysis, like detecting disease or other lesions. In this respect, this article presents *HyperKvasir*, the largest image and video dataset of the gastrointestinal tract available today. The data is collected during real gastro- and colonoscopy examinations at Bærum Hospital in Norway and partly labeled by experienced gastrointestinal endoscopists. The dataset contains 110,079 images and 374 videos, and represents anatomical landmarks as well as pathological and normal findings. The total number of images and video frames together is around 1 million. Initial experiments demonstrate the potential benefits of artificial intelligence-based computer-assisted diagnosis systems. The *HyperKvasir* dataset can play a valuable role in developing better algorithms and computer-assisted examination systems not only for gastro- and colonoscopy, but also for other fields in medicine.

Background & Summary

The human gastrointestinal (GI) tract is subject to numerous different abnormal mucosal findings ranging from minor annoyances to highly lethal diseases. For example, according to the International Agency for Research on Cancer (https://gco.iarc.fr/today/fact-sheets-cancers), the specialized cancer agency of the World Health Organization (WHO), GI cancer globally accounts for about 3.5 million new cases each year. These cancer types usually have combined mortality of about 63% and 2.2 million deaths per year[1–3].

Endoscopy is currently the gold-standard procedure for examining the GI tract, but its effectiveness is considerably limited by the variation in operator performance[4–6]. This causes, for example, an average 20% polyp miss-rate in the colon[7]. Thus, improved endoscopic performances, high-quality clinical examinations, and systematic screening are significant factors to prevent GI disease-related morbidity and deaths. The recent rise of artificial intelligence (AI)-enabled support systems has shown promise in giving healthcare professionals the tools needed to provide quality care at a large scale[8,9]. The core of an efficient AI-based system is the combination of quality data and algorithms which teach a model to solve real-world problems like detecting pre-cancerous lesions or cancers in images. Today's AI-based systems are predominantly using a subfield of AI called machine learning (ML), which usually requires training on thousands of data samples to perform well on any given task.

[1]SimulaMet, Oslo, Norway. [2]Oslo Metropolitan University, Oslo, Norway. [3]University of Oslo, Oslo, Norway. [4]Department of Medical Research, Bærum Hospital, Bærum, Norway. [5]University of Bergen, Bergen, Norway. [6]Augere Medical AS, Oslo, Norway. [7]UIT The Arctic University of Norway, Tromsø, Norway. [8]Simula Research Laboratory, Oslo, Norway. [9]Department of Medicine (Solna), Karolinska Institutet, Stockholm, Sweden. [10]Cancer Registry of Norway, Oslo, Norway. [11]Klagenfurt University, Klagenfurt, Austria. [12]Medical Department, Sahlgrenska University Hospital-Mölndal, Mölndal, Sweden. [13]SINTEF Digital, Oslo, Norway. [14]Department of Medicine, Ersta hospital, Stockholm, Sweden. [15]These authors contributed equally: Hanna Borgli, Vajira Thambawita, Pia H. Smedsrud, Steven Hicks, Debesh Jha, Hugo L. Hammer, Michael A. Riegler, Pål Halvorsen, Thomas de Lange. [16]These authors jointly supervised this work: Michael A. Riegler, Pål Halvorsen, Thomas de Lange. ✉e-mail: paalh@simula.no

Dataset	Findings	Size	Availability
CVC-356[18]	Polyps	356 images†	by request●
CVC-ClinicDB[19] (also named CVC-612)	Polyps	612 images†	open academic
CVC-VideoClinicDB[18] (also named CVC-12k)	Polyps	11954 images†	by request●
CVC-ColonDB[62]	Polyps	380 images†Ψ	by request●
Endoscopy Artifact detection 2019[63]	Endoscopic Artifacts	5,138 images	open academic
ASU-Mayo polyp database[20]	Polyps	18,781 images†	by request●
ETIS-Larib Polyp DB[64]	Polyps	196 images†	open academic
KID[65]◇	Angiectasia, bleeding, inflammations, polyps	2371 images and 47 videos	open academic●
GIANA 2017[66]◇	Polyps & Angiodysplasia	3462 images and 38 videos	by request
GIANA 2018[67,68]◇	Polyps & Small bowel lesions	8262 images and 38 videos	by request
GASTROLAB[69]	GI lesions	Some 100s of images and few videos	open academic♣
WEO Clinical Endoscopy Atlas[70]	GI lesions	152 images	by request♣
GI Lesions in Regular Colonoscopy Data Set[71]	GI lesions	76 images†	by request
Atlas of Gastrointestinal Endoscope[72]	GI lesions	1295 images	unknown●
El salvador atlas of gastrointestinal video endoscopy[73]	GI lesions	5071 video clips	open academic♣
Kvasir[22]	Polyps, esophagitis, ulcerative colitis, Z-line, pylorus, cecum, dyed polyp, dyed resection margins, stool	8000 images	open academic
Kvasir-SEG[49]	Polyps	1000 images†	open academic
Nerthus[74]	Stool - categorization of bowel cleanliness	21 videos	open academic

Table 1. An overview of existing GI datasets. †Including ground truth segmentation masks. ◇Video capsule endoscopy. ●Not available anymore. ΨContour. ♣Not really a dataset usable for machine learning. It is more a medical atlas or database for education with a several low-quality samples of various findings in the GI tract.

However, health data is often sparse and hard to obtain due to legal constraints and structural problems in data collection. Nevertheless, an increasing number of promising AI solutions aimed for diagnostics in endoscopy[10–17] are being developed. The datasets used for these systems, like CVC[18,19] and the ASU-Mayo polyp database[20], are rather small in the context of ML research. In other non-medical ML areas, datasets such as ImageNet[21] consists of 14 million images. Table 1 gives an overview of all, to the best of our knowledge, existing datasets of images and videos from the human GI tract. As can be observed, they are rather small, and often limited to polyps. Several of these have also lately become unavailable.

The images and videos in *HyperKvasir* were collected prospectively from routine clinical examinations performed at a Norwegian hospital from 2008 to 2016. We retrieved the images from the Picsara image documentation database (CSAM, Norway), a plug-in to the electronic medical record system, in 2016. As a first step, 4,000 of these images were labeled into eight different classes by medical experts and published as the Kvasir dataset[22]. The dataset was later extended to 8,000 images. Using Kvasir, researchers all over the world have started developing different ML models and AI systems for GI endoscopy[23–38]. Moreover, the Kvasir dataset has been used to organize international competitions, i.e., the Medico Task at MediaEval in 2017[39] and 2018[40] and the ACM Multimedia 2019 BioMedia Grand Challenge[41].

Based on the lessons learned from publishing the Kvasir dataset and organizing competitions, it became clear that one of the biggest challenges in medical AI is still data availability. Data is hard to retrieve from the health care systems, approvals from medical committees are hard to get, medical experts have limited time, and there are no efficient tools to label such data. Therefore, with *HyperKvasir*, we significantly increase both the amount of labeled medical data for supervised learning and also release a large amount of unlabeled data. The new dataset contains 110,079 images and 374 videos from various GI examinations, resulting in 1 million images and frames in total. Regarding the value of unlabeled data, recent work in the ML community has shown remarkable improvements to tackle the challenge of lack of data. Instead of learning from a large set of annotated data, algorithms can now learn from sparsely labeled and unlabeled data. This technique is known as semi-supervised learning and has lately been successfully applied in different medical image analyses[42]. Examples of semi-supervised learning are self-learning[43,44] and neural graph learning[45], which both make use of unlabeled data in addition to a small number of labeled data to extract additional information[43,44,46]. We believe these new algorithms might be the development needed to make AI even more useful for medical applications. The unlabeled data of *HyperKvasir* is intended to be used in medical and technical communities to explore semi-supervised and unsupervised methods, and users of the data might even consider employing their own local experts to provide labels . Subsequently, in addition to the data description, we provide a baseline analysis using the labeled classes of the dataset and feasible future research directions for researchers interested in using the dataset.

Methods

The image and video data were collected using standard endoscopy equipment from Olympus (Olympus Europe, Germany) and Pentax (Pentax Medical Europe, Germany) at the Department of Gastroenterology, Bærum Hospital, Vestre Viken Hospital Trust, Norway. Vestre Viken provides health care services to 490,000 people, of which 189,000 are covered by Bærum hospital. Parts of the collected data were annotated with class labels and segmentation masks. The annotations were done by at least one experienced gastroenterologist from Bærum

Appendix A. Published Articles

hospital, the Cancer Registry of Norway or Karolinska University Hospital in Sweden, and one or more experienced persons working in the medical field such as a junior doctor or Ph.D. student. Though several physicians have assessed each labeled data record of the dataset, there is a chance that some of the assessments might be biased by the well-known observer variation, particularly regarding subtle changes like low-grade reflux esophagitis and ulcerative colitis. Such discrepancies have been demonstrated in previous studies[47,48]. To tackle this further, we decided to combine some of the findings that are prone to bias into one class (details about the classes and combinations can be found in the data records descriptions). Finally, a large number of unlabeled images are provided.

The study was approved by Norwegian Privacy Data Protection Authority and exempted from patient consent because the data were fully anonymous. All metadata was removed, and all files renamed to randomly generated file names before the internal IT department at Bærum hospital exported the files from a central server. The study was exempted from approval from the Regional Committee for Medical and Health Research Ethics - South East Norway since the collection of the data did not interfere with the care given to the patient. Since the data is anonymous, the dataset is publicly shareable based on Norwegian and General Data Protection Regulation (GDPR) laws. Apart from this, the data has not been pre-processed or augmented in any way.

Class labels per image. The method for labeling images can be split into three distinct steps. First, experienced gastroenterologists involved in the project decided which classes should be included in the labeling process, based on medical relevance and the data collected. The selected classes were described in detail by medical experts. Second, two junior doctors or Ph.D. students working in the field annotated a subset of the images to the provided classes. Once this pre-labeling step was done, the medical experts checked the labels and adjusted when necessary. Cases where no consent could be found were discarded and replaced with new images from the dataset. The first dataset we created consisted of 4,000 images from eight classes[22]. This was later extended to 8,000 images for the same eight classes. For *HyperKvasir*, the dataset is further extended to 10,662 images and 23 classes. In total, *HyperKvasir* contains 110,079 images (10,662 labeled and 99,417 unlabeled images) from the GI tract.

Segmentation masks per image. *HyperKvasir* includes images with corresponding segmentation masks and bounding boxes for 1,000 images from the polyp class. To create the segmentation masks, we uploaded 1,000 polyp images to the Labelbox platform (https://www.labelbox.com/). Labelbox is a tool that allows pixel-accurate labeling of image regions. First, a junior doctor and a Ph.D. student pre-segmented the 1,000 images. A gastroenterologist subsequently went trough all images verifying and correcting the pre-labeled segmentation masks. A detailed description of the annotation process and an example use-case of the dataset can be found in[49,50].

Descriptions per video. To get the labels per video, we uploaded the video data to a video annotation platform provided by Augere Medical AS (Oslo, Norway). Each video was analyzed and labeled by an experienced gastroenterologist. The class labels selected by the gastroenterologist were representing the main finding in the video as accurately as possible. For example, if the video contained footage of a polyp, the label for that video would be polyp. Additionally, there are examples of multiple findings in the same video. If so, these and more detailed descriptions are included in the *video-labeling.csv* file.

Data Records

The full *HyperKvasir*[51] dataset, with all its images, videos and metadata, is available from the Open Science Framework (OSF) via the link https://doi.org/10.17605/OSF.IO/MH9SJ. The dataset is also available at https://datasets.simula.no/hyper-kvasir. *HyperKvasir* is open access and licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0). In total, the dataset consists of four main data records. The records are labeled images, segmented images, unlabeled images, and labeled videos. All the various labeled classes are shown in Fig. 1, i.e., 16 classes from the upper GI tract (Fig. 1a) and 24 classes from the lower GI tract (Fig. 1b). The dataset has a size of circa 66.4GB (not including metadata files and segmentation masks), 32.5GB for videos and 33.9GB for images. An overview of all data records in the dataset is given in Table 2. Some of the images and videos contain a picture in picture (green thumbnail in the lower left corner) which represents the Olympus ScopeGuide™ (Olympus Europe, Germany), used by the endoscopist to get a topographic view of the colon. Details about image and video resolutions and video frame rates can be found in the Figs. 2 and 3. The following subsections provide additional details for each data record.

Labeled images. In total, the dataset contains 10,662 labeled images stored using the JPEG format, where Fig. 4 shows the 23 different classes representing the labeled images and the number of images in each class. A CSV file is provided (image-labels.csv) giving the mapping between the image (file name) and the labeling for each image. These classes are structured according to location in the GI tract and the type of finding as shown in Fig. 5. We defined four main categories of findings where the first and the third are found both in the upper an lower GI tract:

- **Anatomical landmarks:** Anatomical landmarks are characteristics of the GI tract used for orientation during endoscopic procedures. Furthermore, they are used to confirm a complete extent of the examination. Landmarks exist both in the upper GI tract (esophagus, stomach and duodenum) and in the lower GI tract (terminal ileum, colon and rectum). However, in the small bowel, there are no specific landmarks to be used for topographical localization of a lesion.
- **Quality of mucosal views:** Complete visualization of all the mucosa is crucial not to overlook pathological findings. In the colon, there exist a classification for the quality of the mucosal vizualisation, the Boston Bowel Preparation Scale (BBPS)[52].

A.1. Paper I - HyperKvasir, a Comprehensive Multi-class Image and Video Dataset for Gastrointestinal Endoscopy

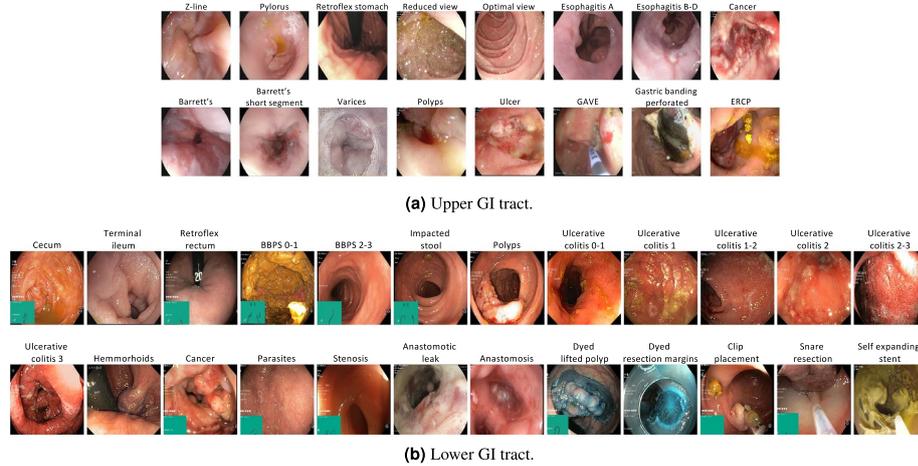**(a)** Upper GI tract.

(b) Lower GI tract.

Fig. 1 Image examples of the various labeled classes for images and/or videos.
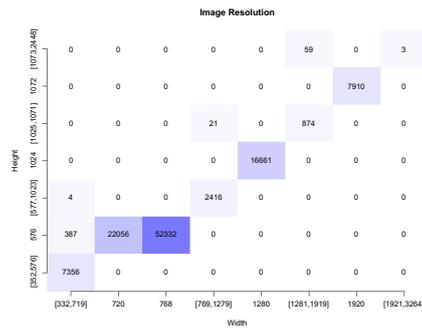
Fig. 2 Resolution of the 110,079 images in *HyperKvasir*.

Data Record	# Files	Description	Size (MB)
Labeled images	10,662 images	23 classes of findings	3,960
Segmented Images	1,000 images	Segmentation masks for polyp findings	57
Unlabeled Images	99,417 images	Unlabeled	29,940
Videos	374 videos	30 different classes	32,539

Table 2. Overview of the data records in the *HyperKvasir* dataset.

- **Pathological findings:** All parts of the gastrointestinal tract can be affected by abnormalities or findings due to disease. Most pathological findings can be seen as more or less obvious changes in the intestinal wall mucosa. These findings are classified according to the Minimal Standard Terminology, defined by the World Endoscopy Organization[53].
- **Therapeutic interventions:** When a lesion or pathological finding is detected, a therapeutic intervention is frequently required to treat the condition, e.g., lifting and resecting a polyp, dilation of a stenosis or injection of a bleeding ulcer.

Each class and the images belonging to it is stored in the corresponding folder of the category it belongs to. For example, the 'polyp' folder in the category pathological findings in the lower GI tract contains all polyp images,
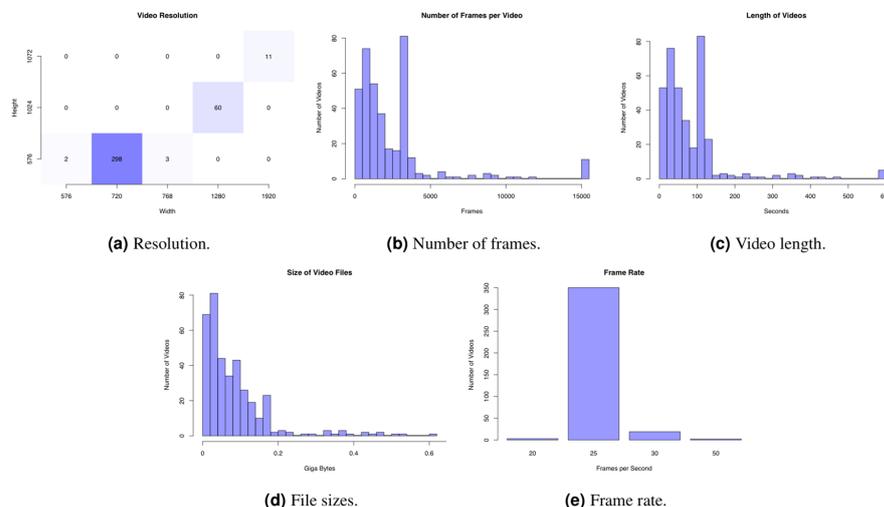
Appendix A. Published Articles

(a) Resolution.   **(b)** Number of frames.   **(c)** Video length.

(d) File sizes.   **(e)** Frame rate.

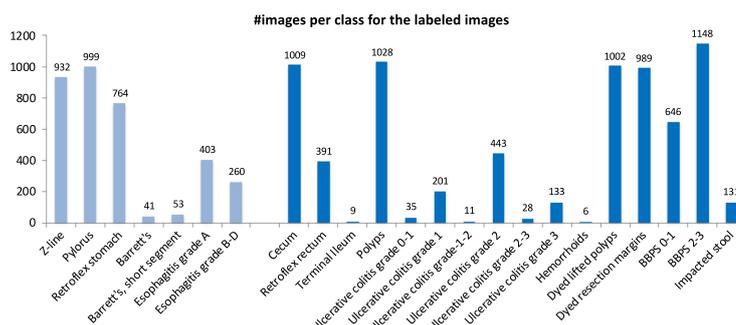**Fig. 3** Statistics of the 374 videos in *HyperKvasir*.

Fig. 4 The number of images in the various *HyperKvasir* labeled image classes according to the file folders.

the 'barrett's' folder in the category pathological findings in the upper GI tract contains all images of Barrett's esophagus, etc. As observed in Fig. 2, the number of images per class are not balanced, which is a general challenge in the medical field due to the fact that some findings occur more often than others. This adds an additional challenge for researchers, since methods applied to the data should also be able to learn from a small amount of training data. Below, we detail each class further.

Upper Gastrointestinal tract.   The upper GI tract examined by endoscopy includes the esophagus, stomach, and duodenum. Below, we give a description of the various classes of findings found here.

As seen in Fig. 5, we have labeled three classes of *anatomical landmarks* in the upper GI tract. The normal **Z-line** is the anatomical junction between the squamous epithelium of the esophagus and columnar epithelium of the stomach. A normal Z-line is located at the same level as the gastroesophageal junction. **Retroflex stomach** means that the endoscope is retroflexed, looking back to visualize the cardia and fundus in the upper parts of the stomach. The **pylorus** is the anatomical junction between the stomach and duodenal bulb, and a sphincter regulating the emptying process of the stomach into the duodenum.

All the following classes are defined as *pathological findings* in the upper GI tract. Reflux esophagitis is an inflammation mostly affecting the lower third of the esophagus, near the Z-line. Reflux esophagitis can be graded according to the Los Angeles (LA) classification[54]. The esophagitis LA classification is defined into four classes as (A) mucosal breaks shorter than 5mm, without continuity across mucosal folds where subtle changes can be difficult to differentiate from a normal Z-line; (B) mucosal breaks longer than 5mm that does not extend between the
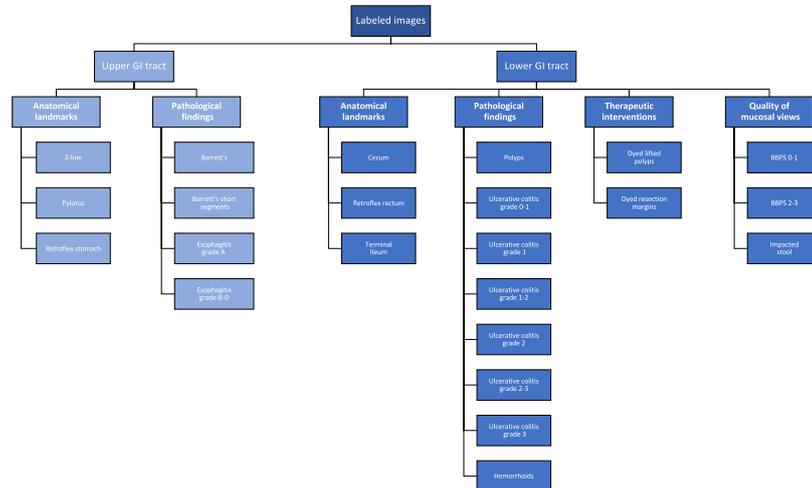
Fig. 5 The various image classes structured under position and type, also the structure of the stored images.

tops of two mucosal folds; (C) one (or more) mucosal break that is continuous between the tops of two or more mucosal folds, but which involves less than 75% of the circumference; and (D) one (or more) mucosal break that is continuous between the tops of two or more mucosal folds and involves more than 75% of the circumference. We have split esophagitis into two classes because there exists an important observer variation in the assessment of low grade esophagitis[47]. The two classes are **esophagitis A** and **esophagitis B-D**. This binary classification of the images makes it possible to assess whether mis-classification between normality and esophagitis only concern grade A. Barrett's esophagus represents a metaplastic transformation of the squamous epithelium of the esophagus into a gastric like columnar epithelium. Barrett's esophagus is considered a premalignant condition, meaning it might develop into cancer. Biopsies showing the presence of specialized intestinal metaplasia confirms the diagnosis. Barrett's esophagus can be graded according to the Prague classification, describing the circumferential and longitudinal extension of the disease[55]. We have split the images of Barrett's esophagus into two classes. **Barrett's** long-segment and **Barrett's, short-segment** esophagus where a short segment is characterized by a longitudinal extension of less than 3 cm[55].

Lower gastrointestinal tract. The lower GI tract examined by colonoscopy includes the terminal ileum (last part of the small bowel), the colon and the rectum (the large bowel). Below, we describe the classes of the lower GI tract in the dataset.

We have labeled three classes of *anatomical landmarks* in the lower GI tract. The ileum is the distal 2/3 of the small bowel, recognized by visible intestinal villi. Endoscopically, the ileum cannot be distinguished from other parts of the small bowel. During colonoscopy, the distal 5–20 cm of the ileum, named **terminal ileum**, can be reached and examined. The visualization of the terminal ileum confirms complete colonoscopy. **Cecum** is the proximal end of the large bowel and is characterized by the visualization of the appendiceal orifice and the ileo-cecal valve. Complete examination of the whole colon can only be confirmed if the medial wall of the cecum has been visualized, that is the area between the appendiceal orifice and the ileo-cecal valve. The most distal part of the rectum is one of the blind zones of the colon. Therefore, the endoscope is retroflexed in the rectum to visualize the dentate line and the circumference of the proximal orifice of the anal canal, which is called **retroflex rectum**.

The *quality of the mucosal views* is a key quality indicator and should always be evaluated because a clean bowel is essential to detect pathological findings. In this respect, the degree of bowel cleansing during a colonoscopy is described by the Boston Bowel Preparation Scale (BBPS)[56]. BBPS consists of four different degrees which are: (BBPS 0) unprepared colon segment with no mucosa seen due to solid stool that cannot be cleared; (BBPS 1) portions of the mucosa of the colon segment seen, but other areas of the colon segment not well seen due to staining, residual stool and/or opaque liquid; (BBPS 2) minor amount of small fragments of stool and/or opaque liquid, but mucosa of colon segment seen well; and (BBPS 3) entire mucosa of colon segment seen well with no residual fragments of stool or opaque liquid. The bowel cleansing is deemed adequate if the BBPS score is 2 or 3 in all three segments of the colon after flushing. Therefore, we have labeled our images into the two **BBPS 0-1** and **BBPS 2-3** classes where class 0–1 represents inadequate bowel preparations, and the class 2–3 represents adequate bowel preparation. Moreover, a frequent finding in persons above the age of 50 years are pockets in the colon wall called diverticula and if numerous called diverticulosis. Sometimes stool is impacted in these diverticula and may increase the risk of diverticulitis. In the dataset, this is presented in the **impacted stool** class.

Appendix A. Published Articles

The following classes are defined as *pathological findings* in the lower GI tract. Ulcerative colitis is a chronic inflammatory bowel disease often debuting in the twenties. The degree and extent of the disease is determined by colonoscopy and can be classified according to the Mayo Score[57]. The Mayo Score for ulcerative colitis is defined: (Score 0) inactive, where the mucosa only has normal vascular patterns; (Score 1) mild with erythema, decreased vascular pattern, mild friability; (Score 2) moderate with erythema, absent vascular pattern, mild friability, erosions; and (Score 3) severe with spontaneous bleeding and ulcerations. For ulcerative colitis, we provide six different labeled classes, both the Mayo Score classes (**Ulcerative colitis 1/2/3**) and some classes in-between where it is difficult to determine the exact class and because previous studies have shown important observer variation in the assessment of the degree of inflammation (**Ulcerative colitis 0-1/1-2/2-3**)[48]. **Polyps** are most frequently neoplastic lesions of the large bowel. They have mainly three different shapes; protruding in the lumen, flat or excavated according to the Paris Classification[58]. Their size vary from 1 mm to several cm. The prevalence increases with age. The most common types of polyps are premalignant and can transform into cancer. Thus, it is important to discover polyps and remove the suspicious polyps during endoscopy. **Hemorrhoids** are pathologically swollen veins in the anus or lower rectum. When present in the rectum, they are called internal hemorrhoids, and when found in the anus, they are called external hemorrhoids.

Finally, *therapeutic interventions* show treatments of detected pathological findings. It includes for example lifting and removal of neoplastic tissue (polyps) and injection therapy of bleeding ulcer. The **dyed lifted polyps** class contains images of polyps lifted with submucosal injection using a solution containing indigo carmine. This is done prior to polyp resection for better diagnosis and easier resection. The dye is recognized by the blue color underneath the polyp. After resection of dyed polyps with a snare, the resection margins and site appears blue due to the indigo carmine solution. Images of these type of resection margin are presented in the **dyed resection margins** class.

Segmented images.   For the set of segmented images, we provide the original image, a segmentation mask and a bounding box for 1,000 images from the polyp class. In the mask, the pixels depicting polyp tissue, the region of interest, are represented by the foreground (white mask), while the background (in black) does not contain polyp pixels. The bounding box is defined as the outermost pixels of the found polyp. For this segmentation set, we have two folders, one for images and one for masks, each containing 1,000 JPEG-compressed images. The bounding boxes for the corresponding images are stored in a JavaScript Object Notation (JSON) file. The image and its corresponding mask have the same filename. The images and files are stored in the segmented images folder. It is important to point out that the images of polyp class from the Kvasir dataset had duplicates in the images folder. These duplicates were replaced by high-quality polyp images from the colon and segmented.

Unlabeled images.   In total, the dataset contains 99,417 unlabeled images. The unlabeled images can be found in the unlabeled folder which is a subfolder in the image folder, together with the other labeled image folders. In addition to the unlabeled image files, we also provide the extracted global features and possible unsupervised clustering assignments in the *HyperKvasir* Github repository as Attribute-Relation File Format (ARFF) files. ARFF files can be opened and processed using, for example, the WEKA machine learning library, or they can easily be converted into Comma-Separated Values (CSV) files.

Labeled videos.   The labeled videos are recorded for clinical purposes and thus represent daily practice. In total, 374 videos are provided in the dataset, which correspond to 9.78 hours of videos and 889,372 video frames that can be converted to images if needed. In total, an experienced gastroenterologist have identified 30 classes of findings, and Fig. 6 shows how many videos we have identified for each class. The class describes the video as a whole using the main finding, but additionally, many videos include more than one category and several classes where, for example, a single video can contain polyps, dyed lifted polyps and dyed resection margins. The video file format is Audio Video Interleave (AVI), and they are stored in the folder called labeled videos. As seen in Fig. 7, the videos are further organized and stored according to either upper or lower GI tract and then the four main categories as for the labeled images described above. In addition to the video files, a CSV file is provided (video-labels.csv) containing the videos' *videoID* and *labeling*. Here, the VideoID contains the corresponding video file name, and the labeling includes the upper or lower location, the category and the class with some detailed descriptions of the video. Below, we describe the new classes per category for the in total 60 videos from the upper GI tract and the 60 videos from the lower GI tract.

Upper Gastrointestinal tract.   As seen in Fig. 7, we have many of the same classes for videos and for images, but since we have labeled all our videos, more classes are added for both the upper and lower GI-tract. In the upper GI tract, the three classes of *anatomical landmarks* (**Z-line**, **Pylorus** and **Retroflex stomach**) are described in the section for labeled images above. In the category of *pathological findings*, both **Barrett's** esophagus and **esophagitis** are also described above, but here we also added some new classes. The first is **polyps** where the description above of polyps in the colon is also valid for the upper GI-tract. In addition, five new classes not previously described are included. Mucosal **ulcers** are quite common in the upper GI tract. Ulcers are nearly always caused by Helicobacter pylori infection, ulcerogenic medication, or cancer. Ulcers are characterized according to the Forrest classification to predict the risk of bleeding[59]. Forrest I represents ongoing bleeding, Forrest II presents some signs of previous bleeding; and Forrest III does not show any sign of bleeding. The second class **Gastric antral vascular ectasia (GAVE)** represents dilated small superficial vessels in the mucosa of the gastric antrum. These lesions may cause chronic bleeding and subsequent anemia and are frequently treated by argon plasma coagulation (APC) to prevent further bleeding. **Varices** (dilated veins) in both the esofagus and the fundus of the stomach are most frequently caused by chronic liver diseases complicated with liver cirrhosis. The varices represent a major risk for severe bleeding. **Cancer** of the esophagus and the stomach are common findings in the upper
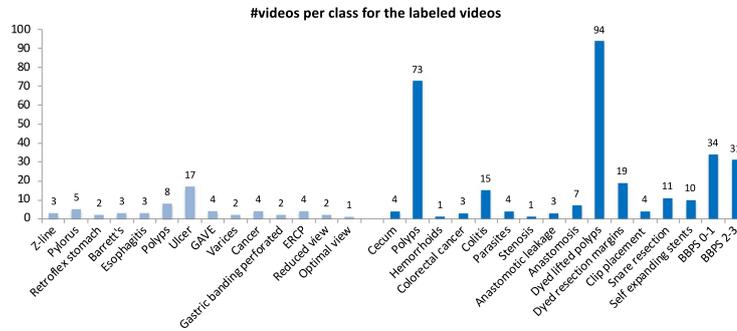
Fig. 6 The number of videos in the various *HyperKvasir* labeled video classes according to the file folders.
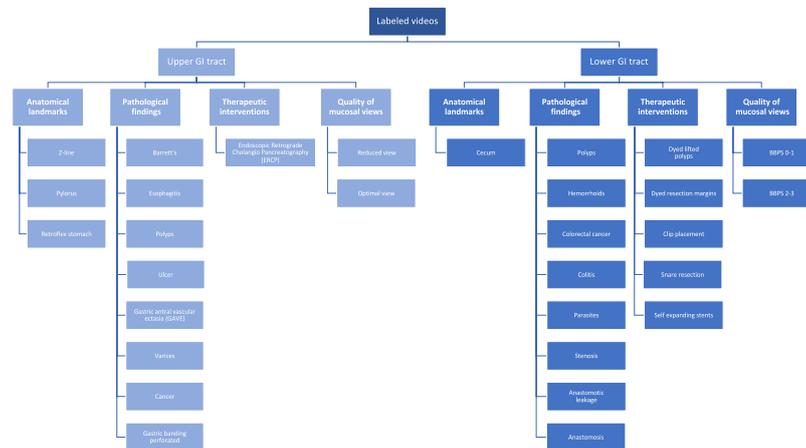

Fig. 7 The various video classes structured under position and type, which is also the structure of the video folders.

GI-tract. The last class **gastric banding perforated** shows a rare finding, which is the complication of previous gastric banding operation where the band perforates the wall of the stomach. The category of *therapeutic interventions* are introduced for the Upper GI-tract especially because they are nearly always best illustrated by videos and can also serve important educational purposes. Since most of the *therapeutic interventions* are presented as secondary to a pathological finding we only include **Endoscopic Retrograde Choleangio-Pancreatografi (ERCP)** a procedure to treat gall-duct abnormalities as an independent class. However, other common therapeutic interventions such as the two thermal methods; APC and heatherprobe as well as injection therapy with adrenaline and application of hemospray to stop bleeding can be found under second findings in the csv file. In the category *quality of mucosal view*, we also added a footage showing **reduced view** due to opaque liquid in the stomach or air bubbles in the duodenum. Reduced view increases the risk of missing lesions. In opposite, **optimal view** demonstrates excellent visualization of the duodenum.

Lower Gastrointestinal tract. The videos from the lower GI tract illustrate mainly the same categories and classes as the labeled images. Nevertheless, they increase the diversity of the dataset. The category *anatomical landmarks* differs from the labeled images as it only contains the **cecum** class and does not include the classes of terminal ileum and retroflex rectum, only defined as second findings. The two categories pathological findings and therapeutic intervention also are a bit different compared to the labeled images. In the category *pathological findings*, we still have the above described **polyps** and **hemorrhoids** classes. However, all classes of ulcerative colitis are merged to **colitis** and also includes ischemic colitis and infectious colitis. The new class **colorectal cancer**, the second most deadly cancer worldwide[60], was added. Colorectal cancer may present itself in different ways in the colon, from tiny lesions with a diameter of 1 cm to larger tumors obstructing the entire lumen of the bowel and

Appendix A. Published Articles

cover bowel segments of several centimeters. Moreover, **parasites**, a common finding of small worms moving around in the colon, are more often encountered in tropical areas. **Stenosis** is characterized by a narrow obstruction of the bowel caused either by inflammation or malignant diseases. Large neoplastic lesions like cancers are surgically resected and subsequently an **anastomosis** is made to restore normal bowel function. The anastomosis can be visualized during follow-up colonoscopies. A feared complication after large bowel surgery is **anastomotic leakage**, potentially causing smaller or larger cavities of anastomotic leak especially in the rectum. The last decade mini-invasive endoscopic *therapeutic interventions* has to some extent replaced traditional and laparoscopic surgery regarding the treatment of large polyps and stenosis of the colon. The classes **dyed lifted polyp** and **dyed resection margin** are described under labeled images but videos improve the illustration of the technique. Three new classes are presented showing removal of polyps by simple **snare resection** or endoscopic mucosal resection (EMR). To prevent or stop bleeding after these resections, **clip placement** of metallic clips are illustrated. **Self expanding stents** are used to open and dilate either benign or malignant stenosis. Finally, in the *quality of mucosal views* category, we have removed the impacted stool class we have for images, and include only the above described **BBPS 0-1** and **BBPS 2-3** classes. Here, it is also worth noting that many of the videos in BBPS 2-3 are perfectly clean (BBPS 3), i.e., as then described in the csv-file, these contain videos of normal mucosa (also marked as finding 2) which can be extracted in normal images or videos when needed.

Technical Validation

To demonstrate the technical quality of the dataset, we performed multiple experiments to provide some baseline metrics and to give some insight into the dataset's statistical qualities. If the reader wants information about classification and segmentation approaches and experiments comparing state of the art methods using parts of this dataset, the reader is referred to other studies[49].

Baseline for supervised image classification. The presented dataset is suited for a variety of different tasks, one of which is image classification. As a preliminary step to evaluate how state-of-the-art methods perform on the labeled part of *HyperKvasir*, we performed a series of experiments based on methods that have previously achieved good results on GI tract image classification. The purpose of these experiments is merely to give example baseline results to be used by future researches to compare and measure their results. In total, we ran five experiments using different methods. The methods were primarily selected from the best performing methods presented at the MediaEval Medico Task[39,40]. Each method is based on deep convolutional neural networks, which is currently state-of-the-art within image classification. Common for all experiments is that the images were resized to 224×224 before being fed into the networks. All networks are based on common architectures, slightly modified to accommodate our task of classifying 23 different classes of images. The specifics of each method is further explained below:

- *Pre-Trained ResNet-50* is a TensorFlow implementation of the ResNet-50 architecture using ImageNet initialized weights. The network was trained in two steps. First, an initial training over 7 epochs, and then a fine-tuning step over 3 epochs which only trained the layers after the 100th index. Images were loaded using a batch size of 32, and the weights were optimized using Adam with a learning rate of 0.001.
- *Pre-Trained ResNet-152* is a PyTorch implementation of the ResNet-152 architecture using ImageNet initialized weights. The network was trained over 50 epochs using a batch size of 32, and optimized using Stochastic gradient descent (SGD) with a learning rate of 0.001. No fine-tuning was used for this method.
- *Pre-Trained DenseNet-161* is a PyTorch implementation of the standard DenseNet-161 architecture using ImageNet initialized weights. The network was trained over 50 epochs using a batch size of 32, and optimized using SGD with a learning rate of 0.001. No fine-tuning was used for this method.
- *Averaged ResNet-152 + DenseNet-161*[38,61] is an approach that combines the ResNet-152 and DenseNet-161 approach by averaging the output of both models as the final prediction. Both models were trained simultaneously by backpropagating the averaged loss through both models. Overall, the networks were trained for 50 epochs using a batch size of 32. SGD was used to optimize the weights with a learning rate of 0.001. Both the ResNet-152 and DenseNet-161 models were initialized using the best weights of the above Pre-Trained ResNet-152 and Pre-Trained DenseNet-161 implementations.
- *ResNet-152 + DenseNet-161 + MLP*[38,61] is similar to the previous method using both ResNet-152 and DenseNet-161 to generate a prediction. However, instead of averaging the output of each model, this method uses a simple multilayer perceptron (MLP) to estimate the best way to average the output of each model. All networks were trained simultaneously over 50 epochs using a batch size of 32. The weights were optimized using SGD with a learning rate of 0.001. Both the ResNet-152 and DenseNet-161 models were initialized using the best weights of the above two implementations of Pre-Trained ResNet-152 and Pre-Trained DenseNet-161.

Each method was evaluated using standard classification metrics including the macro-averaged and micro-averaged F1-score, precision, and recall. Additionally, we calculated the Matthews correlation coefficient (MCC) for each experiment using the multi-class generalization which is also known as the R_K. The results in Table 3 show that each method beats the random and majority class baseline by a large margin. However, the presented numbers also indicate that there is room for improvement. Looking at the confusion matrices in Fig. 8, we see that some classes are harder to identify than others. For example, there is a lot of confusion surrounding the difference between the grades of ulcerative colitis and esophagitis. Furthermore, there is also some confusion between specific classes such as dyed lifted polyps and dyed resection margins, and distinguishing Barrett's from esophagitis or a normal Z-line. At least the confusion between classes of Z-line, esophagitis and Barrett's esophagus is similar to the human variation in the assessment of these lesions. Thus, it is challenging to create a ground truth.

A.1. Paper I - HyperKvasir, a Comprehensive Multi-class Image and Video Dataset for Gastrointestinal Endoscopy

Method	Macro Average			Micro Average			
	Precision	Recall	F1-score	Precision	Recall	F1-score	MCC (R_K)
Pre-Trained ResNet-50	0.589	0.536	0.530	0.839	0.839	0.839	0.826
Pre-Trained ResNet-152	0.639	0.605	0.606	0.906	0.906	0.906	0.898
Pre-Trained DensNet-161	0.640	0.616	0.619	0.907	0.907	0.907	0.899
Averaged ResNet-152 + DenseNet-161	**0.633**	**0.615**	**0.617**	**0.910**	**0.910**	**0.910**	**0.902**
ResNet-152 + DenseNet-161 + MLP	0.612	0.606	0.605	0.909	0.909	0.909	0.902
Random Guessing	0.044	0.038	0.034	0.044	0.044	0.044	0.000
Majority Class	0.004	0.043	0.008	0.108	0.108	0.108	N/A

Table 3. Average results for the five tested classification approaches, i.e., average of the results for the two splits.

(a) *Averaged ResNet-152 + DenseNet-161* evaluated on split 0.

(b) *Averaged ResNet-152 + DenseNet-161* evaluated on split 1.

(c) *Pre-Trained DenseNet-161* evaluated on split 0.

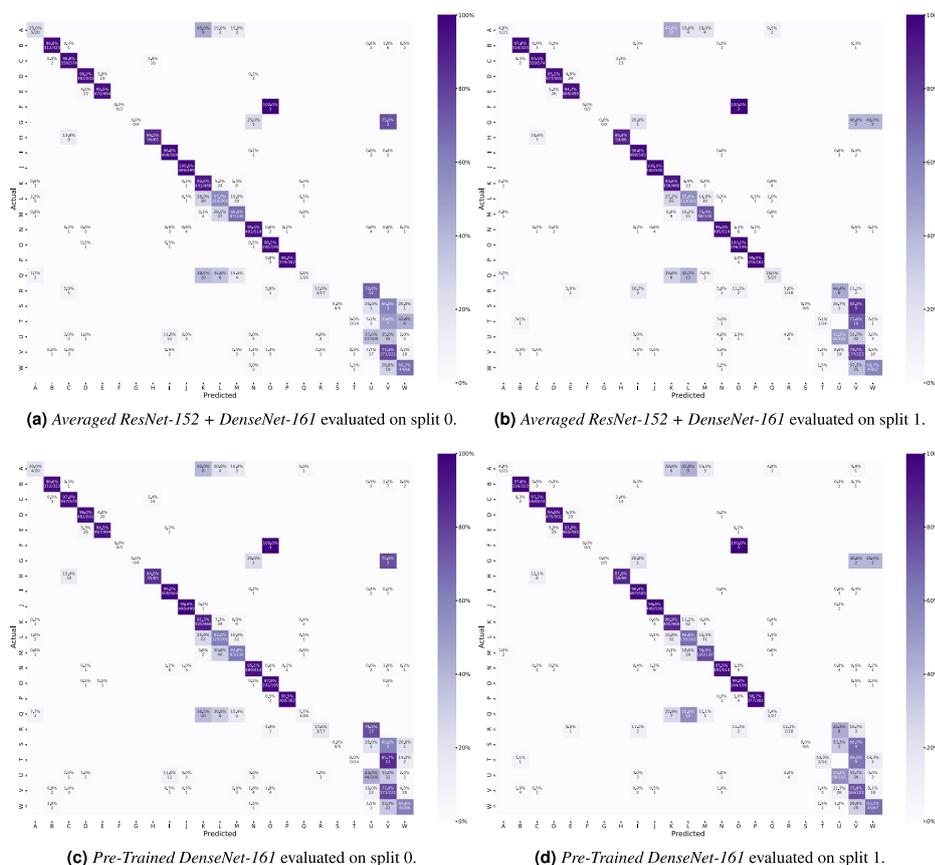**(d)** *Pre-Trained DenseNet-161* evaluated on split 1.

Fig. 8 Confusion matrices for Averaged ResNet-152 + DenseNet-161 and Pre-Trained DenseNet-161 including both splits. These confusion matrices were selected based on their performance. Averaged ResNet-152 + DenseNet-161 achieved the best micro-averaged results while the Pre-Trained DenseNet-161 achieved the best macro-averaged result. The color codes represent the percentages of the total number of images within each class. The labeling of the classes is as follows: (A) Barrett's; (B) bbps-0-1; (C) bbps-2-3; (D) dyed lifted polyps; (E) dyed resection margins; (F) hemorrhoids; (G) ileum; (H) impacted stool; (I) normal cecum; (J) normal pylorus; (K) normal Z-line; (L) oesophagitis-a; (M) oesophagitis-b-d; (N) polyp; (O) retroflex rectum; (P) retroflex stomach; (Q) short segment Barrett's; (R) ulcerative colitis grade 0-1; (S) ulcerative colitis grade 1-2; (T) ulcerative colitis grade 2-3; (U) ulcerative colitis grade 1; (V) ulcerative colitis grade 2; (W) ulcerative colitis grade 3.
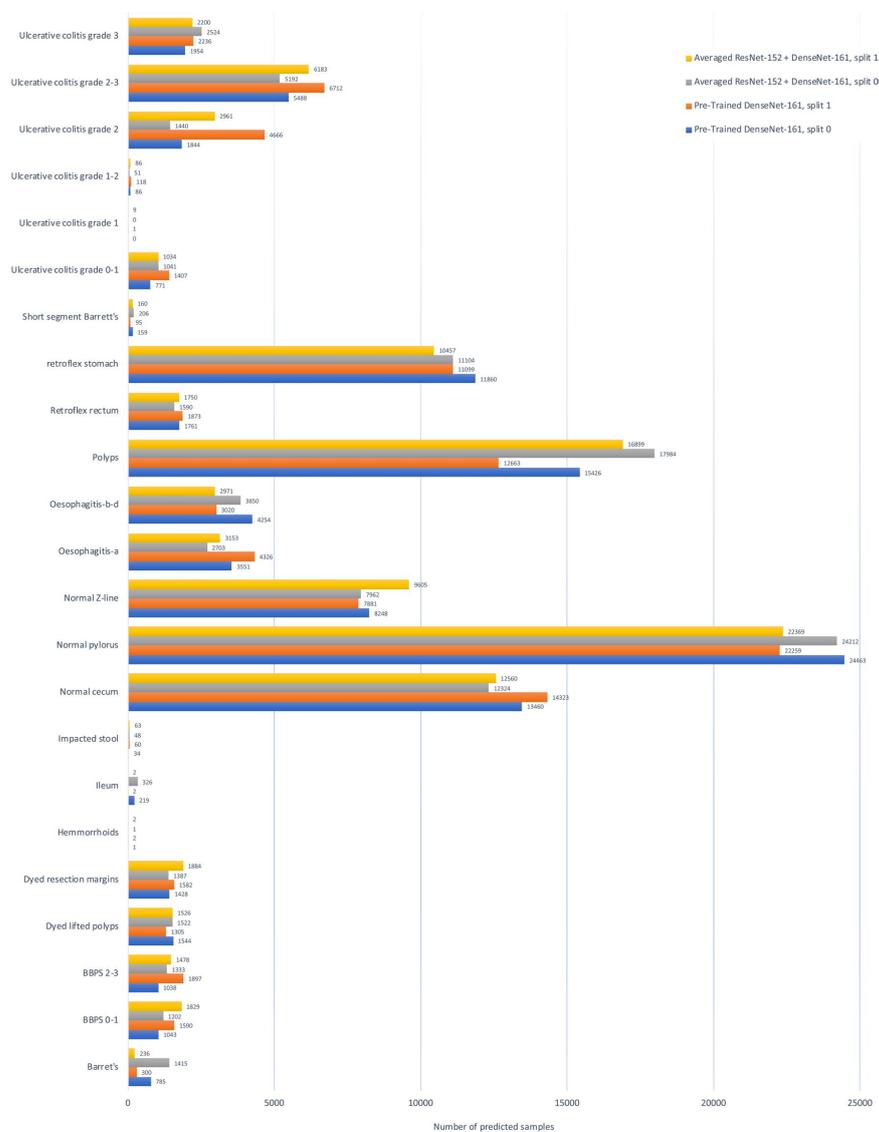
Appendix A. Published Articles

Fig. 9 Unlabeled image data predictions for *Averaged ResNet-152 + DenseNet-161* and *Pre-Trained DenseNet-161*.

Composition of unlabeled data. In order to show the approximate composition of the unlabelled data, we present some initial experiments to analyze the provided data which do not have annotated labels from medical experts We used our pre-trained classification model to simply classify the unlabeled data to indicate how many of the labeled classes are in the unlabeled data and to get an overall idea about data distribution of the 99,417 images. In particular, we used the best two classification models from the previous experiments, i.e., Pre-Trained DenseNet-161 and Averaged ResNet-152 + DenseNet-161 using split_0 and split_1 from the previous experiment. The result are shown in Fig. 9. In the results, we can observe that a large number of predictions are assigned to the class normal pylorus, while a smaller number of predictions are assigned to the classes hemorrhoids and ulcerative colitis grade 1-2. However, these predictions are similar to that of the class-level accuracies of the ML

142

model on the labeled data. Therefore, we can assume that the classes which achieved a high number of correct predictions on the labeled images are also more accurate on the unlabeled data. In contrast, it is hard to make any conclusions on the labels which had a low number of predictions as the models are not accurate enough. For future work, researchers could go trough the classifications of the unlabeled data and, for example, create a larger labeled dataset or perform failure analysis to find out why classes were confused or miss-classified. The class labels created during this experiments are available in the GitHub repository.

Validation Summary
In the technical validation section, we provided baseline metrics and gave insight into the dataset's statistical qualities to demonstrate it's technical quality. With the large number of images available in *HyperKvasir*, we encourage other researchers to investigate and develop new and improved methods for the medical domain. This also includes an improved methodology for creating the ground truth in classes where there is a substantial inter-observer variation in the assessment, which might be used by other researchers to increase the number of labels and segmentations for the dataset.

Usage Notes
In our research on detecting, classifying, and segmenting normal and abnormal findings in the GI tract, we have collected, to the best of our knowledge, the largest and most diverse dataset. These data are made available as a resource to the research community enabling researchers not only to have the ability to research the detection or classification of various GI findings but also differentiate between severity of the findings.

In short, we have used the labeled data to research the classification and segmentation of GI findings using both computer vision and ML approaches to potentially be used in live and post-analysis of patient examinations. Areas of potential utilization are analysis, classification, segmentation, and retrieval of images and videos with particular findings or particular properties from the computer science area. The labeled data can also be used for teaching and training in medical education. Having expert gastroenterologists providing the ground truths over various findings, *HyperKvasir* provides a unique and diverse learning set for future clinicians. Moreover, the unlabeled data is well suited for semi-supervised and unsupervised methods, and, if even more ground truth data is needed, the users of the data can use their own local medical experts to provide the needed labels. Finally, the videos can in addition be used to simulate live endoscopies feeding the video into the system like it is captured directly from the endoscopes enable developers to do image classification.

The dataset includes a series of scripts and text files that aim to help researchers quickly get started using the dataset for standard ML tasks such as classification. These are available at the GitHub repository for the dataset: http://www.github.com/simula/hyper-kvasir. Moreover, we provide three official splits of the dataset that can be used for cross-validation experiments. Keeping splits consistent between methods helps maintain a fair comparison of results. The scripts used to generate the plots, split data into different folds, and generate annotation files are included for reproducibility and transparency. These files may also be used to further experiment with the dataset. Finally, we include the files used to create our preliminary experiments.

There is currently a lot of research being performed in the field of GI image and video analysis, and we welcome and encourage future contributions in this area. This is not limited to using the dataset for comparisons and reproducibility of experiments, but also publishing and sharing new data in the future.

Code availability
In addition to releasing the data, we also make available the code used in the experiments. All code and additional data required for the experiments are available on GitHub at http://www.github.com/simula/hyper-kvasir.

References
1. Brenner, H., Kloor, M. & Pox, C. P. Colorectal cancer. *The Lancet* **383**, 1490–502, https://doi.org/10.1016/S0140-6736(13)61649-9 (2014).
2. Torre, L. A. *et al.* Global cancer statistics, 2012. *CA: A Cancer J. for Clin.* **65**, 87–108, https://doi.org/10.1056/NEJMoa0907667 (2015).
3. World Health Organization - International Agency for Research on Cancer. Estimated Cancer Incidence, Mortality and Prevalence Worldwide in 2012 (2012).
4. Hewett, D. G., Kahi, C. J. & Rex, D. K. Efficacyandeffectivenessofcolonoscopy: howdowebridgethegap? *Gastrointest. Endosc. Clin.* **20**, 673–684, https://doi.org/10.1016/j.giec.2010.07.011 (2010).
5. Lee, S. H. *et al.* Endoscopic experience improves interobserver agreement in the grading of esophagitis by los angeles classification: conventional endoscopy and optimal band image system. *Gut liver* **8**, 154, https://doi.org/10.5009/gnl.2014.8.2.154 (2014).
6. Van Doorn, S. C. *et al.* Polyp morphology: an interobserver evaluation for the paris classification among international experts. *The Am. J. Gastroenterol.* **110**, 180–187, https://doi.org/10.1038/ajg.2014.326 (2015).
7. Kaminski, M. F. *et al.* Quality indicators for colonoscopy and the risk of interval cancer. *New Engl. J. Medicine* **362**, 1795–1803, https://doi.org/10.1056/NEJMoa0907667 (2010).
8. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nat. Medicine* **25**, 44–56, https://doi.org/10.1038/s41591-018-0300-7 (2019).
9. Riegler, M. *et al.* Multimedia and medicine: Teammates for better disease detection and survival. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 968–977, https://doi.org/10.1145/2964284.2976760 (2016).
10. Riegler, M. *et al.* EIR - efficient computer aided diagnosis framework for gastrointestinal endoscopies. In *Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, 1–6, https://doi.org/10.1109/CBMI.2016.7500257 (2016).
11. Alammari, A. *et al.* Classification of ulcerative colitis severity in colonoscopy videos using cnn. In *Proceedings of the ACM International Conference on Information Management and Engineering (ACM ICIME)*, 139–144, https://doi.org/10.1145/3149572.3149613 (2017).
12. Wang, Y., Tavanapong, W., Wong, J., Oh, J. H. & De Groen, P. C. Polyp-alert: Nearreal-timefeedbackduringcolonoscopy. *Comput. Methods Programs Biomed.* **120**, 164–179, https://doi.org/10.1016/j.cmpb.2015.04.002 (2015).
13. Hirasawa, T., Aoyama, K., Fujisaki, J. & Tada, T. 113 application of artificial intelligence using convolutional neural network for detecting gastric cancer in endoscopic images. *Gastrointest. Endosc.* **87**, AB51, https://doi.org/10.1016/j.gie.2018.04.025 (2018).

Appendix A. Published Articles

14. Wang, L., Xie, C. & Hu, Y. Iddf2018-abs-0260 deep learning for polyp segmentation. *Gut* **67**, A84–A85, https://doi.org/10.1136/gutjnl-2018-IDDFabstracts.181 (2018).
15. Mori, Y. *et al.* Real-Time Use of Artificial Intelligence in Identification of Diminutive Polyps During Colonoscopy: A Prospective Study. *Annals Intern. Medicine* **169**, 357–366, https://doi.org/10.7326/M18-0249 (2018).
16. Bychkov, D. *et al.* Deep learning based tissue analysis predicts outcome in colorectal cancer. *Sci. Reports* **8**, 3395, https://doi.org/10.1038/s41598-018-21758-3 (2018).
17. Min, M. *et al.* Computer-aided diagnosis of colorectal polyps using linked color imaging colonoscopy to predict histology. *Sci. reports* **9**, 2881, https://doi.org/10.1038/s41598-019-39414-7 (2019).
18. Bernal, J. & Aymeric, H. Miccai endoscopic vision challenge polyp detection and segmentation. https://endovissub2017-giana.grand-challenge.org/home/, Accessed: 2017-12-11 (2017).
19. Bernal, J. *et al.* Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Comput. Med. Imaging Graph.* **43**, 99–111, https://doi.org/10.1016/j.compmedimag.2015.02.007 (2015).
20. Tajbakhsh, N., Gurudu, S. R. & Liang, J. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Med. Imaging* **35**, 630–644, https://doi.org/10.1109/TMI.2015.2487997 (2016).
21. Deng, J. *et al.* ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255, https://doi.org/10.1109/CVPR.2009.5206848 (2009).
22. Pogorelov, K. *et al.* Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the ACM Multimedia Systems Conference (ACM MMSYS)*, 164–169, https://doi.org/10.1145/3083187.3083212 (2017).
23. Pogorelov, K. *et al.* Deep learning and hand-crafted feature based approaches for polyp detection in medical videos. In *Proceedings of the IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, 381–386, https://doi.org/10.1109/CBMS.2018.00073 (2018).
24. Berstad, T. J. D. *et al.* Tradeoffs using binary and multiclass neural network classification for medical multidisease detection. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, 1–8, https://doi.org/10.1109/ISM.2018.00009 (2018).
25. de Lange, T., Halvorsen, P. & Riegler, M. Methodology to develop machine learning algorithms to improve performance in gastrointestinal endoscopy. *World J. Gastroenterol.* **24**, 5057–5062, https://doi.org/10.3748/wjg.v24.i45.5057 (2018).
26. Hicks, S. *et al.* 383 deep learning for automatic generation of endoscopy reports. *Gastrointest. Endosc.* **89**, AB77, https://doi.org/10.1016/j.gie.2019.04.053 (2019).
27. Ahmad, J., Muhammad, K., Lee, M. Y. & Baik, S. W. Endoscopic image classification and retrieval using clustered convolutional features. *J. Med. Syst.* **41**, 196, https://doi.org/10.1007/s10916-017-0836-y (2017).
28. Owais, M., Arsalan, M., Choi, J., Mahmood, T. & Park, K. R. Artificial intelligence-based classification of multiple gastrointestinal diseases using endoscopy videos for clinical diagnosis. *J. Clin. Medicine* **8**, 986, https://doi.org/10.3390/jcm8070986 (2019).
29. Ahmad, J., Muhammad, K. & Baik, S. W. Medical image retrieval with compact binary codes generated in frequency domain using highly reactive convolutional features. *J. Med. Syst.* **42**, 24, https://doi.org/10.1007/s10916-017-0875-4 (2017).
30. Harzig, P., Einfalt, M. & Lienhart, R. Automatic disease detection and report generation for gastrointestinal tract examination. *Proceedings of the ACM International Conference on Multimedia (ACM MM)* **5**, 2573–2577, https://doi.org/10.1145/3343031.3356066 (2019).
31. Kasban, H. & Salama, D. H. A robust medical image retrieval system based on wavelet optimization and adaptive block truncation coding. *Multimed. Tools Appl.* **78**, 35211–35236, https://doi.org/10.1007/s11042-019-08100-3 (2019).
32. Ghatwary, N., Zolgharni, M. & Ye, X. Gfd faster r-cnn: Gabor fractal densenet faster r-cnn for automatic detection of esophageal abnormalities in endoscopic images. *International Workshop on Machine Learning in Medical Imaging (MLMI)* **11861**, 89–97, https://doi.org/10.1007/978-3-030-32692-0_11 (2019).
33. Ghatwary, N. M., Ye, X. & Zolgharni, M. Esophageal abnormality detection using densenet based faster r-cnn with gabor features. *IEEE Access* **7**, 84374–84385, https://doi.org/10.1109/ACCESS.2019.2925585 (2019).
34. Hicks, S. A. *et al.* Mimir: an automatic reporting and reasoning system for deep learning based analysis in the medical domain. In *Proceedings of the ACM Multimedia Systems Conference (ACM MMSYS)*, 369–374, https://doi.org/10.1145/3204949.3208129 (2018).
35. Hicks, S. *et al.* Dissecting deep neural networks for better medical image classification and classification understanding. In *Proceedings of the IEEE International Symposium on Computer-Based Medical Systems (CBMS)*, 363–368, https://doi.org/10.1109/CBMS.2018.00070 (2018).
36. Hicks, S. A. *et al.* Comprehensible reasoning and automated reporting of medical examinations based on deep learning analysis. In *Proceedings of the ACM Multimedia Systems Conference (ACM MMSYS)*, 490–493, https://doi.org/10.1145/3204949.3208113 (2018).
37. Pogorelov, K. *et al.* Opensea: open search based classification tool. In *Proceedings of the ACM Multimedia Systems Conference (ACM MMSYS)*, 363–368, https://doi.org/10.1145/3204949.3208128 (2018).
38. Thambawita, V. L. *et al.* An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification. *ACM Transactions on Comput. for Healthc.* (2020).
39. Riegler, M. *et al.* Multimedia for medicine: the medico task at mediaeval 2017. In *Proceeding of the MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval)* (2017).
40. Pogorelov, K. *et al.* Medico multimedia task at mediaeval 2018. In *Proceeding of the MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval)* (2018).
41. Hicks, S. *et al.* Acm multimedia biomedia 2019 grand challenge overview. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 2563–2567, https://doi.org/10.1145/3343031.3356058 (2019).
42. Cheplygina, V., de Bruijne, M. & Pluim, J. P. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Med. Image Analysis* **54**, 280–296, https://doi.org/10.1016/j.media.2019.03.009 (2019).
43. Hénaff, O. J., Razavi, A., Doersch, C., Eslami, S. & Oord, A. V. D. Data-efficient image recognition with contrastive predictive coding. arXiv preprint arXiv:*1905.09272* (2019).
44. Misra, I. & van der Maaten, L. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991* (2019).
45. Bui, T. D., Ravi, S. & Ramavajjala, V. Neural graph learning: Training neural networks using graphs. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 64–71, https://doi.org/10.1145/3159652.3159731 (2018).
46. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722* (2019).
47. Amano, Y. *et al.* Interobserver agreement on classifying endoscopic diagnoses of nonerosive esophagitis. *Endoscopy* **38**, 1032–1035, https://doi.org/10.1055/s-2006-944778 (2006).
48. De Lange, T., Larsen, S. & Aabakken, L. Inter-observer agreement in the assessment of endoscopic findings in ulcerative colitis. *BMC gastroenterology* **4**, 9, https://doi.org/10.1186/1471-230X-4-9 (2004).
49. Jha, D. *et al.* Kvasir-seg: A segmented polyp dataset. In *Proceeding of International Conference on Multimedia Modeling (MMM)*, vol. 11962, 451–462, https://doi.org/10.1007/978-3-030-37734-2_37 (2020).
50. Jha, D. *et al.* Resunet++: An advanced architecture for medical image segmentation. In *Proceedings of International Symposium on Multimedia (ISM)*, 225–230, https://doi.org/10.1109/ISM46123.2019.00049 (2019).
51. Borgli, H. *et al.* The HyperKvasir Dataset. *Open Science Framework*, https://doi.org/10.17605/OSF.IO/MH9SJ (2020).
52. Calderwood, A. H. & Jacobson, B. C. Comprehensive validation of the boston bowel preparation scale. *Gastrointest. endoscopy* **72**, 686–692, https://doi.org/10.1016/j.gie.2010.06.068 (2010).
53. Aabakken, L. *et al.* Standardized endoscopic reporting. *J. Gastroenterol. Hepatol.* **29**, 234–240, https://doi.org/10.1111/jgh.12489 (2014).

144

A.1. Paper I - HyperKvasir, a Comprehensive Multi-class Image and Video Dataset for Gastrointestinal Endoscopy

54. Lundell, L. R. *et al.* Endoscopic assessment of oesophagitis: clinical and functional correlates and further validation of the los angeles classification. *Gut* **45**, 172–180, https://doi.org/10.1136/gut.45.2.172 (1999).
55. Sharma, P. *et al.* The development and validation of an endoscopic grading system for barrett's esophagus: The prague c & m criteria. *Gastroenterology* **131**, 1392–1399, https://doi.org/10.1053/j.gastro.2006.08.032 (2006).
56. Lai, E. J., Calderwood, A. H., Doros, G., Fix, O. K. & Jacobson, B. C. The boston bowel preparation scale: a valid and reliable instrument for colonoscopy-oriented research. *Gastrointest. Endosc.* **69**, 620–625, https://doi.org/10.1016/j.gie.2008.05.057 (2009).
57. Schroeder, K. W., Tremaine, W. J. & Ilstrup, D. M. Coated oral 5-aminosalicylic acid therapy for mildly to moderately active ulcerative colitis. *The New Engl. J. Medicine* **317**, 1625–1629, https://doi.org/10.1056/NEJM198712243172603 (1987).
58. Lambert, R. Theparisendoscopicclassificationofsuperficialneoplasticlesions: esophagus,stomach,andcolon: November 30 to december 1, 2002. *Gastrointest Endosc* **58**, S3–S43, https://doi.org/10.1016/S0016-5107(03)02159-X (2003).
59. Forrest, J. H., Finlayson, N. & Shearman, D. Endoscopy in gastrointestinal bleeding. *The Lancet* **304**, 394–397, https://doi.org/10.1016/s0140-6736(74)91770-x (1974).
60. Bray, F. *et al.* Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: a cancer journal for clinicians* **68**, 394–424, https://doi.org/10.3322/caac.21492 (2018).
61. Thambawita, V. *et al.* The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. In *Proceeding of the MediaEval Benchmarking Initiative for Multimedia Evaluation Workshop (MediaEval)* (2018).
62. Bernal, J., Sánchez, J. & Vilarino, F. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognit.* **45**, 3166–3182, https://doi.org/10.1016/j.patcog.2012.03.002 (2012).
63. Ali, S. *et al.* Endoscopy artifact detection (ead 2019) challenge dataset. *arXiv preprint arXiv:1905.03209* (2019).
64. Silva, J., Histace, A., Romain, O., Dray, X. & Granado, B. Toward embedded detection of polyps in wce images for early diagnosisofcolorectalcancer. *Int. J. Comput. Assist. Radiol. Surg.* **9**, 283–293, https://doi.org/10.1007/s11548-013-0926-3 (2014).
65. Koulaouzidis, A. *et al.* Kid project: an internet-based digital video atlas of capsule endoscopy for research purposes. *Endosc. international open* **5**, E477–E483, https://doi.org/10.1055/s-0043-105488 (2017).
66. Bernal, J. & Aymeric, H. Gastrointestinal Image ANAlysis (GIANA) Angiodysplasia D&L challenge. https://endovissub2017-giana.grand-challenge.org/home/, Accessed: 2017-11-20 (2017).
67. Angermann, Q. *et al.* Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis. *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures (CARE CLIP)* **10550**, 29–41, https://doi.org/10.1007/978-3-319-67543-5_3 (2017).
68. Bernal, J. *et al.* Polyp detection benchmark in colonoscopy videos using gtcreator: A novel fully configurable tool for easy and fast annotation of image databases. In Proceedings of Computer Assisted Radiology and Surgery (CARS), https://hal.archives-ouvertes.fr/hal-01846141 (2018).
69. Gastrolab - the gastrointestinal site, http://www.gastrolab.net/index.htm. Accessed: 2019-12-12.
70. Weo clinical endoscopy atlas, http://www.endoatlas.org/index.php. Accessed: 2019-12-12.
71. Gastrointestinal lesions in regular colonoscopy dataset, http://www.depeca.uah.es/colonoscopy_dataset/, Accessed: 2019-12-12.
72. The atlas of gastrointestinal endoscope, http://www.endoatlas.com/atlas_1.html. Accessed: 2019-12-12.
73. El salvador atlas of gastrointestinal video endoscopy, http://www.gastrointestinalatlas.com/index.html. Accessed: 2019-1216.
74. Pogorelov, K. *et al.* Nerthus: A bowel preparation quality video dataset. In *Proceedings of the ACM Multimedia Systems Conference (ACM MMSYS)*, 170–174, https://doi.org/10.1145/3083187.3083216 (2017).

Acknowledgements

Author contributions

S.A.H., V.T., P.H., H.L.H., M.A.R. and T.d.L. conceived the experiment(s), S.A.H., V.T., H.L.H. and M.A.R. conducted the experiment(s), H.B., S.A.H., M.A.R., P.H. and T.d.L. prepared and cleaned the data for publication, and all authors analyzed the results and reviewed the manuscript.

Competing interests

Authors P.H.S., D.J., C.G., M.A.R., P.H. and T.d.L. all own shares in the Augere Medical AS company developing AI solutions for colonoscopies. The Augere video annotation system was used to label the videos. There is no commercial interest from Augere regarding this publication and dataset. Otherwise, the authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to P.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A.2   Paper II - Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros

Authors: Henrik Svoren, **Vajira Thambawita**, Pål Halvorsen, Petter Jakobsen, Enrique Garcia-Ceja, Farzan Majeed Noori, Hugo L. Hammer, Mathias Lux, Michael Alexander Riegler, Steven Alexander Hicks

Abstract: Games are often defined as engines of experience, and they are heavily relying on emotions, they arouse in players. In this paper, we present a dataset called Toadstool as well as a reproducible methodology to extend on the dataset. The dataset consists of video, sensor, and demographic data collected from ten participants playing Super Mario Bros, an iconic and famous video game. The sensor data is collected through an Empatica E4 wristband, which provides high-quality measurements and is graded as a medical device. In addition to the dataset and the methodology for data collection, we present a set of baseline experiments which show that we can use video game frames together with the facial expressions to predict the blood volume pulse of the person playing Super Mario Bros. With the dataset and the collection methodology we aim to contribute to research on emotionally aware machine learning algorithms, focusing on reinforcement learning and multimodal data fusion. We believe that the presented dataset can be interesting for a manifold of researchers to explore exciting new interdisciplinary questions.

Published: The ACM Multimedia Systems Conference (MMSys) - 2020

Candidate contributions: Vajira contributed to the conception and designing of the theoretical models. He contributed to collecting data as a participant also. Vajira contributed to publishing data in osf.io and organizing it. He contributed to drafting the paper and revising it.

Thesis objectives: Sub-objective I, Sub-objective II

Toadstool: A Dataset for Training Emotional Intelligent Machines Playing Super Mario Bros

Henrik Svoren[*]
henrik.svo@gmail.com
University of Oslo
Oslo, Norway

Vajira Thambawita[‡]
vajira@simula.no
SimulaMet
Oslo, Norway

Pål Halvorsen[‡]
paalh@simula.no
SimulaMet
Oslo, Norway

Petter Jakobsen[†]
peja@helse-bergen.no
NORMENT, Haukeland University
Hospital
Bergen, Norway

Enrique Garcia-Ceja
enrique.garcia-ceja@sintef.no
SINTEF Digital
Oslo, Norway

Farzan Majeed Noori
farzanmn@ifi.uio.no
University of Oslo
Oslo, Norway

Hugo L. Hammer[*]
hugoh@oslomet.no
OsloMet
Oslo, Norway

Mathias Lux
mlux@itec.aau.at
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria

Michael Alexander Riegler[§]
michael@simula.no
SimulaMet
Oslo, Norway

Steven Alexander Hicks[‡][§]
steven@simula.no
SimulaMet
Oslo, Norway

ABSTRACT

Games are often defined as engines of experience, and they are heavily relying on emotions, they arouse in players. In this paper, we present a dataset called *Toadstool* as well as a reproducible methodology to extend on the dataset. The dataset consists of video, sensor, and demographic data collected from ten participants playing *Super Mario Bros*, an iconic and famous video game. The sensor data is collected through an Empatica E4 wristband, which provides high-quality measurements and is graded as a medical device. In addition to the dataset and the methodology for data collection, we present a set of baseline experiments which show that we can use video game frames together with the facial expressions to predict the blood volume pulse of the person playing Super Mario Bros. With the dataset and the collection methodology we aim to contribute to research on emotionally aware machine learning algorithms, focusing on reinforcement learning and multimodal data fusion. We believe that the presented dataset can be interesting for a manifold of researchers to explore exciting new interdisciplinary questions.

CCS CONCEPTS

• **Applied computing** → **Health informatics**; • **Computing methodologies** → *Visual inspection*; *Neural networks*; *Classification and regression trees*.

KEYWORDS

Multimedia Datasets, Neural Networks, Emotional Machines, Machine Learning

---

[*]Also affiliated with SimulaMet, Norway
[†]Also affiliated with University of Bergen, Norway
[‡]Also affiliated with Oslo Metropolitan University, Norway
[§]These authors share the senior position of the article.

---

1 INTRODUCTION

"Stop Dave. Stop Dave. I am afraid. I am afraid Dave." This iconic quote from Stanley Kubrick's *2001: A Space Odyssey* is taken from a scene where the sentient computer system HAL 9000 is pleading for life as the human operator is about to shut it down. The movie was released in 1969, and looking at the state of artificial intelligence

Appendix A. Published Articles

Figure 1: Frames are taken from each of the 32 levels contained within *Super Mario Bros*. Note that each image is taken from the very first frame of each level. Levels in *Super Mario Bros*. are organized in groups of four and called worlds, so the first level is world 1-1, the second level is world 1-2, the fifth level is world 2-1, etc.

(AI) today, we can make two observations. First, people in the 60s and 70s were very optimistic about the future capabilities of AI. Second, we are far away from anything near the emotional intelligence that HAL 9000 expresses throughout the movie. For the most part, current AI systems are focused on performing well on specific tasks like classification, object detection or regression, while a machine that can express general intelligence is still far off. This is not negative in and of itself [7], but it is quite different from what people in the past imagined AI would be in the future, and what we might imagine today.

Using machine learning to interpret or detect human emotions is a growing field of research. This is commonly done using different types of media, such as images [2], sensor data [11], text [23], or some combination of the three [4, 14]. Recent works in this field have also moved to look at how human emotion data may affect the training and performance of deep learning algorithms. McDuff et al. [17] explore how human emotional response may affect the performance of a self-driving agent trained in a simulated environment. They showed that adding human-like signals, such as the blood volume pulse (BVP), helped improve the driving performance of the algorithm. The idea of supplementing today's machines with emotional or physiological signals is supported by the large amount of literature that shows that pure rational decision making is often not optimal in humans [3, 9, 18, 22]. Prior research shows that emotional content can help guide the decision-making process as well as make it more efficient [16]. Some early work also tried to use this for artificial agents [10]. Such findings suggest the possibility that similar benefits might be had by artificial agents, especially when engaged in human-like tasks or behavior.

Inspired by the work done by McDuff et al. [17], we look at other areas where the same principles may be applied, which in this case, is playing the well-known classical video game *Super Mario Bros*. While this game is not representative for all video games that are available right now, it is commonly accepted as a well-known, good example for a video game and can be considered representative

for the jump and run and the arcade game genres. To perform experiments in that direction, we first need a dataset that contains both the frames from *Super Mario Bros*. and the sensory output of the player. As no such dataset exists, we collected gameplay data, sensor data, and facial expression data from ten different participants. Furthermore, we also made the dataset and all sources to re-produce the games played publicly available. We think this dataset is of great interest to many research communities as it consists of multiple modalities and is applied to a unique use case. The contributions of this paper are three-fold:

(1) We present a publicly available, multimodal dataset which focuses on the human component of intelligent machines along with a reproducible methodology to extend the dataset with additional data collection.

(2) We present a set of baseline experiments that aim to show how the dataset can be used to predict specific sensor values using a combination of data from the video game and facial expressions.

(3) We outline future applications and interesting research questions using the dataset.

To the best of our knowledge, this is the first open dataset that provides the (i) video frames of a person's facial expressions, (ii) the sensory output of the person playing a game, and (iii) data from the video game synchronized with the facial expressions and sensor data. The dataset opens up for a wide range of new and interesting analyses, and a proper and fair comparison between different methods, both from a psychological and a multimedia perspective. In the following, the process of collecting the data, as well as the resulting data, are described. Moreover, a baseline evaluation is presented, including suggestions for future research directions using the dataset.

ID	Age	Sex	Dominant hand	Hours per week	Years active	Prior experience	Game score
0	26	Male	Right	4-8	22	Lots	17, 100
1	48	Male	Left	0-1	1	Little	3, 000
2	28	Male	Right	0-1	0	None	300
3	32	Male	Right	4-8	4	Some	13, 300
4	32	Female	Right	0-1	5	Some	6, 400
5	30	Female	Right	0-1	5	Little	2, 700
6	35	Male	Left	1-4	30	Lots	14, 300
7	34	Female	Right	1-4	14	Some	3, 800
8	31	Female	Right	0-1	2	Little	200
9	27	Female	Right	0-1	5	Little	10, 600

Table 1: This table shows an overview of all participants included in the dataset.

2 DATA COLLECTION

The dataset was collected at our research laboratory located in Oslo, Norway. Participants were selected based on a set of criteria, mostly focused on their prior gaming experience. We wanted to collect data from people with a wide range of different game experience backgrounds. This includes those who have barely touched a video game to those who have been playing since childhood. Furthermore, we aimed to collect data from a balanced set of genders, meaning an even split of male and female participants. Each participant was asked to fill out a short questionnaire about their previous video game experience in addition to some information about themselves. An overview of the answers can be seen in Table 1. In total, ten participants were selected for the study, where each participant provided a written form of consent, allowing for their video, gameplay data, and sensor data to be shared openly for research and teaching purposes under the license Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)[1]. The dataset can be accessed via (https://datasets.simula.no/toadstool) or (https://osf.io/qrkcf/).

As for collecting the gameplay data, we developed a protocol that describes what data should be collected and how. This protocol went through multiple iterations as we performed a preliminary test run before applying it to all participants in the study. From this initial test run, we learned that, in some cases, the conductivity between the participant and the wristband (Empatica E4) did not gather data in line with what we expected. Some anomalies included little to no detected activity and substantial value differences between participants. Furthermore, we noticed that the activity would vary a lot between the start and end of a gameplay session. The primary cause of this was mostly due to the dry conditions in which the data was collected. For the wristband to accurately pick up a person's sensor data, the electrodes need some sweat to act as a conductor between the skin and wristband. Based on these observations, we changed the protocol to include a short warm-up session before playing the video game and a 15-minute period where the participant would sit still to develop a baseline. The warm-up consisted of walking up and down a flight of stairs spanning six floors two times. This exercise was selected based on tests with some people in the laboratory. The final protocol is shared with the dataset.

Before playing, the participants were told that their performance in the game would be measured based on how many stages were cleared in the time given and on the number of player avatar deaths. Furthermore, we informed participants that their performance would be measured against other participants and that there would be a prize for the highest achiever. The motivation behind making the game more competitive was to make the players want to perform well, and feel like there was some consequence if they either died (in the game) or did not beat levels fast enough. The number of points earned by each player was kept secret from all participants to avoid them giving up or relaxing due to other players too high or too low score. Scores were calculated based on two primary factors; the number of deaths and levels cleared. Starting a level, the player starts with a base score of 1000. For every death, the score is reduced by 100 points down to a minimum of 200. If the player manages to beat the level, he/she is awarded between 200 and 1000. If the player runs out of time, he/she is awarded 0 points and is moved to the next stage. The final score of each participant is included in the dataset and can be seen in Table 1.

After the participants had established a sufficient baseline, they started the primary game session where the video of the participants, video game frames, and sensor data was collected. Each game session lasted for approximately 35 minutes. The game was played directly in the gym-super-mario-bros environment [13], which is a gym [1] based environment for *Super Mario Bros.* For reproducibilities sake, the repository for the gym environment has been added to the official GitHub repository of Toadstool[2]. There are four different graphic environments offered by gym-super-mario-bros, which include the standard graphics, as well as three different downsampled versions (downsample, pixel, and rectangle). The data in our dataset is collected in the standard environment, but sessions may be replayed in any environment if needed.

The gym environment version of the game still differs somewhat from the original gaming experience found in *Super Mario Bros.* by Nintendo (a consumer electronics company from Japan). Firstly, all game-freezing animations and cutscenes are removed from the game. This includes transitions between levels and traveling through pipes. Second, there is no music or other sound effects. Third, there are no limits on game lives, and power-ups do not carry over to new stages. Last, the order of the levels has been changed compared to the original game. There was one additional rule we told participants before playing. In *Super Mario Bros.*, some pipes can warp the player to a new stage that is closer to the final stage of the game. To keep the levels played consistent between players,

---

[1] https://creativecommons.org/licenses/by-nc/4.0/

[2] https://github.com/simula/toadstool

Appendix A. Published Articles

we asked participants to refrain from using any of the available warp pipes (one located in world 1-2 and two located in world 4-2). Overall, it took approximately one hour to collect data from a single participant.

3 DATASET DETAILS

For each participant, we have included a video of them playing the game (camera facing the face), the controller input performed on each frame of the game, and the sensor data collected from an Empatica E4 wristband [6]. The camera used to collect the facial expression data was a 1.3-MP webcam attached to a Samsung Series 9 Notebook NP900X4C. The webcam captured video at 30 frames per second with a resolution of 640×480. The controller used to play the game was a wired USB controller from retro-bit, which is modeled after the original controller for the Nintendo Entertainment System. Note that the video game frames are not included in the dataset, but can be extracted by using the provided video game actions files included with each participant. This can be done by using a script that is included in the dataset. The reason for not including the video game frames was mostly due to the exponential increase in storage size. Another possible benefit of this approach is the ability to replay the game session in any of the several environments offered by the gym-super-mario-bros framework to produce different representations of game frames. The first frame for each of the 32 levels of *Super Mario Bros.* can be seen in Figure 1. The dataset contains the following files:

- **participants** is the directory that contains the information of each participant. This includes the video of them playing, the controller input of each game frame, and the Empatica E4 wristband sensor data.
- **scripts** is a directory that holds a set of Python scripts meant to aid the user in getting an easy start to using the dataset. The files include a script for replaying gameplay using the provided controller inputs, a script for matching the gameplay session to the facial expression video, and a script for matching the raw signal outputs to the gameplay session.
- **protocol.pdf** is the protocol used to collect the video game session data.
- **questionnaire.pdf** is the questionnaire that was filled out by each participant before starting the game session.
- **questionnaire_answers.csv** is a summary of all the answers to the questionnaire.
- **consent.pdf** is the consent form that was signed by each participant.
- **README.txt** is a short information file which describes the contents of the dataset.
- **LICENSE** is the file that signifies which license in which the dataset is distributed under.

Contained within the *participants* directory is a separate directory per participant included in the dataset. Each directory has a name corresponding to the participant's ID, i.e., *participant_<ID>*, where *<ID>* is replaced with the ID of the participant. For each participant, we have stored the participant's sensor data collected from the Empatica E4 wristband, a JSON file containing information about the participants game session, the video recording of the participant playing the game stored in ".avi" format, and another JSON

file which contains information about the video. The JSON file that holds the game session data, called *participant_<ID>_session.json*, contains the actions performed during the game, the start and end times of the game session, and the achieved gameplay score. As for the sensor data, the wristband uses four separate sensors to collect different sensory outputs from the wearer, such as the electrodermal activity (EDA), interbeat intervals (IBI), heart rate (HR), and blood volume pulse (BVP). The four sensors of the wristband is a photoplethysmography sensor, an electrodermal activity sensor, 3-axis accelerometer, and an optical thermometer. Of the four sensors, the thermometer is the only one not graded for clinical use. All data collected by the wristband are stored in CSV files that can be downloaded from the wristband. For the dataset, these CSV files have been matched to the game session. We have also opted to include the raw source files as they were collected from the wristband. The CSV files and a short description of the contents are further explained below.

- **ACC.csv** contains the data collected from the 3-axis accelerometer sensor in the range [-2g, 2g] sampled at 32 Hz. The accelerometer measures the movement of the wearer.
- **EDA.csv** holds the data collected by the EDA sensor sampled at 4 Hz. EDA measures the electrical conductivity of the skin and measurements have been proven to be correlated with emotions since the late 1800s [5]. EDA is also sometimes called psychogalvanic reflex or skin conductance.
- **BVP.csv** contains the data collected from the photoplethysmograph sensor, which measures the BVP, and is sampled at 64 Hz.
- **IBI.csv** stores the interbeat intervals (IBI). The IBI measures the time interval between individual heartbeats and can be used to estimate the instantaneous heart rate as well as heart rate variability. The wristband calculates the values contained within this file based on the BVP signals.
- **HR.csv** contains the average heart rate values, computed in spans of 10 seconds. The heart rate measures the number of times a person's heartbeats per minute. Similar to the IBI, these values are calculated based on the BVP signal.
- **TEMP.csv** holds the information collected by the thermometer, which is the temperature of the person playing the game expressed in degrees Celsius (℃) sampled at 4Hz.
- **info.txt** gives a brief description of all variables collected by the wristband.

In addition to the data collected throughout the study, we also include a set of scripts that aim to make the dataset more accessible. First of all, as previously mentioned, video game frames are not included in the dataset. However, we include the necessary information to extract the frames by using the provided video game inputs to "replay" the game session and collect the video frames directly.

4 PRELIMINARY EXPERIMENTS

We performed a set of preliminary experiments to showcase how the presented dataset can be used to train machine learning algorithms and perform simple predictive modeling. In Section 5, we mention that a possible use case for the dataset is to predict the sensor value of the wristband using the video game frames or facial

ID	CNN		ZeroR	
	MAE	RMSE	MAE	RMSE
0	0.076	0.100	0.075	0.099
1	0.104	0.132	0.103	0.131
2	0.071	0.104	0.075	0.100
3	0.050	0.070	0.050	0.070
4	0.078	0.103	0.069	0.094
5	0.091	0.129	0.094	0.121
6	0.091	0.119	0.090	0.116
7	0.110	0.142	0.109	0.139
8	0.061	0.096	0.060	0.090
9	0.126	0.157	0.109	0.134
All	0.105	0.126	0.090	0.117

Table 2: This table shows the results of all experiments of trying to predict the BVP amplitude using video game frames and facial expressions alone. Note that all experiments were trained over three-fold cross-validation.

expressions as input. In the following experiments, we trained a deep convolutional neural network to predict the BVP amplitudes utilizing a combination of the face data and game frame data. This is similar to what McDuff et al. [17] did when modeling the *emotional* input to their self-driving reinforcement agent.

Before the training step, we needed to prepare the input data, i.e., the video game recording and facial expression video. As the gameplay frames were recorded at 60 frames per second, while the facial expression video was recorded at 30 frames per second, we down-sampled the video game frames by skipping every other frame. After that, the BVP amplitudes had to be extracted from the raw BVP signals and matched to the input frames. The reason for not predicting the raw BVP values is due to the cyclic nature of a beating heart. The BVP has the properties of a sine-like wave that is composed of valleys and peaks which appear in tandem with every heartbeat. We are not interested in the exact value on the signal curve, but the highest point of a cardiac cycle, also known as the systolic peak. This peak-value gives us some information about the emotional state of the human player [20]. To extract the BVP amplitudes, we detect a systolic peak in the given BVP signal and measure its height from the baseline. This peak-value is then repeated until the next detected peak, and-so-forth. The result is a square-like signal in comparison to the sine-like wave that is the BVP. The BVP values were then matched with the input data by taking the average BVP amplitude values over one second (64 measurements in total). The extracted peaks used for the experiments are shared on GitHub together with the code used to produce the baseline experiments[3].

We trained one convolutional neural network (CNN) per participant in addition to one trained on all participants mixed. The purpose of training a model on a single participant is because we generally want to model the emotional response of a single person, not necessarily everyone at once. The model was based on the TensorFlow implementation of ResNet50 [8], where we input two video game frames (first and the last frame of one second) and the

___
[3]https://github.com/simula/toadstool

facial expression from the last frame of the second corresponding to the video game frames. These three images were grayscaled and stacked channel-wise before being processed by the model. In addition to the CNN-based model, we also calculated the error when using the mean of the response variable in the training for prediction (also sometimes called ZeroR), which should give us some indication about how well our model performs. Since the response is a continuous measurement, we used the metrics mean absolute error (MAE) and root mean squared error (RMSE). Furthermore, we used three-fold cross-validation to not bias the results towards a pre-defined split of the data.

The results of predicting the BVP amplitudes using video game frames and facial expressions can be seen in Table 2. We observe that the trained model sometimes outperforms the baseline and sometimes not, which is an indicator of a challenging task, but not impossible. There is still much room for improvement, where methods recurrent neural networks may be used to increase performance, but this is out of scope for this paper. Furthermore, we want to point out that for classification tasks, such as classifying emotional states based on sensor and video data, one should use standard classification metrics such as precision, recall/sensitivity, and F1-score to determine the quality of the trained model. Optimally, one should also report the true positives, true negatives, false positives, and false negatives so that readers themselves can calculate the metrics manually. The code used to run all experiments is available online in the datasets official GitHub repository (https://github.com/simula/toadstool).

5 POSSIBLE APPLICATIONS OF THE DATASET

We expect that this dataset can be used for many different use cases and scenarios. First of all, we imagine it can be used to detect relationships between the player sentiment and the current gameplay state. This could solely be based on the gameplay frames and collected sensor data, or could also be combined with the player's facial expression for further analysis. The uncovered relationships could be interesting when studying how players get invested in video games, and what typical scenarios contribute to a strong reaction from players. A more specific example could be predicting a person's facial expression based on a given game state or degree of progress in a game or level. This could also be expanded to the sensor data, as one could predict the sensory output of the wristband using the game state and facial expressions as input, like photoplethysmography [21] does with the heart rate, but connected to the current game state the player is in. These two problems could be modeled as either a regression or classification problem, depending on the application.

From a game design perspective, correlations between game progress, game state, the input of players, and the emotions and sentiment of the players could enable a whole new approach to experience-based game design. With the possibility to use this as a method for playtesting, game designers can evaluate when and how their crafted experience is (or is not) invoked in the players. Moreover, games can be built around the concept of self-adapting challenge and difficulty by counter-acting unnecessary frustration, boredom, or annoyance by adapting the game to the player's emotion and sentiment. Last but not least, the correlation between the

Appendix A. Published Articles

game, emotions, sentiment, and game engagement, especially flow (the state of being fully immersed in an activity while enjoying it), can be investigated [15, 24]. This would lead to a better understanding of what makes games enjoyable and would impact fields like game and media studies, psychology, game engineering, game design, and serious games / educational games.

Another area where this dataset could be used is in the training of emotionally intelligent machines using reinforcement learning. One way to do this would be by training an algorithm to reproduce the physiological signals based on the game-state. The reproduced signals can then be incorporated into the reward function of a reinforcement learning agent. Since physiological signals like BVP and EDA are reliable indicators of emotional states in humans [12, 19], such an approach could be used to mimic human emotional responses. These kinds of emotionally intelligent machines might aid the pure logic of reinforcement learning models and produce improved learning, as well as reveal new insights into how both machines and humans learn. They might also be a step towards machines with even more complex emotional intelligence, like the ability to recognize, express, and respond to human emotions. This is an active research area leading to better personal assistants, more believable automated communication, as well as more enjoyable and believable video game AI.

Some more concrete examples of possible research questions or experiments are:

- How are the sensor measurements related to the facial expressions?
- Can sentiment analysis of the facial expression be connected to measurements in the sensors?
- How can different data sources be combined efficiently (sensor data with videos, etc.).
- Can game actions of the human players be used as a baseline for human performance?
- Can the additional data collected from people be used to train reinforcement learning algorithms?
- Would a model trained on input from non-experienced players behave differently from a model trained on experienced players?
- Can immersion, enjoyment, and flow automatically be inferred from the gathered data?

As one can see from the discussion above, the dataset holds a lot of interesting research potential to follow up, and we hope that other researchers get inspired to work on the dataset.

6 CONCLUSION

In this paper, we present *Toadstool*, a new dataset consisting of people playing *Super Mario Bros.* and physiological signals corresponding to their emotional reaction to playing said game. While the dataset provides a good starting point for applied research in affective computing in games, emotional AI agents in games, playtesting, and game analytics, we strongly believe the *Toadstool* dataset will foster research in many ways and, therefore, we have detailed follow up research questions in several fields, including affective computing, psychology, and game and media studies. With the detailed description of the dataset, the in-depth discussion of the method included in the dataset, and the wide availability of the

game instance used for the experiment and the medical sensors employed, we ensured reproducibility and extension of the dataset. We hope that our work inspires and encourages interdisciplinary and multidisciplinary research to (i) examine how the human element in human-computer interaction can be employed to improve existing machine learning methods by introducing emotional aspects to an otherwise cold and unfeeling machine and (ii) show how interactive entertainment systems, and especially video games, utilize sensory input to provide a personalized and tailored user experience.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:arXiv:1606.01540
[2] Prudhvi Raj Dachapally. 2017. Facial emotion detection using convolutional neural networks and representational autoencoder units. *arXiv preprint arXiv:1706.01509* (2017).
[3] Benedetto De Martino, Dharshan Kumaran, Ben Seymour, and Raymond J Dolan. 2006. Frames, biases, and rational decision-making in the human brain. *Science* 313, 5787 (2006), 684–687.
[4] Liyanage C De Silva, Tsutomu Miyasato, and Ryohei Nakatsu. 1997. Facial emotion recognition using multi-modal information. In *ICICS 1997*, Vol. 1. IEEE, 397–401.
[5] Ch Fere. 1888. Note sur des modifications de la resistance electrique sous l'influence des excitations sensorielles et des motions. *Compt Rend Soc de Biol* 8 (1888), 217–219.
[6] M. Garbarino, M. Lai, D. Bender, R.W. Picard, and S. Tognetti. 2014. Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. In *ICWMCHM 2014*. 39–42.
[7] Uri Hasson, Samuel A Nastase, and Ariel Goldstein. 2020. Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks. *Neuron* 105, 3 (2020), 416–434.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE CVPR*. 770–778.
[9] Michael Hechter and Satoshi Kanazawa. 1997. Sociological rational choice theory. *Annual review of sociology* 23, 1 (1997), 191–214.
[10] Hong Jiang and Jose M Vidal. 2006. From rational to emotional agents. In *AAAI CMASS 2006*.
[11] Eiman Kanjo, Eman M.G. Younis, and Chee Siang Ang. 2019. Deep learning analysis of mobile physiological, environmental and location sensor data for emotion detection. *Information Fusion* 49 (2019), 46 – 56.
[12] Krisztian Kasos, Szabolcs Zimonyi, Eniko Kasos, Avraham Lifshitz, Katalin Varga, and Anna Szekely. 2018. Does the electrodermal system "take sides" when it comes to emotions? *APBF Journal* 43, 3 (2018), 203–210.
[13] Christian Kauten. 2018. Super Mario Bros for OpenAI Gym. https://github.com/Kautenja/gym-super-mario-bros.
[14] Anil Kumar K.M., Kiran B.R., Shreyas B.R., and Sylvester J. Victor. 2015. A Multimodal Approach To Detect User's Emotion. *Procedia Computer Science* 70 (2015), 296 – 303.
[15] Derek A Laffan, John Greaney, Hannah Barton, and Linda K Kaye. 2016. The relationships between the structural video game characteristics, video game engagement and happiness among individuals who play video games. *Computers in Human Behavior* 65 (2016), 544–549.
[16] Jennifer S. Lerner, Ye Li, Piercarlo Valdesolo, and Karim S. Kassam. 2015. Emotion and Decision Making. *Annual Review of Psychology* 66, 1 (2015), 799–823.
[17] Daniel McDuff and Ashish Kapoor. 2019. Visceral Machines: Risk-Aversion in Reinforcement Learning with Intrinsic Physiological Rewards. In *ICLR 2019*.
[18] David Palumbo-Liu. 2005. Rational and Irrational Choices: Form, Affect, and Ethics. *Minor Transnationalism* (2005), 41–72.
[19] Rosalind W. Picard, Szymon Fedor, and Yadid Ayzenberg. 2016. Multiple Arousal Theory and Daily-Life Electrodermal Activity Asymmetry. *Emotion Review* 8, 1 (2016), 62–75.
[20] R. W. Picard, E. Vyzas, and J. Healey. 2001. Toward machine emotional intelligence: analysis of affective physiological state. *IEEE Transactions on PAMI* 23, 10 (Oct 2001), 1175–1191.
[21] Philipp V Rouast, Marc TP Adam, Raymond Chiong, David Cornforth, and Ewa Lux. 2018. Remote heart rate measurement using low-cost RGB face video: a technical literature review. *Frontiers of Computer Science* 12, 5 (2018), 858–872.
[22] John Scott. 2000. Rational choice theory. *Understanding contemporary society: Theories of the present* 129 (2000), 671–85.
[23] Shiv Naresh Shivhare and Saritha Khethawat. 2012. Emotion detection from text. *arXiv preprint arXiv:1205.4944* (2012).
[24] Penelope Sweetser and Peta Wyeth. 2005. GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3, 3 (2005), 3–3.

A.3 Paper III - PMData: A Sports Logging Dataset

Authors: Vajira Thambawita, Steven Alexander Hicks, Hanna Borgli, Håkon Kvale Stensland, Debesh Jha, Martin Kristoffer Svensen, Svein-Arne Pettersen, Dag Johansen, Håvard Dagenborg Johansen, Susann Dahl Pettersen, Simon Nordvang, Sigurd Pedersen, Anders Gjerdrum, Tor-Morten Grønli, Per Morten Fredriksen, Ragnhild Eg, Kjeld Hansen, Siri Fagernes, Christine Claudi, Andreas Biørn-Hansen, Duc Tien Dang Nguyen, Tomas Kupka, Hugo Lewi Hammer, Ramesh Jain, Michael Alexander Riegler, Pål Halvorsen

Abstract: In this paper, we present PMData: a dataset that combines traditional lifelogging data with sports-activity data. Our dataset enables the development of novel data analysis and machine-learning applications where, for instance, additional sports data is used to predict and analyze everyday developments, like a person's weight and sleep patterns; and applications where traditional lifelog data is used in a sports context to predict athletes' performance. PMData combines input from Fitbit Versa 2 smartwatch wristbands, the PMSys sports logging smartphone application, and Google forms. Logging data has been collected from 16 persons for five months. Our initial experiments show that novel analyses are possible, but there is still room for improvement.

Candidate contributions: Vajira contributed to the analysis and interpretation of data. He contributed to drafting the article, hosting the data in an open access data hosting location (osf.io), and revising the manuscript. He presented the paper at MMSys 2020.

Thesis objectives: Sub-objective I, Sub-objective II

PMData: **A Sports Logging Dataset**

Vajira Thambawita*
Steven Alexander Hicks*
Hanna Borgli†
Håkon Kvale Stensland†
Debesh Jha‡
Martin Kristoffer Svensen†
SimulaMet
Norway

Svein-Arne Pettersen
Dag Johansen
Håvard Dagenborg Johansen
Susann Dahl Pettersen
Simon Nordvang
Sigurd Pedersen
Anders Gjerdrum
UiT The Arctic University of Norway
Norway

Tor-Morten Grønli
Per Morten Fredriksen
Ragnhild Eg
Kjeld Hansen
Siri Fagernes
Christine Claudi
Andreas Biørn-Hansen
Kristiania University College
Norway

Duc Tien Dang Nguyen
University of Bergen
Norway

Tomas Kupka
Forzasys AS
Norway

Hugo Lewi Hammer§
OsloMet
Norway

Ramesh Jain
University of California, Irvine
US

Michael Alexander Riegler
SimulaMet
Norway

Pål Halvorsen*
SimulaMet
Norway

ABSTRACT

In this paper, we present PMData: a dataset that combines traditional lifelogging data with sports-activity data. Our dataset enables the development of novel data analysis and machine-learning applications where, for instance, additional sports data is used to predict and analyze everyday developments, like a person's weight and sleep patterns; and applications where traditional lifelog data is used in a sports context to predict athletes' performance. PMData combines input from Fitbit Versa 2 smartwatch wristbands, the PMSys sports logging smartphone application, and Google forms. Logging data has been collected from 16 persons for five months. Our initial experiments show that novel analyses are possible, but there is still room for improvement.

CCS CONCEPTS

• **Applied computing** → *Health informatics*; • **Computing methodologies** → *Machine learning*.

KEYWORDS

Multimedia Dataset, Neural Networks, Machine Learning, Sports Logging, Sensor Data, Questionnaires, Food Pictures

---

*Also affiliated with Oslo Metropolitan University, Norway
†Also affiliated with University of Oslo, Norway
‡Also affiliated with UiT The Arctic University of Norway
§Also affiliated with SimulaMet, Norway

1 INTRODUCTION

In one way or another, many people are recording parts of their lives digitally. This could, for example, be through sensors in a smartwatch, GPS location tracking in smartphones, pictures from highly portable cameras, or through activities on various online social media services. It is not uncommon to see people posting pictures of their food on platforms such as Instagram or boasting about their workouts on Facebook as the events unfold.

The activity of recording one's life digitally, through various input sources, is often referred to as *lifelogging* [11], and a person who engages consciously in such activities is referred to as a *lifelogger*. Recording and analyzing lifelog data is a great opportunity for studying an individual's life experience. It can help monitor a person's activity to improve health [17], help recover memories of past events [19], or analyze social behaviour [3, 15]. From a multimedia perspective, lifelogs are sources of wast rich data for interesting research. For instance, Chokr and Elbassuoni [1] describe a machine-learning approach for predicting the number of calories from pictures of food, and De Choudhury et al. [2] describe how interaction on social media influence our mental health.

Although lifelogs might contain data highly valuable for research, they are often not available to the researchers. A lifelog is typically

Thambawita et al.

not stored centrally in one single service that can be tapped into, but rather exists as the union of data stored in a large number of online and offline data silos [13]. Still, some datasets exist, and existing lifelogging datasets [12] usually contain a person's daily life activities automatically captured and recorded using smartphone applications, wearable devices, and other sensors. One example is the NTCIR Lifelog test collection [10] consisting of lifelogging datasets for the NTCIR-12/13/14 lifelog tasks, which was first released at the NTCIR-12 conference [9]. The images in this dataset are captured by wearable cameras carried by two different lifeloggers. Some work has been done with similar datasets, for example, retrieving moment of interest [5, 15]. However, a key challenge in lifelogging research is the poor availability of test collections [4]. Hence, there is a need for more available lifelog datasets.

Capturing daily life events is also something many sports professionals do. Athletes have kept written training diaries for a long time, using both pen-and-paper, and more recently using digital logging systems. Now, the use of wearables to measure activity and its intensity in both top sport and among the regular physically active population help to improve performance, recovery, and other aspects of health [6]. A challenge is to make sense of the data, and often, the captured data is limited to self-reports since activity logs from smartwatches and phones are hard to understand. Thus, there are still steps needed for integration of data [8] and to find standardized ways to analyze, evaluate, and present data [7]. Another problem in the area of sport is that professional athletes do not control the captured data by themselves, and they need the assistance of coaches, physicians, or support staff [13]. This process adds the burden of informed consent, authorization, and privacy. Moreover, a trainer or team doctor does not have time to look at the myriads of sensor data from the athletes to possibly find something that could be used to improve training. Using PMData, we have launched a competition task in ImageCLEF/LifeCLEF[1], where the goal is to predict the participants' weight and run performance at the end of the data collection period.

To aid these efforts, automatic methods to analyze sensor data and the quantification of self-reports will play an important role in retrieving the information that sports athletes may need. To be able to perform these analyses with the increasing volume of data coming from different devices, new methods and tools are needed. PMData is made available in an effort towards enabling development such support systems. We provide a starting point by combining the idea of lifelogging data collection with sports activity logging. Multiple sport-specific analyses can be performed on such data as predicting sports performance, weight loss, or gain, but there is a lack of available datasets. We have therefore logged objective parameters like heart rate, sleep, calorie consumption, movement distance, activity sessions, weight, and subjective parameters of wellness, training load, injuries, food, and drink intake. We have used the Fitbit Versa 2 smartwatch[2], the PMSys sports logging app,[3] and Google forms for the data collection. For now, the dataset, named PMData, contains logging data for three months from 16

persons. To the best of our knowledge, PMData is the first available dataset to combine both subjective and objective parameters combining both daily life and sports activities.

In the following, we describe the procedure for collecting data and describe the dataset in detail. Furthermore, we present a preliminary experiment using machine learning to predict the possibility of a person gaining, losing, or keeping the current weight from logging. We also provide possible research questions and applications of the dataset.

2 DATA COLLECTION

The goal of PMData has been to gather lifelog data related to the activities of our participants, but without being too invasive. We planned to collect data from the end of November 2019 to the end of March 2020. We log data about the participant's daily activities, similar to a sport lifelog, and encourage them to exercise at least twice a week. We did not set any restrictions or requirements on the type or duration of the exercise participants can engage in.

2.1 Fitbit Versa 2: Objective Biometrics and Activity Data

To log objective biometrics and activity data, we used the Fitbit Versa 2 fitness smartwatch (see Figure 1). Each participant was encouraged to wear the watch as much as possible, also when sleeping. All settings were set to default, i.e., sleep tracking in normal mode and auto-exercise recognition on for all activities longer than 15 minutes. When training, participants were told to log in using the exercise menu option in the watch (e.g., run or treadmill).

Figure 1: Fitbit Versa 2

2.2 PMSys: Subjective Wellness, Training Load, and Injuries

Subjective assessments of each participant's wellness, training load, and injuries have been logged using the PM Reporter Pro smartphone application[4] where Figure 2 shows an example of a reporting sequence. PM Reporter is part of the PMSys online sports logging system that enables athletes to monitor individual training load, daily subjective wellness parameters, and injuries [20]. Wellness is reported typically once a day through a sequence of questionnaires. Training load or Session Rating of Perceived Exertion (sRPE) is a metric calculated from the product of the session length and the reported Rating of Perceived Exertion (RPE). The training load is reported after every training session. Finally, the injuries questionnaire is recommended completed once a week, regardless of having an injury or not, where the participants press on a body part to indicate a minor or major injury or pain. To increase the reporting rate, PMSys sends scheduled push messages directly to the participants' smartphones, reminding them to report.

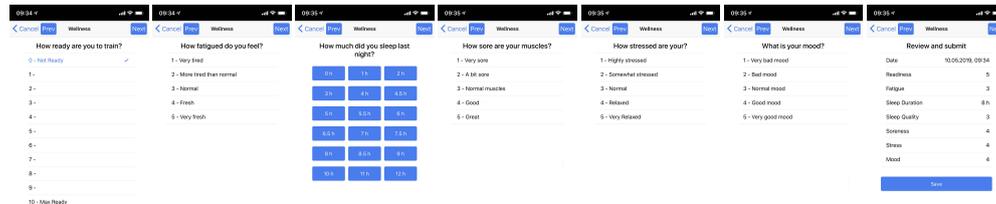[1]https://www.imageclef.org/2020/lifelog
[2]https://www.fitbit.com/no/versa
[3]https://forzasys.com/pmsys.html

[4]https://bitbucket.org/corporesano/pm-reporter

Figure 2: Entering wellness data into PMSys

2.3 Google Forms: Demographics, Food, Drinking, and Weight

A Google Form questionnaire was used to collect information about food intake and weight development. Every day, the participants were asked to report eaten meals (breakfast, lunch, dinner, evening), the number of glasses of fluid (water, coffee, milk, juice, soda, etc.) that they consumed. They were also asked about their weight and whether they have consumed alcohol or not. To increase the reporting rate, we used the PMSys push-messaging system to send reminders to the participants' smartphones. A one-time questionnaire (see dataset home page) was used to ask for age, gender, height, and whether the person has a Type A or Type B personality [14]. Most participants regard themselves as having a Type A personality, and generally wakes up early (potentially also goes to bed early), rather than one who wakes up late (Type B).

2.4 Food Images

The reports on eaten meals collected using the Google Forms questionnaire indicate how often and regularly a person consumes food, but leaves out important details about their content, like nutrients and calories. Therefore, selected participants were asked to take photos of everything they have been eating or drinking using their smartphones. This is a time-consuming task and hard to remember activity, i.e., severely influencing the daily behavior of the participants. The collection period is therefore limited to two months.

3 DATASET DETAILS

PMDATA contains data collected from 16 persons: twelve men and three women, in the age range 25–60 years, with an average age of 34 years. The reporting period is from the start of November 2019 to the end of March 2020. The participants range from a broad background with regards to training and exercises. Some are active athletes, some previous athletes, and some rarely exercised at all.

An overview of the participants' demographic information is provided in the *participant-overview.xlsx* file where information like age, height, gender, measured max heart rate, test run results, and walk and run stride lengths are included. Furthermore, there is a directory per participant that contains the data from the Fitbit, PMSys, Google Forms, and Food image data sources. An overview of the dataset ontology can be found in Figure 3. Statistics about the Fitbit JSON-files can be found in Table 1 and statistics about the CSV-Files can be found in Figure 4. Note that all files have timestamps that must be used to connect the data from the different files.

Categories	File	Rate of entries	Number of entries
Calories	calories.json	Per minute.	3377529
Steps	steps.json	Per minute.	1534705
Distance	distance.json	Per minute.	1534705
Sleep	sleep.json	When it happens (usually daily).	2064
Lightly active minutes	lightly_active_minutes.json	Per day.	2244
Moderately active minutes	moderately_active_minutes.json	Per day.	2396
Very active minutes	very_active_minutes.json	Per day.	2396
Sedentary minutes	sedentary_minutes.json	Per day.	2396
Heart rate	heart_rate.json	Per 5 seconds.	20991392
Time in heart rate zones	time_in_heart_rate_zones.json	Per day.	2178
Resting Heart Rate	resting_heart_rate.json	Per day.	1803
Excercise	excercise.json	When it happens. 100 entries per file.	2440
Sleep Score	sleep_score.csv	When it happens (usually daily).	1836
Google Forms reporting	reporting.csv	Per day.	1569
Wellness	wellness.csv	Per day.	1747
Injury	injury.csv	Per week.	225
SRPE	srpe.csv	Per exercise.	783

Fitbit | Google Forms | PMSys

Figure 3: Overview of the dataset.

All participants have been informed about the collection and publication of the data related to this project and signed a form consenting to this. The Norwegian Centre for Research Data (NSD) has evaluated the project and found it to be in accordance with Norwegian and EU data protection laws.

The dataset is available at the Open Science Framework (OSF) at the following URL: https://osf.io/vx4bk/; or at the Simula datasets site: https://datasets.simula.no/pmdata/. The dataset is free to use for research and teaching purposes under the license Attribution-NonCommercial 4.0 International (CC BY-NC 4.0).[5]

3.1 Fitbit

The data from the Fitbit Versa 2 smartwatch has been extracted into CSV and JSON files. The *fitbit* directory contains the following files:

---

[5]https://creativecommons.org/licenses/by-nc/4.0/

Table 1: Number of Fitbit entries for each participant.

| participant	very and moderately active minutes	sleep	sleep score	calories	heart rate	steps and distance	sedentary minutes	exercise	light active minutes	time in heart rate zones	resting heart rate
P01	152	155	150	218880	1573165	218836	152	190	152	152	152
P02	152	158	138	218880	1472629	107326	152	324	152	148	91
P03	152	84	74	218880	808341	53042	152	57	152	117	152
P04	152	188	140	218473	1571315	86457	152	161	152	146	35
P05	152	133	117	218880	1370967	111231	152	145	152	145	95
P06	152	165	147	218880	1579882	117780	152	161	152	152	152
P07	148	161	140	212816	1581947	108048	148	176	148	147	148
P08	143	143	132	205920	1613326	100451	143	261	143	139	143
P09	152	142	132	218880	1305520	85271	152	54	152	150	152
P10	148	103	98	213120	1083257	75427	148	140	148	114	148
P11	152	128	119	218880	1383149	92982	152	96	152	123	98
P12	152	8	1	218880	801264	83752	152	93	0	134	0
P13	152	57	47	634746	48629	152	50	152	80	0
P14	140	138	115	129600	1251156	68703	140	270	140	135	140
P15	145	148	140	208800	1563024	98198	145	243	145	144	145
P16	152	153	146	218880	1397704	78572	152	19	152	152	152
Mean	150	129	115	211096	1311962	95919	150	153	150	136	129
All	2396	2064	1836	3377529	20991392	1534705	2396	2440	2244	2178	1803

(a) PMSys

(b) Google Docs
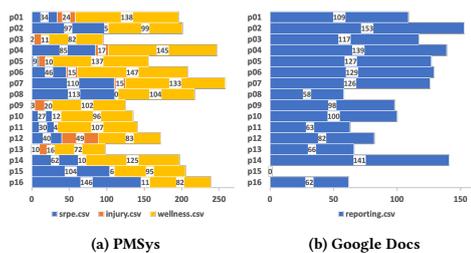
Figure 4: Number of self-reports.

calories.json shows how many calories the person has burned the last minute.

distance.json gives the distance moved per minute. Distance is in centimeters.

exercise.json describes each activity in more detail. It contains the date with start and stop time, time in different activity levels, type of activity, and various performance metrics depending on the type of exercise, e.g., for running, it contains distance, time, steps, calories, speed, and pace.

heart_rate.json shows the number of heartbeats per minute (bpm) at a given time.

sedentary_minutes.json sums up the number of sedentary minutes per day.

lightly_active_minutes.json sums up the number of lightly active minutes per day.

moderately_active_minutes.json sums up the number of moderately active minutes per day.

very_active_minutes.json sums up the number of very active minutes per day.

resting_heart_rate.json gives the resting heart rate per day.

sleep_score.csv helps understand the sleep each night so you can see trends in the sleep patterns. It contains an overall 0-100 score calculated from the composition, revitalization

and duration scores, the number of deep sleep minutes, the resting heart rate, and a restlessness score.

sleep.json is a per sleep breakdown of the sleep into periods of light, deep, REM sleeps, and time awake.

steps.json displays the number of steps per minute.

time_in_heart_rate_zones.json gives the number of minutes in different heart rate zones. Using the common formula of 220 minus your age to find the max heart rate, Fitbit[6] will calculate your maximum heart rate and then create three target heart rate zones — fat burn (50 to 69 percent of your max heart rate), cardio (70 to 84 percent of your max heart rate), and peak (85 to 100 percent of your max heart rate).

As can be observed, there are various parameters included. For example, as we can see in Table 1, in total, there are 2,440 activity sessions (manual and 15-min-auto reports), 20,991,392 heart rate measurements, and 1,836 days of sleep scores included. It can, of course, be discussed how accurate data from a smartwatch can be. For example, we have observations that indicate that the Versa step-counter is influenced by other activities than walking or running and that the estimated distances are slightly inaccurate. For heart rates, the watch seems to be surprisingly accurate when we performed small comparisons using several devices at the same time. Thus, the Fitbit Versa 2 is not the best watch on the market, and the absolute values might be slightly off. However, the collected data should give reasonable indications of activities, and the relative differences between logs at least show if there have been positive or negative changes.

3.2 PMSys

In terms of subjective PMSys reporting, there are three CSV-files:

srpe.csv contains a training session's end-time, type of activity, the perceived exertion (RPE), and the duration in the number of minutes. This is, for example, used to calculate the session's training load or sRPE (RPE × duration).

wellness.csv includes parameters like time and date, fatigue, mood, readiness, sleep duration (number of hours), sleep quality, soreness (and soreness area), and stress. Fatigue, sleep quality, soreness, stress, and mood all have a 1-5 scale. Score 3 is normal, and 1-2 are scores below normal, and 4-5 are scores above normal. Sleep length is just a measure of how long the sleep was in hours, and readiness (scale 0-10) is an overall subjective measure of how ready you are to exercise, i.e., 0 means not ready at all, and 10 indicates that you cannot feel any better and are ready for anything!

injury.csv shows injuries with a time and date and corresponding injury locations and a minor and major severity.

Discussions in many fora are about the accuracy of subjective reports, as one is completely dependent on the truthfulness of the reporter. However, sport is not only a physical activity, and an athlete's psychological "state-of-mind" may greatly influence the performance. Thus, if reported correctly, the subjective information may be of huge value, and there may be important information to be found and predicted [18, 21]. In total, as seen in Figures 3 and 4a, there are 783 training sessions, 1,747 wellness reports, and 225
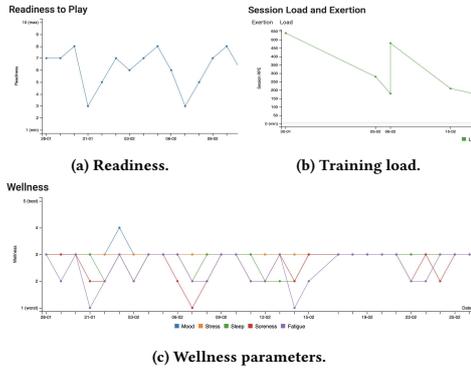
---

[6]https://blog.fitbit.com/max-heart-rate-by-age/

(a) Readiness.   **(b) Training load.**

(c) Wellness parameters.

Figure 5: Examples of PMSys data that can be extracted.

Figure 6: Weight development of a few selected participants.

injury reports submitted. From the figure, we can see the difference in reporting activity among the participants. The plots from the PMSys trainer pages in Figure 5 show examples of data that can be retrieved.

3.3 Google Forms

The *googledocs* directory contains the *reporting.csv* data file, which contains daily reporting data. The data file contains one line per report, including the date reported for, timestamp of the report submission time, the eaten meals (breakfast, lunch, dinner, and evening meal), the participants weigh this day, the number of glasses drunk, and whether one has consumed alcohol.

In total, there are 1,569 reports (Figures 3 and 4b). Similarly to the PMSys data, these reports are also subjective, and some data points are missing. Nevertheless, the submitted data gives some indications of consumed food and drinks and might give important insights into calorie intake. Together with reported activity, this can indicate weight loss or gain, as shown in Figure 6.

3.4 Food Images Details

Participants 1, 3, and 5 took pictures of everything they consumed, except water, for two months (February and March 2020). Some example images can be seen in Figure 7. There are 644 images included, divided between the participants. Information about the day and time of capture can be found in the Exif image headers. The participants used their mobile phone cameras to collect the images (iPhone 6s, iPhone X, and iPhone XS). MacOS Photos was used to export the photos in full quality.

Figure 7: Examples of the captured images of food and drink.

4 INITIAL EXPERIMENTS

To demonstrate how the PMData dataset can be used, this section shows how machine learning can be applied to the data to predict weight gain or loss. More precisely, we define this as the problem of predicting weight change for the next day based on what was reported by the user the previous day. We model this as a classification problem, where we try to classify data from one day into three possible weight change classes for the next day. The three classes are: (0) weight goes down, (1) weight goes up, and (2) no weight change. For these experiments, we are using the following data sources from the PMData dataset: (i) Google doc reports, (ii) PMys wellness reports, and (iii) Fitbit sleep scores. We chose these three to show how the different data within the dataset can be combined and because we also had an intuition that well-being and sleep might correlate with weight change. The exact features used are weight_previous_day, water, alcohol, breakfast, lunch, dinner, evening, fatigue, mood, readiness, sleep_d, sleep_q, soreness, stress, overall_score, composition_score, and revitalization_score. We used only entries from the dataset that had at least the weight reported. Some of the data instances are missing values due to not being reported. We replaced the missing values with zeros[7]. This lead to a total of 1578 data instances. The distributions between the classes are 229 with weight goes down, 247 with weight goes up, and 1102 with no change of weight.

All experiments are performed using 10-fold cross-validation. The experiments are performed using two different algorithms: Random Forest and Classification Decision Tree (CDT). As a baseline, we provide ZeroR (majority class baseline). For all tested algorithms, we report the following metrics: false positive rate, precision, recall, F1-score, and Matthew Correlation Coefficient (MCC).

Table 2 shows the results for the experiments using all features. We can see that both Random Forest and CDT outperform the ZeroR baseline. The best classifier is CDT, with an MCC of weighted average MCC of 0.450. Predicting that weight goes down or up seems equally difficult. One might think that using the previous day's weight is a very important feature. To test this, we also conducted experiments with the two best working classifiers where the previous day's weight is removed as a feature. The results are presented in Table 3, where we can observe that the performance drops significantly. Both methods are having problems beating the majority class baseline significantly if the weight of the previous day is excluded as a feature. For this scenario, Random Forest is better than CDT, with an MCC of 0.259.

---

[7]We also tested to remove them, but replacing with zeros got overall a better score than removing the entries completely.

Thambawita et al.

Table 2: Classification performance (10-fold cross-validation) including weight previous day feature.

Classifier	Class	False-Positive Rate	Precision	Recall	F1-Score	MCC
ZeroR baseline	weighted average	0.698	0.000	0.698	0.000	0.000
Random Forest	weight up	0.062	0.468	0.296	0.362	0.284
Random Forest	weight down	0.060	0.426	0.262	0.324	0.249
Random Forest	no change	0.532	0.802	0.933	0.863	0.471
Random Forest	weighted average	0.390	0.695	0.736	0.706	0.410
CDT	weight up	0.056	0.513	0.316	0.391	0.320
CDT	weight down	0.056	0.503	0.336	0.403	0.333
CDT	no change	0.504	0.811	0.937	0.870	0.504
CDT	weighted average	0.369	0.720	0.753	0.727	0.450

Table 3: Classification performance (10-fold cross-validation) excluding weight previous day feature.

Classifier	Class	False-Positive Rate	Precision	Recall	F1-Score	MCC
ZeroR baseline	weighted average	0.698	0.000	0.698	0.000	0.000
Random Forest	weight up	0.043	0.387	0.146	0.212	0.159
Random Forest	weight down	0.050	0.299	0.127	0.178	0.112
Random Forest	no change	0.725	0.751	0.946	0.838	0.313
Random Forest	weighted average	0.520	0.629	0.702	0.644	0.259
CDT	weight up	0.032	0.276	0.065	0.105	0.064
CDT	weight down	0.033	0.318	0.092	0.142	0.103
CDT	no change	0.821	0.731	0.965	0.832	0.244
CDT	weighted average	0.583	0.600	0.697	0.618	0.195

5 APPLICATIONS OF THE DATASET

PMData contains a large number of logged parameters that can be used for various analyzes like classification and prediction of a person's well-being and sports performance. Some examples using various selections of parameters include predicting a person's readiness to train for training planning, selecting the best team for the next competition, differences between genders or age, the results of the next competition, etc. The combination of the various parameters gives a unique opportunity to better find, for example, the total training load of a person, at an individual level, including data from even outside the training sessions. Thus, it is of large interest from the sports science point of view. Additionally, from a technical point of view, the time-series dataset is noisy, making it a challenge to analyze where one must handle missing data and find outliers, and the possibility to fuse various data sources raises diverse challenges. We plan to use the dataset for future projects, one being a system using PMData to estimate health states [16].

6 CONCLUSION

We have presented the PMData sports logging dataset, containing both objective and subjective parameters from sport and lifelogging, enabling the development of several interesting analysis applications. Our initial experiments show that such analyses are possible, but the dataset has great potential beyond what we have demonstrated in this paper. Other researchers using the dataset might want to look into some of the applications described in the application of the dataset section or come up with entirely new experiments and hypotheses.

REFERENCES

[1] Manal Chokr and Shady Elbassuoni. 2017. Calories prediction from food images. In *Proc. of the 29th Innovative Applications of Artificial Intelligence (IAAI) Conference*.

[2] Munmun De Choudhury, Emre Kiciman, Mark Dredze, Glen Coppersmith, and Mrinal Kumar. 2016. Discovering shifts to suicidal ideation from mental health content in social media. In *Proc. of the 2016 Conference on Human Factors in Computing Systems (CHI)*. 2098–2110.

[3] Tung Duy Dinh, Dinh-Hieu Nguyen, and Minh-Triet Tran. 2018. Social Relation Trait Discovery from Visual LifeLog Data with Facial Multi-Attribute Framework. In *Proc. of the 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM)*. 665–674.

[4] Aaron Duane and Cathal Gurrin. 2019. User interaction for visual lifelog retrieval in a virtual environment. In *Multimedia Modeling (LNCS)*, Vol. 11295. Springer, 239–250.

[5] Aaron Duane and Cathal Gurrin. 2020. Baseline Analysis of a Conventional and Virtual Reality Lifelog Retrieval System. In *MultiMedia Modeling (LNCS)*, Vol. 11962. Springer, 412–423.

[6] Peter Düking, Silvia Achtzehn, Hans-Christer Holmberg, and Billy Sperlich. 2018. Integrated Framework of Load Monitoring by a Combination of Smartphone Applications, Wearables and Point-of-Care Testing Provides Feedback that Allows Individual Responsive Adjustments to Activities of Daily Living. *Sensors* 18, 5 (2018).

[7] Peter Düking, Franz Konstantin Fuss, Hans-Christer Holmberg, and Billy Sperlich. 2018. Recommendations for Assessment of the Reliability, Sensitivity, and Validity of Data Provided by Wearable Sensors Designed for Monitoring Physical Activity. *JMIR mHealth and uHealth* 6, 4 (30 April 2018), e102.

[8] Peter Düking, Christian Stammel, Billy Sperlich, Shaun Sutehall, Borja Muñiz-Pardos, Giscard Lima, Liam P Kilduff, Iphigenia Keramitsoglou, Guoping Li, Fabio Pigozzi, and Yannis P. Pitsiladis. 2018. Necessary Steps to Accelerate the Integration of Wearable Sensors Into Recreation and Competitive Sports. *Current sports medicine reports* 17, 6 (2018), 178–182.

[9] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. 2016. NTCIR Lifelog: The first test collection for lifelog research. In *Proc. of the International ACM SIGIR conference on Research and Development in Information Retrieval*. 705–708.

[10] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, Rashmi Gupta, Rami Albatal, Dang Nguyen, and Duc Tien. 2017. Overview of NTCIR-13 Lifelog-2 task. In *Proc. of the 13th NTCIR Conference on Evaluation of Information Access Technologies*. NTCIR.

[11] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. 2014. Lifelogging: Personal big data. *Foundations and Trends in Information Retrieval* 8, 1 (June 2014), 1–125.

[12] Bogdan Ionescu, Henning Müller, Renaud Péteri, Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Minh-Triet Tran, Mathias Lux, Cathal Gurrin, Yashin Dicente Cid, Vitali Liauchuk, Vassili Kovalev, Asma Ben Abacha, Sadid A. Hasan, Vivek Datla, Joey Liu, Dina Demner-Fushman, Obioma Pelka, Christoph M. Friedrich, Jon Chamberlain, Adrian Clark, Alba García Seco de Herrera, Narciso Garcia, Ergina Kavallieratou, Carlos Roberto del Blanco, Carlos Cuevas Rodríguez, Nikos Vasillopoulos, and Konstantinos Karampidis. 2019. ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction (LNCS)*, Vol. 11696. Springer, 358–386.

[13] Håvard Johansen, Cathal Gurrin, and Dag Johansen. 2015. Towards consent-based lifelogging in sport analytic. In *Multimedia Modeling (LNCS)*, Vol. 8936. Springer, 335–344.

[14] D. W. Johnston. 1993. The current status of the coronary prone behaviour pattern. *Journal of the Royal Society of Medicine* 86, 7 (1993), 406.

[15] Nguyen-Khang Le, Dieu-Hien Nguyen, Trung-Hieu Hoang, Thanh-An Nguyen, Thanh-Dat Truong, Duy-Tung Dinh, Quoc-An Luong, Viet-Khoa Vo-Ho, Vinh-Tiep Nguyen, and Minh-Triet Tran. 2019. HCMUS at the NTCIR-14 Lifelog-3 Task. In *Proc. of the NTCIR Conference on Evaluation of Information Access Technologies*. 48–60.

[16] Nitish Nag. 2020. *Health State Estimation*. Ph.D. Dissertation. University of California, Irvine.

[17] N. Nag and R. Jain. 2019. A Navigational Approach to Health: Actionable Guidance for Improved Quality of Life. *Computer* 52, 4 (April 2019), 12–20.

[18] Svein A. Pettersen, Håvard D. Johansen, Ivan A. M. Baptista, Pål Halvorsen, and Dag Johansen. 2018. Quantified Soccer Using Positional Data: A Case Study. *Frontiers in Physiology* 9 (2018).

[19] Thanh-Dat Truong, Tung Dinh-Duy, Vinh-Tiep Nguyen, and Minh-Triet Tran. 2018. Lifelogging retrieval based on semantic concepts fusion. In *Proceedings of the ACM Workshop on The Lifelog Search Challenge (LSC)*. 24–29.

[20] Kennet Vuong. 2015. *PmSys: a monitoring system for sports athlete load, wellness & injury monitoring*. Master's thesis. University of Oslo.

[21] Theodor Wiik, Håvard D Johansen, Svein-Arne Pettersen, Ivan Baptista, Tomas Kupka, Dag Johansen, Michael Riegler, and Pål Halvorsen. 2019. Predicting Peek Readiness-to-Train of Soccer Players Using Long Short-Term Memory Recurrent Neural Networks. In *Proc. of the 2019 International Conference on Content-Based Multimedia Indexing (CBMI)*. IEEE, 1–6.

A.4 Paper IV - PSYKOSE: A Motor Activity Database of Patients with Schizophrenia

Authors: Petter Jakobsen, Enrique Garcia-Ceja, Lena Antonsen Stabell, Ketil Joachim Oedegaard, Jan Oystein Berle, **Vajira Thambawita**, Steven Alexander Hicks, Pål Halvorsen, Ole Bernt Fasmer, Michael Alexander Riegler

Abstract: Using sensor data from devices such as smart-watches or mobile phones is very popular in both computer science and medical research. Such movement data can predict certain health states or performance outcomes. However, in order to increase reliability and replication of the research it is important to share data and results openly. In medicine, this is often difficult due to legal restrictions or to the fact that data collected from clinical trials is seen as very valuable and something that should be kept "in-house". In this paper, we therefore present PSYKOSE, a publicly shared dataset consisting of motor activity data collected from body sensors. The dataset contains data collected from patients with schizophrenia. Schizophrenia is a severe mental disorder characterized by psychotic symptoms like hallucinations and delusions, as well as symptoms of cognitive dysfunction and diminished motivation. In total, we have data from 22 patients with schizophrenia and 32 healthy control persons. For each person in the dataset, we provide sensor data collected over several days in a row. In addition to the sensor data, we also provide some demographic data and medical assessments during the observation period. The patients were assessed by medical experts from Haukeland University hospital. In addition to the data, we provide a baseline analysis and possible use-cases of the dataset.

Published: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)

Candidate contributions: Vajira contributed to hosting the data on a public data repository (osf.io) and preparing the corresponding wiki pages as a manual for users who will use the dataset. He contributed to revising the manuscript based on the reviews.

Thesis objectives: Sub-objective I, Sub-objective II

PSYKOSE: A Motor Activity Database of Patients with Schizophrenia

Petter Jakobsen
Haukeland University Hospital, Norway
petter.jakobsen@helse-bergen.no

Enrique Garcia-Ceja
SINTEF Digital, Norway
enrique.garcia-ceja@sintef.no

Lena Antonsen Stabell
Haukeland University Hospital, Norway
lena.antonsen.stabell@helse-bergen.no

Ketil Joachim Oedegaard
Haukeland University Hospital, Norway
ketil.joachim.odegaard@helse-bergen.no

Jan Oystein Berle
Haukeland University Hospital, Norway
jan.berle@psyk.uib.no

Vajira Thambawita
SimulaMet, Norway
vajira@simula.no

Steven Alexander Hicks
SimulaMet, Norway
steven@simula.no

Pål Halvorsen
SimulaMet, Norway
paalh@simula.no

Ole Bernt Fasmer
University of Bergen, Norway
Ole.Fasmer@uib.no

Michael Alexander Riegler
SimulaMet, Norway
michael@simula.no

Abstract—Using sensor data from devices such as smartwatches or mobile phones is very popular in both computer science and medical research. Such movement data can predict certain health states or performance outcomes. However, in order to increase reliability and replication of the research it is important to share data and results openly. In medicine, this is often difficult due to legal restrictions or to the fact that data collected from clinical trials is seen as very valuable and something that should be kept "in-house". In this paper, we therefore present PSYKOSE, a publicly shared dataset consisting of motor activity data collected from body sensors. The dataset contains data collected from patients with schizophrenia. Schizophrenia is a severe mental disorder characterized by psychotic symptoms like hallucinations and delusions, as well as symptoms of cognitive dysfunction and diminished motivation. In total, we have data from 22 patients with schizophrenia and 32 healthy control persons. For each person in the dataset, we provide sensor data collected over several days in a row. In addition to the sensor data, we also provide some demographic data and medical assessments during the observation period. The patients were assessed by medical experts from Haukeland University hospital. In addition to the data, we provide a baseline analysis and possible use-cases of the dataset.

Index Terms—Schizophrenia, Actigraphy, Motor Activity, Machine Learning, Artificial Intelligence, Dataset

I. INTRODUCTION

Objective physiological parameters collected from sensors and analyzed by machine learning techniques has gained considerable interest as a tool to support the existing subjective diagnostic practice within mental health [1]. To perform reliable and reproducible research with such data it is important to share both data and results openly. In the medical field, sharing data is often problematic due to various privacy policies. We have previously shared the DEPRESJON dataset [2], containing motor activity data collected from bipolar and unipolar patients. In this paper, we present our second openly shared anonymized dataset on motor activity, containing actigraph data collected from patients with schizophrenia. The Norwegian Regional Medical Research Ethics Committee West approved the original protocol for the study collecting the data for both datasets, and all processes were in accordance with the Helsinki Declaration of 1975 [3].

Actigraphy is a non-invasive method of monitoring human rest and activity cycles, and is normally recorded with a wrist-worn device that registers gravitational acceleration units [3]. Data from actigraphs have been applied to studies of sleep [4] and psychiatric diagnosis like bipolar disorder [5] and ADHD [6], and in some extent in the investigation of Schizophrenia. Schizophrenia is characterized by "positive" symptoms like hallucinations and delusions, "negative" symptoms like diminished motivation and cognitive symptoms like slower mental processing [7]. A recent systematic review summarised motor activity studies of schizophrenia, all applying traditional statistical analysis [8]. Overall, patients with schizophrenia are associated with lower motor activity levels as well as repetitious and rigid patterns of behavior when compared to healthy controls. Motor activity also reflects the symptomatic state. Increasing positive symptoms correlates with augmented complexity in activity patterns and increased sleep disturbance. Increased negative symptoms associates with overall reduced activity and amplified nighttime sleep disturbance [8].

The circadian system, an internal self-regulating clock, regulates the diurnal oscillating cycles of nighttime sleep and daytime activity [9]. Integrated and interlocked in the circadian clock are various ultradian rhythms of shorter duration regulating patterns like rest/activity cycles, feeding habits, and hormone levels. Time series of motor activity is an articulation of this recurring complex clock system in interaction with daily social rhythms [10]. Disturbed sleep patterns and lurched rest/active cycles are characterizing symptoms of schizophrenia [7].

An alternative method to detect and classify schizophrenia is electroencephalography (EEG) measuring electrical activity in the brain [11]. Machine learning appears promising in differentiating between schizophrenic patients and healthy controls

in such data [12]. Still, data collected with electrodes placed on the scalp seems like a substantially more cumbersome and demanding process than a simple wrist-worn actigraph registering motor activity.

The aim of this paper is to provide a comprehensive dataset of motor activity of patients with schizophrenia and make it publicly available. More, to enable additional investigation by sharing the dataset and ideas for further research. The main contributions of this paper are:

1) A new publicly available dataset containing sensor and demographic data of a substantial number of patients with schizophrenia.
2) The dataset contains additionally sensor data from a large number of healthy control persons.
3) Baseline experiments that can be used by other researchers to compare their results. Classifying schizophrenic versus non-schizophrenic patterns, including recommendations for evaluation metrics.

In the following, we describe the diagnosis of Schizophrenia (Section II), how the data was collected and the attributes of the data itself (Section III). Section IV lists some of the potential applications of this dataset. Section V presents some suggested evaluation metrics. This is followed by an experiment section containing the baseline experiments (Section VI). In Section VII we also discuss possible future research questions using the dataset and give a conclusion.

II. MEDICAL BACKGROUND

Schizophrenia is a severe mental disorder that affects approximately one percent of the global population. Symptoms of schizophrenia begin in early adulthood, and the debut age is younger for males than females. The disorder tends to be chronic and relapsing, however with a highly variable disease burden and degree of disability between individuals. A range of different symptoms, including "positive" symptoms like hallucinations, delusions or psycho-motoric agitation, "negative" symptoms like impaired affective experience or expression and diminished motivation, and cognitive symptoms like problems with focus or paying attention and problem solving may occur [7], [13]. The main treatment of schizophrenia is antipsychotic medication, both for acute psychotic episodes and for relapse prevention. The therapeutic effects of, and side effects related to, antipsychotics vary substantially among individuals [7]. Antipsychotics target the dopamine system, and the antidopaminergic effect may influence motor activity through side effects such as extrapyramidal syndrome and akathisia [14]. Akathisia is characterized by subjective and objective psycho-motoric restlessness [15]. These distressing side effects are an important factor for patients quitting their prescribed antipsychotic medications [16]. Akathisia investigated in motor activity studies appears therefore as a relevant and important topic for future research. Unfortunately, this is not possible in the present dataset. In retrospect, we have identified several patient characteristics that would have been beneficial to studies like this, however requiring a larger sample size. Variables like previous antipsychotic use, duration of use, type

of antipsychotic including dosage and alteration of dosage, serum concentration of antipsychotics to verify intake, patient status (inpatient/outpatient), duration of untreated psychosis, alcohol consumption and substance use may all be valuable in further motor activity studies in schizophrenia [15], [17].

III. DATASET DETAILS

Motor activity was collected with a wrist-worn actigraph device (Actiwatch, Cambridge Neurotechnology Ltd, England, model AW4) entailing a piezoelectric accelerometer programmed to record the integration of intensity, amount and duration of movement in x, y and z axes. The sampling frequency was 32 Hz and movements over 0.05 g recorded. The output is an integer value proportional to the movement intensity for 1 minute epochs [3]. Figure 1 shows a 24 hour subset of the actigraphy data produced by the device for one of the patients.
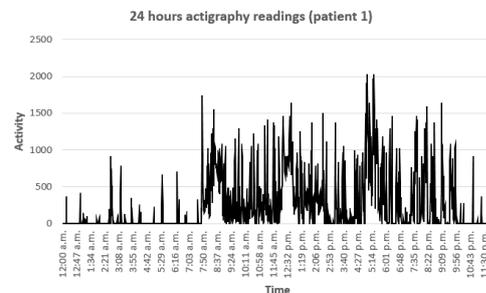

Fig. 1. Example actigraphy data of patient 1 from midnight to midnight. The x axis is time and the y axis is the activity level as stored by the device.

The dataset consists of actigraph data collected from 22 psychotic patients hospitalized at a long-term open psychiatric ward at Haukeland University hospital. All are diagnosed with schizophrenia, and all used antipsychotic medications. The group contained 3 females and 19 males with an average age of 46.2 ± 10.9 years (range $27 - 69$ years). The mean age at first time of hospitalization was 24.8 ± 9.3 (range $10 - 52$ years). Clinical experts diagnosed the patients using a semi-structured interview based on DSM-IV criteria [18]. 17 of the patients were recognized as paranoid schizophrenic. For the other 5 patients no subtype of schizophrenia was specified, beyond that they were non-paranoid. DSM-5, the currently used diagnostic manual do not recognize schizophrenia subtypes [19]. The present psychotic symptomatic state of the patients were rated on the Brief Psychiatric Rating Scale (BPRS), a frequently used rating scale for measuring the overall psychopathology of schizophrenic patients. BPRS consists of 18 items rated from 1 to 7, and higher sum scores indicate a more severe condition [20]. 17 of the patients were rated on the BPRS scale, mean score was 50.0 ± 8.8 (range 34 $- 59$). Further details on the dataset are presented in previous

papers analyzing the dataset with various linear and nonlinear statistical approaches [3], [21]–[23].

The dataset also contains actigraphy data from 32 healthy control persons, consisting of 23 hospital employees, 5 nursing students, and 4 healthy persons recruited from a general practitioner. None had a history of either psychotic or affective disorders. The group consists of 20 females and 12 males, with a mean age of 38.2 ± 13.0 (range $21 - 66$ years). The gender composition is mismatched between the groups. Nevertheless, previous studies of motor activity within mental health have not identified gender differences in activation [24].

The participants used the actigraph devices for an average of 12.7 days in the control and condition groups. See Table I for details. The battery life of the device is about 14 days, thus, it didn't need charging during the study. The total number of collected days was 687 comprising 402 days in the control group and 285 in the condition group. Note that the actigraph files might contain more days, but only the first n days were considered in our analysis where n is the number of days reported in the *days.csv* file. These are the days during which the study took place. Figure 2 shows a boxplot of the average activity per day for the condition and control group. Here, it can be seen that the condition group has lower activity levels compared to the control group.

TABLE I
STATISTICS OF NUMBER OF COLLECTED DAYS BY GROUP.

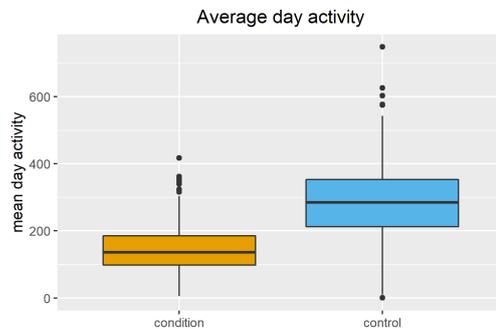|	Control group	Condition group
Mean	12.6	12.95
Sd.	2.3	0.37
Max	20	14
Min	8	12

Fig. 2. Boxplots of average activity per day of the condition and control groups.

A. Dataset Structure

The root folder of the dataset contains five items. Two folders, one contains the activity data for the controls and the other the data for the patients. For each patient and control, a CSV file is provided containing the actigraphy activity measurements over time. The columns in the patient and control files are timestamp (one-minute intervals), date (date of measurement), activity (activity measurement from the actigraph watch). Figure 3 shows an extract of the first 10 lines of data from patient 18.

Fig. 3. First 10 lines of acitgraphy data from patient 18.

The root folder also contains a file named *patients_info.csv*. This file contains the following columns: Number (patient identifier), gender (male or female), age (age of the patient), days (whole days the patient wore the actigraph), schtype (type of schizophrenia), bprs (BPRS sum score), cloz (did the patient use clozapine as antipsychotic medication), trad (did the patient use traditional neuroleptic or modern antipsychotic medication), moodst (did the patient use mood stabilizing medications), agehosp (age first time hospitalized).

Another file in the root folder is named *days.csv*. This file contains the number of days the patient and controls are in the study. It contains the columns id (identifier) and days (number of full days).

Finally, the root folder contains a file named *schizophrenia_features.csv*. This contains the statistical features used for the baseline experiments. The file contains four columns: userid (patient identifier), class (class to predict binary), class_str (class name as string), f.mean (the mean), f.sd (the standard deviation), f.propZeros (proportion of zeros).

The dataset can be accessed via: **https://osf.io/dgjzu/** or directly downloaded from **https://datasets.simula.no/psykose/**. The license for the PSYKOSE dataset is *Creative Commons Attribution-NonCommercial 4.0 International*.

IV. APPLICATIONS OF THE DATASET

The main goal of publishing this dataset is to encourage other researchers to use the data to improve the quality of life for mental health patients. The dataset has several application areas, of which some will be discussed in the following. Some suggested future research directions using this dataset could be:

- Use machine learning for schizophrenia v.s. non-schizophrenia classification.
- Analysis of circadian and ultradian cycles in schizophrenia compared to non-schizophrenia.

- Diurnal and nocturnal activity analysis of schizophrenia versus non-schizophrenia.

We believe that this dataset can be useful to the machine learning community since during the last years the use of machine learning for mental health has shown promising results [1], [25], [26].

In addition, we also want to point out that this dataset can be combined with our previously published *Depresjon* dataset [2], to increase the number of persons and measurements for both datasets. When comparing the motor activity profiles of depressed patients, schizophrenic patients and healthy controls, the distribution and length of active and resting periods differentiate in motor activity [22]. Complexity analyzes have also identified motor activity profiles segregating the three groups [21] [27]. Therefore, by combing these two datasets, some potential applications emerge:

- Use machine learning for schizophrenia, depression state classification.
- Compare attributes of schizophrenia and depression patients.
- Differences in diurnal/nocturnal patterns and/or the rest/activity cycles of schizophrenia versus non-schizophrenia versus depressed.

In addition to these specific medical research questions, more general research questions in the field of machine learning could also be addressed using this dataset. For example, comparing different algorithms and metrics on the dataset, over and under-sampling techniques and their effect measured using the dataset, and researching and developing more advanced time-series based analysis algorithms. Examples of more advanced algorithms include those based on deep learning, such as convolutional neural networks or recurrent neural networks.

V. Suggested Metrics

The evaluation of classification algorithms can be done in a variety of different ways. Sometimes, metrics that measure the same thing have different names depending on the discipline in which they are discussed. For example, recall in information retrieval is often called sensitivity in a medical context. In the following, we will present two experiments using different metrics that we recommend for this dataset. In general, there are two important things to take into account. Firstly, medical datasets are often imbalanced (one class is presented more often than another). For an imbalanced dataset like this, it is important to weigh the metrics based on the number of classes. Such weighting is specifically applicable to binary classifications. Secondly, it is good practice to present a comprehensive set of outcome metrics, beyond the frequently reported limited subset of accuracy or precision, recall, and F1-score.

All outcome metrics we recommend are calculated by using True positives ((TP) number of correctly classified patients with schizophrenia), true negatives ((TN) number of correctly classified controls), false positives ((FP) number of misclassified controls) and false negatives ((FN) number of misclassified patients with schizophrenia). The metrics used for this dataset are, False-Positive Rate, Precision, Recall/Sensitivity, Matthews Correlation Coefficient (MCC) and F1-score. In addition, we recommend using Precision-Recall-Curves (PRC) and Receiver-Operating-Characteristic-Curves (ROC). Additionally, to obtain better generalizable models, a cross-validation approach ought to be utilized. We propose either N-fold or leave-one-patient-out cross-validation.

VI. Baseline Performance

To provide a baseline performance and also to inspire future work, we present two baseline experiments using the dataset. The goal of both experiments is to classify patients into schizophrenia or non-schizophrenia. For all experiments, we used statistical features extracted from the activity data. The features used are standard deviation, proportion of zeros and mean. The features are calculated per full day per patient. That is, one feature vector is extracted per day and per participant. This leads to 687 data points (feature vectors) which corresponds to the total number of study days across all participants. From these, 285 are from schizophrenic patients and 402 from controls. Details about how many days per participant were collected can be found in the *days.csv* file of the dataset. The extracted features used for the experiments are shared with the dataset for reproducibility.

Figure 4 shows a projection of the extracted features into a 2D plane using Multidimensional Scaling (MDS). It can be seen that those features, to some extent, are able to separate both groups but not perfectly, though. In the next few sections, we present baseline results using machine learning classifiers to infer each points' class (no-schizophrenia and schizophrenia).
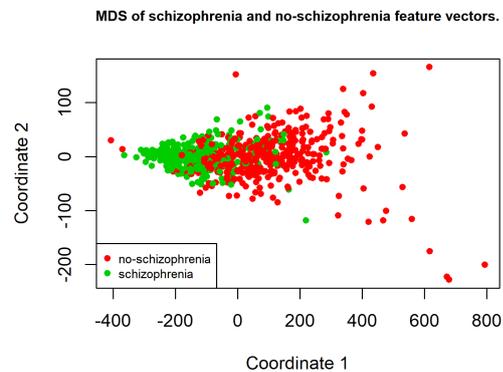
MDS of schizophrenia and no-schizophrenia feature vectors.

Fig. 4. Computed features projected into a 2D plane using MDS.

A. Experiment 1

For the first experiment, we perform 10-fold cross-validation for the training and leave a certain amount of data out for testing (90%, 66%, 50%, 33%, and 10%). The data left out

TABLE II
CLASSIFICATION PERFORMANCE FOR EXPERIMENT 1 (10-FOLDED
CROSS-VALIDATION) REPORTING AVERAGE PRECISION AND AREA UNDER
THE CURVE FOR DIFFERENT TEST SET SIZES.

Metric	Testset size	LR	RF	XGB	LGB	Ensemble
Average Precision	10	0.89	0.86	0.86	0.89	0.92
Average Precision	33	0.94	0.90	0.90	0.93	0.91
Average Precision	50	0.94	0.89	0.89	0.93	0.91
Average Precision	66	0.92	0.89	0.90	0.92	0.89
Average Precision	90	0.92	0.82	0.82	0.89	0.90
Area under the Curve	10	0.81	0.78	0.82	0.85	0.91
Area under the Curve	33	0.90	0.85	0.86	0.90	0.88
Area under the Curve	50	0.89	0.84	0.86	0.90	0.89
Area under the Curve	66	0.89	0.83	0.87	0.90	0.87
Area under the Curve	90	0.88	0.76	0.79	0.84	0.87

for testing is stratified, meaning the number of schizophrenic and non-schizophrenic data points is balanced if possible (this cannot be done for the 90% test data case).

The experiments are performed using four different algorithms, namely, Logistic Regression (LR) [28], Random Forest (RF) [29], Extreme Gradient Boosting (XGB) [30] and Light Gradient Boosting (LGB) [31]. All four are commonly used for machine learning tasks. In addition, we also used ensemble to combine the four different algorithms to perform a combined classification. For all tested algorithm,s we report the average precision (from the PRC) and the area under the curve (from the ROC). For the best working one, we also present plots of the PRC and ROC. Implementations are made using Scikit-learn [32] and the packages XGBoost[1] and LightGBM[2] for the two respective algorithms. The implementation details and configurations are shared with the dataset.

Looking at table II, we can observe that all algorithms perform well with average precision and area under the curve above 0.80. Overall, the logistic regression performs best in terms of average precision and area under the curve. Figure 5 shows the precision-recall curve for the LR and 90% of the data as a testset. It is interesting to see that the performance is very good, even with a small number of training data. The random baseline threshold would be 0.41 (true positive divided by all samples). For the ROC shown in Figure 6, we can make the same observation with an area under the curve of 0.92.

B. Experiment 2

For experiment 2, we changed the evaluation of cross-validation training and separate test set to leave one patient out cross-validation. This means we leave one patient out of the training and use that for testing. This is repeated until all patients have been assigned once to the test set. For these experiments, we used the WEKA [33] machine learning library. We are reporting the weighted average of the metrics. The tested algorithms are ZeroR (which is the majority class baseline), Random Tree (RT), Random Forest (RF), and classification via Regression (CVR). From the results in table III, we can see that all algorithms outperform the ZeroR baseline. Looking at the Matthews correlation

[1]https://xgboost.readthedocs.io/en/latest/index.html
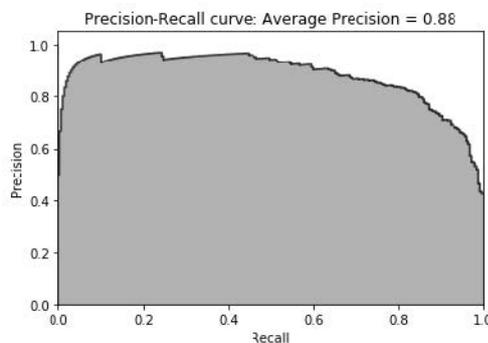[2]https://github.com/microsoft/LightGBM

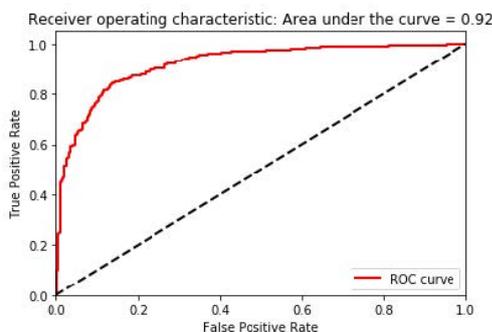Fig. 5. PRC for the logistic regression using 90% of the data as testset.

Fig. 6. ROC for the logistic regression using 90% of the data as testset.

TABLE III
CLASSIFICATION PERFORMANCE (LEAVE ONE PATIENT OUT CROSS
VALIDATION). THE BEST PERFORMING CLASSIFIER ON THE WEIGHTED
AVERAGE IS BOLD.

Classifier	Class	False-Positive Rate	Precision	Recall	F1-Score	MCC
RT	Non-Schizophrenia	0.232	0.835	0.828	0.831	0.596
RT	Schizophrenia	0.172	0.760	0.768	0.764	0.596
RT	weighted average	0.207	0.804	0.803	0.804	0.596
RF	Non-Schizophrenia	0.232	0.844	0.886	0.864	0.662
RF	Schizophrenia	0.114	0.826	0.768	0.796	0.662
RF	**weighted average**	**0.183**	**0.836**	**0.837**	**0.836**	**0.662**
CVR	Non-Schizophrenia	0.098	0.906	0.672	0.771	0.570
CVR	Schizophrenia	0.328	0.661	0.902	0.763	0.570
CVR	weighted average	0.194	0.804	0.767	0.768	0.570
ZeroR	Non-Schizophrenia	1.000	0.585	1.000	0.738	0
ZeroR	Schizophrenia	0.000	0	0.000	0	0
ZeroR	weighted average	0.585	0.515	0.585	0.515	0

coefficient (MCC), we can see that Random Forest is the overall best performing classifier. The other two algorithms seem to have a problem to efficiently detect schizophrenia compared to non-schizophrenia.

Appendix A. Published Articles

C. Experiments Summary

Both sets of experiments showed promising results for using activity data to detect schizophrenia versus non-schizophrenia. However, the results are not optimal, and there is still potential for large improvements. For example, it might be better to look at the complete activity using more sophisticated methods such as recurrent neural networks.

VII. CONCLUSIONS

In this paper, we have presented a dataset containing motor activity data from patients with schizophrenia. The baseline analysis of our experimental results showed the potential for using such data to answer medical relevant research questions. We also discussed possible applications using the dataset such as schizophrenia versus non-schizophrenia classification of patients. In this respect, we hope that this dataset will encourage other researchers to both perform experiments using the data, and also to share their own insights and datasets. The PSYKOSE dataset will hopefully enable reproducible and comparable results and assist in the development of future automated systems supporting the existing subjective diagnostic practice within mental health.

REFERENCES

[1] E. Garcia-Ceja, M. Riegler, T. Nordgreen, P. Jakobsen, K. J. Oedegaard, and J. Tørresen, "Mental health monitoring with multimodal sensing and machine learning: A survey," *Pervasive and Mobile Computing*, vol. 51, pp. 1–26, 2018.

[2] E. Garcia-Ceja, M. Riegler, P. Jakobsen, J. Tørresen, T. Nordgreen, K. J. Oedegaard, and O. B. Fasmer, "Depresjon: a motor activity database of depression episodes in unipolar and bipolar patients," in *Proceedings of the 9th ACM Multimedia Systems Conference*, 2018, pp. 472–477.

[3] J. O. Berle, E. R. Hauge, K. J. Oedegaard, F. Holsten, and O. B. Fasmer, "Actigraphic registration of motor activity reveals a more structured behavioural pattern in schizophrenia than in major depression," *BMC research notes*, vol. 3, no. 1, p. 149, 2010.

[4] E. C. Winnebeck, D. Fischer, T. Leise, and T. Roenneberg, "Dynamics and ultradian structure of human sleep in real life," *Current Biology*, vol. 28, no. 1, pp. 49–59, 2018.

[5] J. Scott, G. Murray, C. Henry, G. Morken, E. Scott, J. Angst, K. R. Merikangas, and I. B. Hickie, "Activation in bipolar disorders: a systematic review," *JAMA psychiatry*, vol. 74, no. 2, pp. 189–196, 2017.

[6] G. L. Faedda, K. Ohashi, M. Hernandez, C. E. McGreenery, M. C. Grant, A. Baroni, A. Polcari, and M. H. Teicher, "Actigraph measures discriminate pediatric bipolar disorder from attention-deficit/hyperactivity disorder and typically developing controls," *Journal of Child Psychology and Psychiatry*, vol. 57, no. 6, pp. 706–716, 2016.

[7] S. R. Marder and T. D. Cannon, "Schizophrenia," *New England Journal of Medicine*, vol. 381, no. 18, pp. 1753–1761, 2019.

[8] Z. Y. Wee, S. W. L. Yong, Q. H. Chew, C. Guan, T. S. Lee, and K. Sim, "Actigraphy studies and clinical and biobehavioural correlates in schizophrenia: a systematic review," *Journal of Neural Transmission*, vol. 126, no. 5, pp. 531–558, 2019.

[9] T. Bollinger and U. Schibler, "Circadian rhythms–from genes to physiology and disease," *Swiss medical weekly*, vol. 144, no. 2930, 2014.

[10] C. Bourguignon and K.-F. Storch, "Control of rest: activity by a dopaminergic ultradian oscillator and the circadian clock," *Frontiers in neurology*, vol. 8, p. 614, 2017.

[11] Z. Dvey-Aharon, N. Fogelson, A. Peled, and N. Intrator, "Schizophrenia detection and classification by advanced analysis of eeg recordings using a single electrode approach," *PloS one*, vol. 10, no. 4, 2015.

[12] L. Zhang, "Eeg signals classification using machine learning for the identification and diagnosis of schizophrenia," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2019, pp. 4521–4524.

[13] R. Tandon, H. A. Nasrallah, and M. S. Keshavan, "Schizophrenia,"just the facts" 4. clinical features and conceptualization," *Schizophrenia research*, vol. 110, no. 1-3, pp. 1–23, 2009.

[14] E. Johnsen, O. B. Fasmer, H. van Wageningen, K. Hugdahl, E. Hauge, and H. A. Jørgensen, "The influence of glutamatergic antagonism on motor variability, and comparison to findings in schizophrenia patients," *Acta neuropsychiatrica*, vol. 25, no. 2, pp. 105–112, 2013.

[15] T. Pringsheim, D. Gardner, D. Addington, D. Martino, F. Morgante, L. Ricciardi, N. Poole, G. Remington, M. Edwards, A. Carson *et al.*, "The assessment and treatment of antipsychotic-induced akathisia," *The Canadian Journal of Psychiatry*, vol. 63, no. 11, pp. 719–729, 2018.

[16] J. E Thomas, J. Caballero, and C. A Harrington, "The incidence of akathisia in the treatment of schizophrenia with aripiprazole, asenapine and lurasidone: a meta-analysis," *Current neuropharmacology*, vol. 13, no. 5, pp. 681–691, 2015.

[17] E. Sacchetti, P. Valsecchi, E. Tamussi, L. Paulli, R. Morigi, and A. Vita, "Psychomotor agitation in subjects hospitalized for an acute exacerbation of schizophrenia," *Psychiatry research*, vol. 270, pp. 357–364, 2018.

[18] A. P. Association *et al.*, "Diagnostic and statistical manual of mental disorders (4th éd.)," *Washington, DC: Author*, 1994.

[19] ——, *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub, 2013.

[20] S. Leucht, J. M. Kane, W. Kissling, J. Hamann, E. Etschel, and R. Engel, "Clinical implications of brief psychiatric rating scale scores," *The British Journal of Psychiatry*, vol. 187, no. 4, pp. 366–371, 2005.

[21] E. R. Hauge, J. Ø. Berle, K. J. Oedegaard, F. Holsten, and O. B. Fasmer, "Nonlinear analysis of motor activity shows differences between schizophrenia and depression: a study using fourier analysis and sample entropy," *PloS one*, vol. 6, no. 1, 2011.

[22] O. B. Fasmer, E. Hauge, J. Ø. Berle, S. Dilsaver, and K. J. Oedegaard, "Distribution of active and resting periods in the motor activity of patients with depression and schizophrenia," *Psychiatry investigation*, vol. 13, no. 1, p. 112, 2016.

[23] E. E. Fasmer, O. B. Fasmer, J. Ø. Berle, K. J. Oedegaard, and E. R. Hauge, "Graph theory applied to the analysis of motor activity in patients with schizophrenia and depression," *PloS one*, vol. 13, no. 4, 2018.

[24] J. Scott, A. E. Vaaler, O. B. Fasmer, G. Morken, and K. Krane-Gartiser, "A pilot study to determine whether combinations of objectively measured activity parameters can be used to differentiate between mixed states, mania, and bipolar depression," *International journal of bipolar disorders*, vol. 5, no. 1, p. 5, 2017.

[25] N. C. Jacobson, H. Weingarden, and S. Wilhelm, "Digital biomarkers of mood disorders and symptom change," *NPJ digital medicine*, vol. 2, no. 1, pp. 1–3, 2019.

[26] J. G. Rodríguez-Ruiz, C. E. Galván-Tejada, L. A. Zanella-Calzada, J. M. Celaya-Padilla, J. I. Galván-Tejada, H. Gamboa-Rosales, H. Luna-García, R. Magallanes-Quintanar, and M. A. Soto-Murillo, "Comparison of night, day and 24 h motor activity data for the classification of depressive episodes," *Diagnostics*, vol. 10, no. 3, p. 162, 2020.

[27] K. Krane-Gartiser, T. E. Henriksen, G. Morken, A. E. Vaaler, and O. B. Fasmer, "Motor activity patterns in acute schizophrenia and other psychotic disorders can be differentiated from bipolar mania and unipolar depression," *Psychiatry research*, vol. 270, pp. 418–425, 2018.

[28] G. Gasso, "Logistic regression," 2019.

[29] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[30] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[31] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

A.5 Paper V - Kvasir-Capsule, a Video Capsule Endoscopy Dataset

Authors: Pia H. Smedsrud, **Vajira Thambawita**, Steven A. Hicks, Henrik Gjestang, Oda Olsen Nedrejord, Espen Næss, Hanna Borgli, Debesh Jha, Tor Jan Derek Berstad, Sigrun L. Eskeland, Mathias Lux, Håvard Espeland, Andreas Petlund, Duc Tien Dang Nguyen, Enrique Garcia-Ceja, Dag Johansen, Peter T. Schmidt, Ervin Toth, Hugo L. Hammer, Thomas de Lange, Michael A. Riegler, Pål Halvorsen

Abstract: Artificial intelligence (AI) is predicted to have profound effects on the future of video capsule endoscopy (VCE) technology. The potential lies in improving anomaly detection while reducing manual labour. Existing work demonstrates the promising benefits of AI-based computer-assisted diagnosis systems for VCE. They also show great potential for improvements to achieve even better results. Also, medical data is often sparse and unavailable to the research community, and qualified medical personnel rarely have time for the tedious labelling work. We present Kvasir-Capsule, a large VCE dataset collected from examinations at a Norwegian Hospital. Kvasir-Capsule consists of 117 videos which can be used to extract a total of 4,741,504 image frames. We have labelled and medically verified 47,238 frames with a bounding box around findings from 14 different classes. In addition to these labelled images, there are 4,694,266 unlabelled frames included in the dataset. The Kvasir-Capsule dataset can play a valuable role in developing better algorithms in order to reach true potential of VCE technology.

Candidate contributions: Vajira contributed to the main baseline experiments discussed in the paper. He performed the baseline experiments using two different deep learning methods (DenseNet-161 and ResNet-152) using Pytorch deep learning framework. In these baseline experiments, he performed deep analysis using two different loss functions such as Normal Cross-Entropy Loss, Weighted Cross-Entropy Loss, and using a weighted sampling method, and his experiments showed the best results in this paper (Refer to Table 3 in the paper). Vajira contributed to drafting and revising the paper. He especially focused on revising the technical

Appendix A. Published Articles

part of the paper.

Thesis objectives: Sub-objective I, Sub-objective II

scientific **data**

Check for updates

OPEN

DATA DESCRIPTOR

Kvasir-Capsule, a video capsule endoscopy dataset

Pia H. Smedsrud[1,3,6,15]✉, Vajira Thambawita[1,2,15], Steven A. Hicks[1,2,15], Henrik Gjestang[1,3], Oda Olsen Nedrejord[1,3], Espen Næss[1,3], Hanna Borgli[1,3], Debesh Jha[1,7,15], Tor Jan Derek Berstad[6], Sigrun L. Eskeland[4], Mathias Lux[10], Håvard Espeland[6], Andreas Petlund[6], Duc Tien Dang Nguyen[5], Enrique Garcia-Ceja[13], Dag Johansen[7], Peter T. Schmidt[8,9], Ervin Toth[14], Hugo L. Hammer[1,2], Thomas de Lange[4,6,11,12,15,16], Michael A. Riegler[1,15,16] & Pål Halvorsen[1,2,15,16]

Artificial intelligence (AI) is predicted to have profound effects on the future of video capsule endoscopy (VCE) technology. The potential lies in improving anomaly detection while reducing manual labour. Existing work demonstrates the promising benefits of AI-based computer-assisted diagnosis systems for VCE. They also show great potential for improvements to achieve even better results. Also, medical data is often sparse and unavailable to the research community, and qualified medical personnel rarely have time for the tedious labelling work. We present *Kvasir-Capsule*, a large VCE dataset collected from examinations at a Norwegian Hospital. *Kvasir-Capsule* consists of 117 videos which can be used to extract a total of 4,741,504 image frames. We have labelled and medically verified 47,238 frames with a bounding box around findings from 14 different classes. In addition to these labelled images, there are 4,694,266 unlabelled frames included in the dataset. The *Kvasir-Capsule* dataset can play a valuable role in developing better algorithms in order to reach true potential of VCE technology.

Background & Summary

The small bowel constitutes the gastrointestinal (GI) tract's mid-part, situated between the stomach and the large bowel. It is three to four meters long and has a surface of about $30\,m^2$, including the villi's surface. As part of the digestive system, it plays a crucial role in absorbing nutrients[1]. Therefore, disorders in the small bowel may cause severe growth retardation in children and nutrient deficiencies in children and adults[1]. This organ may be affected by chronic diseases, like Crohn's disease, coeliac disease, and angiectasias, or malignant diseases like lymphoma and adenocarcinoma[2,3]. These diseases may represent a substantial health challenge for both the patients and the society, and a thorough examination of the lumen is frequently necessary to diagnose and treat them[4]. However, due to its anatomical location, the small bowel is less accessible for inspection by flexible endoscopes commonly used for the upper GI tract and the large bowel. Since early 2000, video capsule endoscopy (VCE)[5] has been used, usually as a complementary test for patients with GI bleeding[4]. A VCE consists of a small capsule containing a wide-angle camera, light sources, batteries, and other electronics. The patient swallows the capsule capturing a video as it moves passively throughout the GI tract. A recorder, carried by the patient or included in the capsule, stores the video before a medical expert examines it after the procedure.

VCE devices exist in various versions and brands such as Given Imaging (Medtronic), Ankon Technologies, Chongqing Science, IntroMedic, CapsoVision, and Olympus. The frame rate typically varies between 1 and 30 frames per second, capturing in total between 50 and 100 thousand frames, with pixel-resolutions in the range of

[1]SimulaMet, Oslo, Norway. [2]Oslo Metropolitan University, Oslo, Norway. [3]University of Oslo, Oslo, Norway. [4]Department of Medical Research, Bærum Hospital, Gjettum, Norway. [5]University of Bergen, Bergen, Norway. [6]Augere Medical AS, Oslo, Norway. [7]UIT The Arctic University of Norway, Tromsø, Norway. [8]Karolinska Institutet, Department of Medicine, Solna, Sweden. [9]Ersta Hospital, Department of Medicine, Stockholm, Sweden. [10]Klagenfurt University, Wörthersee, Austria. [11]Medical Department, Sahlgrenska University Hospital-Mölndal Hospital, Göteborg, Sweden. [12]Department of Molecular and Clinical Medicine, Sahlgrenska Academy, University of Gothenburg, Göteborg, Sweden. [13]SINTEF Digital, Oslo, Norway. [14]Department of Gastroenterology, Skåne University Hospital, Malmö Lund University, Malmö, Sweden. [15]These authors contributed equally: Pia H. Smedsrud, Vajira Thambawita, Steven A. Hicks, Debesh Jha, Thomas de Lange, Michael A. Riegler, Pål Halvorsen. [16]These authors jointly supervised: Thomas de Lange, Michael A. Riegler, Pål Halvorsen. ✉e-mail: pia@simula.no

Appendix A. Published Articles

Dataset	Findings	Size	Availability
KID[54]	Angiectasia, bleeding, inflammations, polyps	2,371 images + 47 videos	open academic•
GIANA 2017[55]	Angiectasia†	600 images	by request
GIANA2018[56,57]	Polyps and small bowel lesions†	8262 images + 38 videos	by request
CAD-CAP[58,59]	Normal frames, fresh blood, vascular lesion, ulcerative and inflammatory lesions	25,000 images	by request◇
Gastrolab[60]	Crohns diseases, small bowel (video) + GI lesions	Few hundred images and videos	open academic•

Table 1. An overview of existing VCE datasets from the GI tract. †Including ground truth segmentation masks. •Not available anymore. ◇The Computer-Assisted Diagnosis for CAPsule endoscopy (CAD-CAP) Database - used for the angiectasia detection.

....

256×256 to 512×512. Some of the vendors have software to remove duplicated frames due to slow movement. However, a large number of frames need to be analysed by a medical expert, resulting in a tedious and error-prone operation. In the related area of colonoscopy, operator variation and detection performance are reported problems[6–8] resulting in high miss rates[9]. In VCE analysis, essential findings are missed due to lack of concentration, insufficient experience and knowledge[10–12]. Furthermore, physicians may have trouble handling the associated technology, and infrequent VCE use leads to lack of confidence[13], resulting in inter-observer and intra-observer variations in the assessments[12].

The technical developments for automated image and video analysis have sky-rocketed, and multimedia solutions in medicine show tremendous potential[14,15]. An increasing number of promising machine learning solutions are being developed for automated diagnosis of colonoscopies[16–23] using open datasets[24–27]. Regarding automated VCE data analyses, machine learning approaches also produce promising results regarding detection and classification rates[28–35]. Machine learning, or artificial intelligence (AI) in general, is likely to have profound effects on the VCE technology's future, not only for improving variation and detection rates but also for estimating the capsule's localisation[13,36].

Regardless of promising initial results, there is room for improvements in detection rate, reduced manual labour, and AI explainability. Large amounts of data are needed[37,38], particularly annotated data[35], and access to these data are often scarce[39]. As shown in Table 1, very few, small VCE datasets are made publicly available, and several have become unavailable. We have previously published the HyperKvasir dataset[27]. Nevertheless, this and similar datasets containing images from *colonoscopies* and *esophagogastroscopies* are not applicable because they do not depict the small bowel, characterised by the intestinal villi displaying a different surface than the rest of the bowel. Also, the image resolution and the frame rate of VCEs are much lower. The small bowel is not air inflated during a VCE examination, as is the case with traditional colonoscopies. Different optics are also used, and the movement of the capsule is uncontrolled in contrast to flexible endoscopes used during manual examinations.

Therefore, we present a large VCE dataset, called *Kvasir-Capsule*, consisting of 117 videos with 4,741,504 frames and 14 classes of findings. The dataset contains labelled images and their corresponding full videos, and also unlabelled videos. Recent work in the machine learning community has shown significant improvements regarding sparsely labelled and unlabelled data value. Semi-supervised learning algorithms are successfully applied in different medical image analyses[40,41] using self-learning[42,43] and neural graph learning[44]. Finally, we provide a baseline analysis and outline possible future research directions using *Kvasir-Capsule*.

Methods

The VCE videos were collected from consecutive clinical examinations performed at the Department of Medicine, Bærum Hospital, Vestre Viken Hospital Trust in Norway, which provides health care services to 490,000 people, of which about 200,000 are covered by Bærum Hospital. The examinations were conducted between February 2016 and January 2018 using the Olympus Endocapsule 10 System[45] including the Olympus EC-S10 endocapsule (Fig. 1a) and the Olympus RE-10 endocapsule recorder (Fig. 1b). Originally, the videos were captured at a rate of 2 frames per second, in a resolution of 336×336, and encoded using H.264 (MPEG-4 AVC, part 10). The videos were exported in AVI format using the Olympus system's export tool packaged and encapsulated in the same H.264 format, i.e., the frame formats are the same, but the frame rate specification is changed to 30 fps by the export tool.

Initially, a trained clinician analysed all videos using the Olympus software, selecting thumbnails from lesions and normal findings as part of their clinical work. In spring 2019, all the 117 anonymous videos and thumbnails were exported from a stand-alone workstation using the Olympus software. The Olympus video capsule system has user-friendly functionalities like Omni-selected Mode, skipping images that overlap with previous ones.

All metadata were removed and files renamed with randomly generated file names, before exporting the videos and thumbnails that were shared. Thus, data in the dataset are fully anonymized, as approved by Privacy Data Protection Authority and in accordance with relevant guidelines and regulations of the Regional Committee for Medical and Health Research Ethics - South East Norway. The data has not been pre-processed or augmented in any way apart from this. Subsequently, for clinical analyses of the videos, a central expert reader selected and categorized thumbnails with pathological findings. These thumbnails were traced to their corresponding video segments and the videos were uploaded to a video annotation platform (provided by Augere Medical AS, Norway) for efficient viewing and labelling. Next, three master students labelled and marked the findings with bounding boxes for each frame. The bounding boxes were designed to include the entire lesion and as little as possible of the surrounding mucosa. If the students were unsure about the labelling, the expert reader verified the frames. All labels regarding anatomical structures and normal clean mucosa were then confirmed by one junior

A.5. Paper V - Kvasir-Capsule, a Video Capsule Endoscopy Dataset

(a) Olympus EC-S10 endocapsule   **(b)** Olympus RE-10 endocapsule recorder

Fig. 1 VCE equipment used for data collection.

Data Record	# Files
Labelled images	47,238
Labelled videos	43
Unlabelled images	4,694,266
Unlabelled videos	74

Table 2. Overview of the data records in the *Kvasir-Capsule* dataset.

medical doctor and the expert reader. Finally, all the annotations were once more verified by the expert reader and subsequently validated by a second expert reader. If the second reviewer disagreed with the annotations, the first reviewer reassessed the images to see whether he then agreed with the second reviewer to get an agreement. After the validation process by the second reviewer there was a disagreement on twenty-six findings in seven examinations; nineteen concerning erroneous terminology of the class lymphoid hyperplasia which was changed to lymphangiectasia. The other seven were related to the interpretation of the finding. After reviewing these findings, the first reviewer agreed with the second one to finally reach a perfect agreement. After this procedure, the video frames were exported as images. Hence, a total of four medical persons have selected, analysed and verified the data, and a total of 47,238 frames are labelled.

The Norwegian Privacy Data Protection Authority approved the export of anonymous images for the creation of the database, without consent from participants. It was exempted from approval from the Regional Committee for Medical and Health Research Ethics - South East Norway. Since the data is anonymised and all metadata removed, the dataset is publicly shareable based on Norwegian and General Data Protection Regulation (GDPR) laws.

Data Records

The *Kvasir-Capsule* dataset is available from the Open Science Framework (OSF)[46]. Table 2 gives an overview of all data records in the dataset. In total, the dataset consists of 4,741,621 main data records, i.e., 47,238 images with labels and bounding box masks, the 43 corresponding labelled videos (the videos from which the images are extracted), and 74 unlabelled videos (from which labelled images have not been extracted). 4,694,266 unlabelled images can further be extracted from all the videos combined. All the various labelled classes are shown in Fig. 2. The dataset has a total size of circa 89 GB. Note that the unlabelled images are not extracted and included in the uploaded data due to unnecessary duplication of data, but can easily be extracted from the videos.

The dataset is stored according to the data records listed above, and described in more detail below. We have a "labelled images" catalogue which contains archive files of each labelled class of images. We have a "labelled videos" catalogue which contains all the videos where we have annotated findings from, and an "unlabelled videos" catalogue containing the videos that are not annotated.

Labelled images. In total, the dataset contains 47,238 labelled images stored using the PNG format, where Fig. 3 shows the 14 different classes representing the labelled images and the number of images in each class. The provided *metadata.csv* comma-separated value (CSV) file gives the mapping between file name, the labelling for the image, the corresponding video, and the video frame number. Moreover, the CSV file gives information about the bounding box outlining the finding. Some samples are given in Fig. 4 where the first line gives the type of each element in the lines below. This means that the file *filename* of the labelled image which is the frame *frame_number* extracted from the *video_id* video. Moreover, the finding is from the category *finding_category* and class *finding_class*. Finally, the four x_i, y_i pairs are the four pixel coordinates for the bounding box, e.g., in the first three lines they are empty, meaning that there is no finding with a bounding box in this labelled image. There is one line in the file per each labelled image.
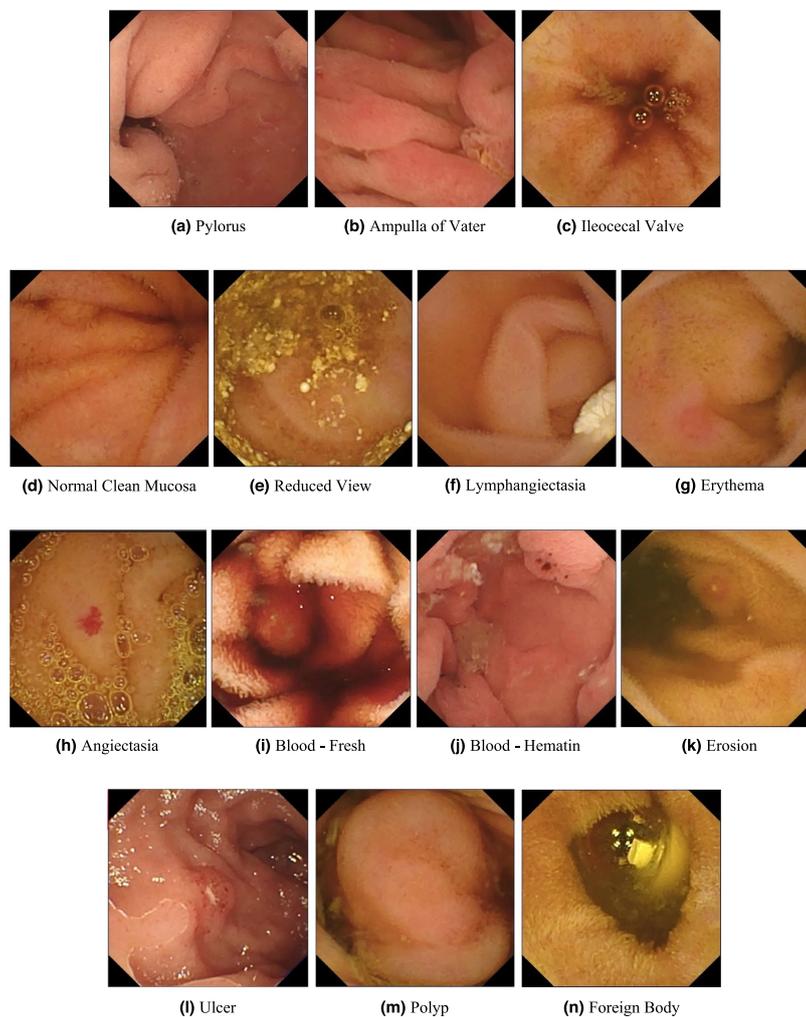
Appendix A. Published Articles

(a) Pylorus **(b)** Ampulla of Vater **(c)** Ileocecal Valve

(d) Normal Clean Mucosa **(e)** Reduced View **(f)** Lymphangiectasia **(g)** Erythema

(h) Angiectasia **(i)** Blood - Fresh **(j)** Blood - Hematin **(k)** Erosion

(l) Ulcer **(m)** Polyp **(n)** Foreign Body

Fig. 2 Image examples of the various labelled classes for images. Images (**a**) to (**c**) are from the *Anatomy* category, and images (**d**) to (**n**) are from the *Luminal findings category*.

We defined two main categories of findings, namely anatomy and luminal findings. Each category, their classes and belonging images are stored in their corresponding folder. As observed in Fig. 3, the number of images per class is not balanced. This is a global challenge in the medical field because some findings are more common than others, which adds a challenge for researchers since methods applied to the data should also be able to learn from a small amount of training data.

Categories of findings. We have organised the dataset in two main categories with their corresponding classes according to the World Endoscopy Association Minimal Standard Terminology version 3.0 (MST 3.0), though we have not included the subcategories or intermediate level to simplify the dataset[47].

Anatomy. The category of *Anatomy* contains anatomical landmarks characterising the GI tract. These landmarks may be used for orientation during endoscopic procedures. However, for small bowel VCE their role is to verify the passage of the capsule trough the entire small bowel to confirm a complete examination. We have labelled three anatomical landmarks, the first two delineate the upper (proximal) and lower (distal) end of the
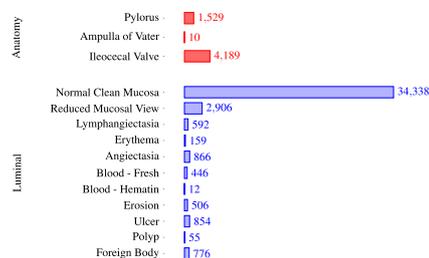
Fig. 3 The number of images in the various Kvasir-Capsule labelled image classes.

```
filename;video_id;frame_number;finding_category;finding_class;x1;y1;x2;y2;x3;y3;x4;y4
...
0728084c8da942d9_22805.jpg;0728084c8da942d9;22805;Luminal;Normal clean mucosa;;;;;;;;
0728084c8da942d9_22806.jpg;0728084c8da942d9;22806;Luminal;Normal clean mucosa;;;;;;;;
0728084c8da942d9_22807.jpg;0728084c8da942d9;22807;Luminal;Normal clean mucosa;;;;;;;;
...
0728084c8da942d9_28789.jpg;0728084c8da942d9;28789;Luminal;Erosion;195;226;244;226;244;265;195;265
0728084c8da942d9_28798.jpg;0728084c8da942d9;28798;Luminal;Erosion;183;212;213;212;213;265;183;265
0728084c8da942d9_28799.jpg;0728084c8da942d9;28799;Luminal;Erosion;197;213;229;213;229;267;197;267
...
```

Fig. 4 Samples from the *metadata.csv* CSV file.

small bowel, respectively. The **pylorus** is the anatomical junction between the stomach and small bowel and is a sphincter (circular muscle) regulating the emptying of the stomach into the duodenum. The **ileocecal valve** marks the transition from the small bowel to the large bowel and is a valve preventing reflux of colonic contents, stool, back into the small bowel. The third one, the **ampulla of Vater**, is the junction between the duodenum and the gall duct.

Luminal findings. Endoscopic examinations may detect various *luminal findings*, this include the subcategories content of the bowel lumen, the aspect of the mucosa and mucosal lesions (pathological findings) that could be either flat, elevated or excavated. These subcategories are not shown in the dataset. Normally, the small bowel contains only a certain amount of yellow or brown liquid considered as clean mucosa. However, larger amounts of content may preclude a complete visualisation of the mucosa crucial to verify normal mucosa and detection of all pathological(abnormal) findings. For the lumen content assessment, we have labelled five classes. **Normal clean mucosa** depicts clean small bowel with no or small amount of fluid and mucosa with healthy villi and no pathological findings. This class can also double as a "normal" class versus the pathological luminal finding class (see below). The class **reduced mucosal view** shows small bowel content reducing the view of the mucosa, like stool or bubbles. However, lesions in the upper GI tract or small bowel may bleed, causing the appearance of **blood - fresh** colouring the liquid red. In cases with minimal bleeding, one may observe small black stripes called **blood - hematin** on the mucosal surface. The **foreign body** class include tablet residue or retained capsules which can also be observed in the lumen.

Abnormalities, called lesions or pathological findings, in the small bowel can be seen as changes to the mucosal surface. Typical mucosal changes sometimes cover larger segments, such as a reddish appearance called erythematous mucosa, is labelled as **erythema**. The mucosal wall can also have different focal lesions. The classes of lesions represented in the *Kvasir-Capsule* dataset are **angiectasia**s; small superficial dilated vessels causing chronic bleeding and subsequently anaemia. It mostly occurs in people with chronic heart and lung diseases[48]. Excavated lesions erode to different extents the surface of the mucosa. Most common are **erosion**s, covered by a tiny fibrin layer, while larger erosions are called **ulcer**s. As an example, Crohn's disease is a chronic inflammation of the small bowel characterised by ulcers and erosions of the mucosa. It may cause strictures of the lumen, making the absorption and passage of nutrients difficult[49]. **Lymphangiectasia**, which represents dilated lymphoid vessels in the mucosal wall, and **polyp**s, which may be precancerous lesions, are visible as protruding from the mucosal wall.

Labelled videos. Labelled videos are the full 43 videos from which we extracted the above mentioned labelled image classes. In total, these videos correspond to approximately 19 hours of video and 47,238 labelled video frames. Several segments of each video was labelled, and these segments are what was exported as the labelled images. As previously mentioned, one can find the frame number and video of origin of each extracted image in the CSV-file. Even though we already have extracted the most interesting frames (images) found by the clinicians from these videos, they do contain 1,932,047 non-labelled frames that could be interesting in future research. One could also extract the video sequences around the various findings.

Appendix A. Published Articles

Method		Macro average			Micro average			
		Precision	Recall	F1-score	Precision	Recall	F1-score	MCC
Normal CEL	DensNet-161 (fold 0)	0.2165	0.2341	0.1923	0.7375	0.7375	0.7375	0.3707
	DensNet-161 (fold 1)	0.3493	0.3158	0.2996	0.7327	0.7327	0.7327	0.4604
	Avereage	0.2829	0.2749	0.2459	0.7351	0.7351	0.7351	0.4156
	ResNet-152 (fold 0)	0.3302	0.2401	0.1970	0.7203	0.7203	0.7203	0.3520
	ResNet-152 (fold 1)	0.3431	0.2805	0.2789	0.7481	0.7481	0.7481	0.4718
	Average	**0.3367**	0.2603	0.2379	0.7342	0.7342	0.7342	0.4119
Weighted CEL	DensNet-161 (fold 0)	0.2933	0.2939	0.2523	0.7195	0.7195	0.7195	0.3998
	DensNet-161 (fold 1)	0.3163	0.2914	0.2581	0.6991	0.6991	0.6991	0.4054
	Average	0.3048	**0.2927**	0.2552	0.7093	0.7093	0.7093	0.4026
	ResNet-152 (fold 0)	0.2136	0.2872	0.2186	0.6568	0.6568	0.6568	0.3588
	ResNet-152 (fold 1)	0.3033	0.2799	0.2478	0.6890	0.6890	0.6890	0.3966
	Average	0.2585	0.2836	0.2332	0.6729	0.6729	0.6729	0.3777
Weighted sampling	DensNet-161 (fold 0)	0.2525	0.2794	0.2315	0.7332	0.7332	0.7332	0.4111
	DensNet-161 (fold 1)	0.3463	0.2830	0.2806	0.7400	0.7400	0.7400	0.4547
	Average	0.2994	0.2812	**0.2560**	**0.7366**	**0.7366**	**0.7366**	**0.4329**
	ResNet-152 (fold 0)	0.2637	0.2930	0.2334	0.7324	0.7324	0.7324	0.4088
	ResNet-152 (fold 1)	0.3088	0.2619	0.2417	0.7316	0.7316	0.7316	0.4520
	Average	0.2862	0.2774	0.2375	0.7320	0.7320	0.7320	0.4304

Table 3. Results for all classification experiments. Experiments were done with and without weighted cross-entropy loss (CEL) and using a weighted sampling technique. Bold numbers represent the best average value of that column.

Unlabelled videos. We also provide 74 videos, which contain approximately 25 hours of video and 2,762,219 video frames, without any labels. As previously mentioned, unlabelled data can still have great value. Sparsely labelled or unlabelled data can be important for recently emerging semi-supervised learning algorithms. These videos are of the same format and quality as the labelled videos, except we do not provide any annotations. This means that users of the dataset can either use medical experts to provide further labels, or use the data in unsupervised or semi-supervised learning approaches.

Technical Validation

To evaluate the technical quality of *Kvasir-Capsule*, we performed a series of classification experiments. We trained two CNN-based classifiers to classify the labelled data. Both architectures have previously shown excellent performance on classifying GI-related imagery from traditional colonoscopies[50,51], and should be a good benchmark for VCE-related data. The two algorithms are based on standard CNN architectures, namely DenseNet-161[52] and ResNet-152[53]. All experiments were performed over two-fold cross-validation using categorical cross-entropy loss with and without class weighting. We also used weighted sampling, which balances the dataset by removing and adding images for each class based on a given set of weights. To ensure a fair and robust evaluation, no video is shared between splits. Thus, the frames used for training were independent from the frames used for validation. This also means that there are frames depicting the same finding in each split which then are related to each other, but no related frames distributed across the splits. The effect should therefore be similar to traditional data augmentation techniques used by many researchers today such as multiple rotations, angles and crops.

The purpose of these experiments is two-fold. First, we create a baseline for future researchers using the *Kvasir-Capsule* dataset. Second, by using an algorithm that has previously shown good results on classifying GI images, we evaluate how challenging the task of categorizing VCE-related data is. Note that for the classification experiments, we removed the blood - hematin, ampulla of Vater, and polyp classes due to the small number of findings. The results for the two classification algorithms are shown in Table 3 and confusion matrices for the best average MCC value in Fig. 5. We estimated micro-averaged and macro-averaged values for precision, recall and F1-score for each method. The Matthews correlation coefficient (MCC) was calculated using the multi-class generalization, also called the R_K. In short, if TP, TN, FP, and FN are the true positives, true negatives, false positives, and false negatives, respectively, these metrics are defined as follows[26]:

Precision. This metric is also frequently called the *positive predictive value*, and shows the ratio of samples that are correctly identified as positive among the returned samples (the fraction of retrieved samples that are relevant):

$$precision = \frac{TP}{\# \; of \; all \; returned \; samples} = \frac{TP}{TP + FP}$$

Recall. This metric is also frequently called *sensitivity*, *probability of detection* and *true positive rate*, and it is the ratio of samples that are correctly identified as positive among all existing positive samples:

$$recall = \frac{TP}{\# \; of \; all \; positives} = \frac{TP}{TP + FN}$$

(a) Confusion matrix for model evaluated on split 0.

(b) Confusion matrix for model evaluated on split 0.

Fig. 5 Confusion matrices for the best average MCC value which is from the weighted sampling technique. The labeling of the classes is as follows: (A) Angiectasia; (B) Blood - fresh; (C) Erosion; (D) Erythema; (E) Foreign Body; (F) Ileocecal valve; (G) Lymphangiectasia; (H) Normal clean mucosa; (I) Pylorus; (J) Reduced Mucosal View; (K) Ulcer.

F1 score (F1). A measure of a test's accuracy by calculating the harmonic mean of the precision and recall:

$$F1\ score = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2TP}{2TP + FP + FN}$$

Matthews correlation coefficient (MCC). MCC takes into account true and false positives and negatives, and is a balanced measure even if the classes are of very different sizes. For the multiclass classification generalization, it is often called the R_k statistic. In following equation, t_k is the number of times class k actually occurred, p_k is the number of times class k was predicted, c is the total number of samples correctly predicted, and s is the total number of samples:

$$MCC = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}}$$

The micro and macro averages are different ways to average metrics calculated over multiple classes. The macro average is the arithmetic mean of all the scores of different classes, i.e., calculates the metric per class and then calculates the average of these over the number of classes. For example, it is defined for precision as the sum of precision scores for all classes ($precicion_1 + \ldots + precicion_n$) divided by the number of classes (n). The micro average is not counting class wise first, but looking at the total number of true and false findings. For example, for precision, it is defined as sum of true positives ($TP_1 + \ldots + TP_n$) for all the n classes divided by the all returned positive predictions ($TP_1 + FP_1 + \ldots + TP_n + FP_n$).

Considering the results, we experience that classifying VCE data is quite a challenging task. For example, several of the classes are erroneously predicted as **Normal clean mucosa**. On the other hand, the class with the most accurate predictions is also **Normal clean mucosa**, reaching 85% in fold one and 91% in fold two. This is expected as the class comprise approximately 73% of the labelled images. This points out the challenges of making reliable systems as there are multiple aspects to consider, e.g., the resolution of VCE frames are lower compared to gastro- or colonoscopies, and many of the findings are subtle where even clinicians have difficulties differentiating between the classes. As noticed when comparing the images in Fig. 2, several findings are hard to see and easily mixed. For example, erosions can often be mistaken as small residues, and it can be difficult to differentiate normal mucosa from slight erythema. Thus, these results show the potential of AI-based analysis, but also further motivates the need to publish this dataset for more investigations and research into better specific algorithms for VCE data. The code used to conduct all experiments, produce all plots, and the images contained in each split are available on GitHub (https://github.com/simula/kvasir-capsule), i.e., to increase reproducibility and facilitate researches to perform comparable experiments on the *Kvasir-Capsule* dataset.

Usage Notes

To the best of our knowledge, we have collected the largest and most diverse public available VCE dataset. *Kvasir-Capsule* is made available to enable researchers to develop detection or classification methods of various GI findings using for example computer vision and machine learning approaches. As the labelled findings also include bounding boxes, areas of potential use are analysis, classification, segmentation, and retrieval of images and videos of particular findings or properties. Moreover, the ground truths of various findings by the expert gastroenterologists provide a unique and diverse learning set for future clinicians, i.e., the labelled data can be used for teaching and training in medical education.

Appendix A. Published Articles

The unlabelled data is well suited for semi-supervised and unsupervised machine learning methods, and, if even more ground truth data is needed, the users of the data can have medical experts provide the needed labels. In this respect, recent work has shown remarkable improvements in the area of semi-supervised learning, also successfully applied in medical image analyses[40]. Instead of learning from a large set of annotated data, algorithms learn from sparsely labelled and unlabelled data. Self-learning[42,43] and neural graph learning[44] are both examples using unlabelled data in addition to a small amount of labelled data to extract additional information[41–43]. In an area with scarce data, these new algorithms might be the technology needed to make AI truly useful for medical applications.

An important note in general for this type of AI-based detection systems is that one should be careful about how the dataset is split into for example training and test sets in order to avoid having related frames in several of the sets. This will give an unfair effect on the results. Thus, the splits should be completely different, probably even at the level of patients. As described below, an example of such a split in found in our GitHub repository (see below in the Code Availability section).

Currently, there is substantial research in GI image and video analysis. We welcome future contributions such as using the dataset for comparisons and reproducibility of experiments and further encourage publishing and sharing of new data. *Kvasir-Capsule* is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original authors and the source.

Code availability

In addition to releasing the data, we also publish code used for the baseline experiments. All code and additional data required for the experiments, including our splits into training and test datasets, are available on GitHub via http://www.github.com/simula/kvasir-capsule.

References

1. Greenwood-Van Meerveld, B., Johnson, A. C. & Grundy, D. Gastrointestinal physiology and function. In *Gastrointestinal Pharmacology*, 1–16 (Springer, 2017).
2. McLaughlin, P. D. & Maher, M. M. Primary malignant diseases of the small intestine. *American Journal of Roentgenology* **201**, W9–W14 (2013).
3. Thomson, A. *et al.* Small bowel review: diseases of the small intestine. *Digestive diseases and sciences* **46**, 2555–2566 (2001).
4. Enns, R. A. *et al.* Clinical practice guidelines for the use of video capsule endoscopy. *Gastroenterology* **152**, 497–514 (2017).
5. Costamagna, G. *et al.* A prospective trial comparing small bowel radiographs and video capsule endoscopy for suspected small bowel disease. *Gastroenterology* **123**, 999–1005 (2002).
6. Hewett, D. G., Kahi, C. J. & Rex, D. K. Efficacy and effectiveness of colonoscopy: how do we bridge the gap? *Gastrointestinal Endoscopy Clinics* **20**, 673–684 (2010).
7. Lee, S. H. *et al.* Endoscopic experience improves interobserver agreement in the grading of esophagitis by los angeles classification: conventional endoscopy and optimal band image system. *Gut and liver* **8**, 154 (2014).
8. Van Doorn, S. C. *et al.* Polyp morphology: an interobserver evaluation for the paris classification among international experts. *The American Journal of Gastroenterology* **110**, 180–187 (2015).
9. Kaminski, M. F. *et al.* Quality indicators for colonoscopy and the risk of interval cancer. *New England Journal of Medicine* **362**, 1795–1803 (2010).
10. Zheng, Y., Hawkins, L., Wolff, J., Goloubeva, O. & Goldberg, E. Detection of lesions during capsule endoscopy: physician performance is disappointing. *American Journal of Gastroenterology* **107**, 554–560 (2012).
11. Chetcuti, S. Z. & Sidhu, R. Capsule endoscopy-recent developments and future directions. *Expert review of gastroenterology & hepatology* **15**, 127–137 (2021).
12. Rondonotti, E. *et al.* Can we improve the detection rate and interobserver agreement in capsule endoscopy? *Digestive and Liver Disease* **44**, 1006–1011 (2012).
13. Cave, D. R., Hakimian, S. & Patel, K. Current controversies concerning capsule endoscopy. *Digestive Diseases and Sciences* **64**, 3040–3047 (2019).
14. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine* **25**, 44 (2019).
15. Riegler, M. *et al.* Multimedia and medicine: Teammates for better disease detection and survival. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 968–977 (2016).
16. Riegler, M. *et al.* EIR - efficient computer aided diagnosis framework for gastrointestinal endoscopies. In *Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing (CBMI)*, 1–6 (2016).
17. Alammari, A. *et al.* Classification of ulcerative colitis severity in colonoscopy videos using cnn. In *Proceedings of the ACM International Conference on Information Management and Engineering (ICIME)*, 139–144 (2017).
18. Wang, Y., Tavanapong, W., Wong, J., Oh, J. H. & De Groen, P. C. Polyp-alert: Near real-time feedback during colonoscopy. *Computer Methods and Programs in Biomedicine* **120**, 164–179 (2015).
19. Hirasawa, T., Aoyama, K., Fujisaki, J. & Tada, T. 113 application of artificial intelligence using convolutional neural network for detecting gastric cancer in endoscopic images. *Gastrointestinal Endoscopy* **87**, AB51 (2018).
20. Wang, L., Xie, C. & Hu, Y. Iddf2018-abs-0260 deep learning for polyp segmentation. *BMJ Publishing Group* (2018).
21. Mori, Y. *et al.* Real-time use of artificial intelligence in identification of diminutive polyps during colonoscopy: a prospective study. *Annals of internal medicine* **169**, 357–366 (2018).
22. Bychkov, D. *et al.* Deep learning based tissue analysis predicts outcome in colorectal cancer. *Scientific Reports* **8**, 1–11 (2018).
23. Min, M. *et al.* Computer-aided diagnosis of colorectal polyps using linked color imaging colonoscopy to predict histology. *Scientific reports* **9**, 2881 (2019).
24. Bernal, J. & Aymeric, H. Miccai endoscopic vision challenge polyp detection and segmentation. *Web-page of the 2017 Endoscopic Vision Challenge*, https://endovissub2017-giana.grand-challenge.org/home/ (2017).
25. Tajbakhsh, N., Gurudu, S. R. & Liang, J. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging* **35**, 630–644 (2016).
26. Pogorelov, K. *et al.* Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the ACM on Multimedia Systems Conference (MMSYS)*, 164–169 (2017).
27. Borgli, H. *et al.* Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Scientific Data* **7**, 1–14 (2020).

28. Yuan, Y. & Meng, M. Q.-H. A novel feature for polyp detection in wireless capsule endoscopy images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5010–5015 (2014).
29. Yuan, Y. & Meng, M. Q.-H. Deep learning for polyp recognition in wireless capsule endoscopy images. *Medical Physics* **44**, 1379–1389 (2017).
30. Karargyris, A. & Bourbakis, N. G. Detection of small bowel polyps and ulcers in wireless capsule endoscopy videos. *IEEE Transactions on Biomedical Engineering* **58**, 2777–2786 (2011).
31. Leenhardt, R. *et al.* A neural network algorithm for detection of gi angiectasia during small-bowel capsule endoscopy. *Gastrointestinal endoscopy* **89 1**, 189–194 (2019).
32. Pogorelov, K. *et al.* Deep learning and handcrafted feature based approaches for automatic detection of angiectasia. In *Proceedings of IEEE Conference on Biomedical and Health Informatics (BHI)*, 365–368 (2018).
33. Pogorelov, K. *et al.* Bleeding detection in wireless capsule endoscopy videos—color versus texture features. *Journal of applied clinical medical physics* **20** (2019).
34. Rahim, T., Usman, M. A. & Shin, S. Y. A survey on contemporary computer-aided tumor, polyp, and ulcer detection methods in wireless capsule endoscopy imaging. *Computerized Medical Imaging and Graphics* **85**, 101767 (2020).
35. Soffer, S. *et al.* Deep learning for wireless capsule endoscopy: a systematic review and meta-analysis. *Gastrointestinal Endoscopy* (2020).
36. Yang, Y. J. The future of capsule endoscopy: The role of artificial intelligence and other technical advancements. *Clinical Endoscopy* **53**, 387 (2020).
37. Park, J. *et al.* Recent development of computer vision technology to improve capsule endoscopy. *Clinical endoscopy* **52**, 328 (2019).
38. Iakovidis, D. K. & Koulaouzidis, A. Software for enhanced video capsule endoscopy: challenges for essential progress. *Nature Reviews Gastroenterology & Hepatology* **12**, 172–186 (2015).
39. Jani, K. K. & Srivastava, R. A survey on medical image analysis in capsule endoscopy. *Current Medical Imaging* **15**, 622–636 (2019).
40. Cheplygina, V., de Bruijne, M. & Pluim, J. P. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis* **54**, 280–296 (2019).
41. He, K., Fan, H., Wu, Y., Xie, S. & Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738 (2020).
42. Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, 4182–4192 (PMLR, 2020).
43. Misra, I. & Maaten, L. V. D. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6707–6717 (2020).
44. Bui, T. D., Ravi, S. & Ramavajjala, V. Neural graph learning: Training neural networks using graphs. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 64–71 (2018).
45. Olympus. The endocapsule 10 system. *Olympus homepage*, https://www.olympus-europa.com/medical/en/Products-and-Solutions/Products/Product/ENDOCAPSULE-10-System.html (2013).
46. Thambawita, V. *et al.* The kvasir-capsule dataset. *Open Science Framework* https://doi.org/10.17605/OSF.IO/DV2AG (2020).
47. Aabakken, L. *et al.* Standardized endoscopic reporting. *Journal of Gastroenterology and Hepatology* **29**, 234–240 (2014).
48. Chetcuti Zammit, S. *et al.* Overview of small bowel angioectasias: clinical presentation and treatment options. *Expert review of gastroenterology & hepatology* **12**, 125–139 (2018).
49. Gomollón, F. *et al.* 3rd european evidence-based consensus on the diagnosis and management of crohn's disease 2016: part 1: diagnosis and medical management. *Journal of Crohn's and Colitis* **11**, 3–25 (2017).
50. Thambawita, V. *et al.* An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification. *ACM Transactions on Computing for Healthcare* **1**, 1–29 (2020).
51. Thambawita, V. *et al.* The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. In *Proceedings of the MediaEval 2018 Workshop* (2018).
52. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269 (2017).
53. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778 (2016).
54. Koulaouzidis, A. *et al.* Kid project: an internet-based digital video atlas of capsule endoscopy for research purposes. *Endoscopy international open* **5**, E477–E483 (2017).
55. Bernal, J. & Aymeric, H. Gastrointestinal Image ANAlysis (GIANA) Angiodysplasia D&L challenge. *Web-page of the 2017 Endoscopic Vision Challenge*, https://endovissub2017-giana.grand-challenge.org/home/ (2017).
56. Angermann, Q. *et al.* Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis. In *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures*, 29–41 (Springer, 2017).
57. Bernal, J. *et al.* Polyp detection benchmark in colonoscopy videos using gtcreator: A novel fully configurable tool for easy and fast annotation of image databases. In *Proceedings of 32nd CARS conference* (2018).
58. Computer-assisted diagnosis for capsule endoscopy (cad-cap) database. *The 2019 GIANA Grand Challenge web-page*, https://giana.grand-challenge.org/WCE/ (2019).
59. Leenhardt, R. *et al.* Cad-cap: a 25,000-image database serving the development of artificial intelligence for capsule endoscopy. *Endoscopy international open* **8**, E415 (2020).
60. Gastrolab. *The Gastrointestinal Site*, http://www.gastrolab.net/index.htm (1996).

Acknowledgements

Author contributions

S.A.H., V.T., P.H., M.A.R., P.H.S. and T.d.L. conceived the experiment(s), S.A.H. and V.T. conducted the experiment(s), P.H.S., H.G., O.O.N., E.N., V.T., S.A.H., M.A.R., P.H. and T.d.L. prepared and cleaned the data for publication, and all authors analysed the results and reviewed the manuscript.

Competing interests

Authors P.H.S., T.J.D.B., H.E., A.P., D.J., T.d.L., M.A.R., and P.H. all own shares in the Augere Medical AS company developing AI solutions for colonoscopies. The Augere video annotation system was used to label the data. There is no commercial interest from Augere regarding this publication and dataset. Otherwise, the authors declare no competing interests.

177

Appendix A. Published Articles

Additional information

Correspondence and requests for materials should be addressed to P.H.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

178

A.6 Paper VI - HTAD: A Home-Tasks Activities Dataset with Wrist-Accelerometer and Audio Features

Authors: Enrique Garcia-Ceja, **Vajira Thambawita**, Steven A. Hicks, Debesh Jha, Petter Jakobsen, Hugo L. Hammer, Pål Halvorsen, Michael A. Riegler

Abstract: In this paper, we present HTAD: A Home Tasks Activities Dataset. The dataset contains wrist-accelerometer and audio data from people performing at-home tasks such as sweeping, brushing teeth, washing hands, or watching TV. These activities represent a subset of activities that are needed to be able to live independently. Being able to detect activities with wearable devices in real-time is important for the realization of assistive technologies with applications in different domains such as elderly care and mental health monitoring. Preliminary results show that using machine learning with the presented dataset leads to promising results, but also there is still improvement potential. By making this dataset public, researchers can test different machine learning algorithms for activity recognition, especially, sensor data fusion methods.

Published: MultiMedia Modeling (MMM), 2021

Candidate contributions: Vajira contributed to organizing data and publishing the dataset in the public data repository called osf.io. He created the wiki page of this dataset and published it to users as a reference manual to the dataset. He contributed to drafting and revising the paper.

Thesis objectives: Sub-objective I, Sub-objective II

HTAD: A Home-Tasks Activities Dataset with Wrist-Accelerometer and Audio Features

Enrique Garcia-Ceja[1( )], Vajira Thambawita[2,3], Steven A. Hicks[2,3],
Debesh Jha[2,4], Petter Jakobsen[5], Hugo L. Hammer[3], Pål Halvorsen[2],
and Michael A. Riegler[2]

[1] SINTEF Digital, Oslo, Norway
`enrique.garcia-ceja@sintef.no`
[2] SimulaMet, Oslo, Norway
`michael@simula.no`
[3] Oslo Metropolitan University, Oslo, Norway
[4] UIT The Arctic University of Norway, Tromsø, Norway
[5] Haukeland University Hospital, Bergen, Norway

Abstract. In this paper, we present HTAD: A Home Tasks Activities Dataset. The dataset contains wrist-accelerometer and audio data from people performing at-home tasks such as sweeping, brushing teeth, washing hands, or watching TV. These activities represent a subset of activities that are needed to be able to live independently. Being able to detect activities with wearable devices in real-time is important for the realization of assistive technologies with applications in different domains such as elderly care and mental health monitoring. Preliminary results show that using machine learning with the presented dataset leads to promising results, but also there is still improvement potential. By making this dataset public, researchers can test different machine learning algorithms for activity recognition, especially, sensor data fusion methods.

Keywords: Activity recognition · Dataset · Accelerometer · Audio · Sensor fusion

1 Introduction

Automatic monitoring of human physical activities has become of great interest in the last years since it provides contextual and behavioral information about a user without explicit user feedback. Being able to automatically detect human activities in a continuous unobtrusive manner is of special interest for applications in sports [16], recommendation systems, and elderly care, to name a few. For example, appropriate music playlists can be recommended based on the user's current activity (exercising, working, studying, etc.) [21]. Elderly people at an early stage of dementia could also benefit from these systems, like by monitoring their hygiene-related activities (showering, washing hands, or brush

teeth) and sending reminder messages when appropriate [19]. Human activity recognition (HAR) also has the potential for mental health care applications [11] since it can be used to detect sedentary behaviors [4], and it has been shown that there is an important association between depression and sedentarism [5]. Recently, the use of wearable sensors has become the most common approach to recognizing physical activities because of its unobtrusiveness and ubiquity, specifically, the use of accelerometers [9,15,17], because they are already embedded in several commonly used devices like smartphones, smart-watches, fitness bracelets, etc.

In this paper, we present HTAD: a Home Tasks Activities Dataset. The dataset was collected using a wrist accelerometer and audio recordings. The dataset contains data for common home tasks activities like *sweeping, brushing teeth, watching TV, washing hands*, etc. To protect users' privacy, we only include audio data after feature extraction. For accelerometer data, we include the raw data and the extracted features.

There are already several related datasets in the literature. For example, the epic-kitchens dataset includes several hours of first-person videos of activities performed in kitchens [6]. Another dataset, presented by Bruno et al., has 14 activities of daily living collected with a wrist-worn accelerometer [3]. Despite the fact that there are many activity datasets, it is still difficult to find one with both: wrist-acceleration and audio. The authors in [20] developed an application capable of collecting and labeling data from smartphones and wrist-watches. Their app can collect data from several sensors, including inertial and audio. The authors released a dataset[1] that includes 2 participants and point to another website (http://extrasensory.ucsd.edu) that contains data from 60 participants. However, the link to the website was not working at the present date (August-10-2020). Even though the present dataset was collected by 3 volunteers, and thus, is a small one compared to others, we think that it is useful for the activity recognition community and other researchers interested in wearable sensor data processing. The dataset can be used for machine learning classification problems, especially those that involve the fusion of different modalities such as sensor and audio data. This dataset can be used to test data fusion methods [13] and used as a starting point towards detecting more types of activities in home settings. Furthermore, the dataset can potentially be combined with other public datasets to test the effect of using heterogeneous types of devices and sensors.

This paper is organized as following: In Sect. 2, we describe the data collection process. Section 3 details the feature extraction process, both, for accelerometer and audio data. In Sect. 4, the structure of the dataset is explained. Section 5 presents baseline experiments with the dataset, and finally in Sect. 6, we present the conclusions.

2   Dataset Details

The dataset can be downloaded via: https://osf.io/4dnh8/.

---

[1] https://www.kaggle.com/yvaizman/the-extrasensory-dataset.

The home-tasks data were collected by 3 individuals. They were 1 female and 2 males with ages ranging from 25 to 30. The subjects were asked to perform 7 scripted home-task activities including: *mop floor*, *sweep floor*, *type on computer keyboard*, *brush teeth*, *wash hands*, *eat chips* and *watch TV*. The *eat chips* activity was conducted with a bag of chips. Each individual performed each activity for approximately 3 min. If the activity lasted less than 3 min, an additional trial was conducted until the 3 min were completed. The volunteers used a wrist-band (Microsoft Band 2) and a smartphone (Sony XPERIA) to collect the data.

The subjects wore the wrist-band in their dominant hand. The accelerometer data was collected using the wrist-band internal accelerometer. Figure 1 shows the actual device used. The inertial sensor captures motion from the x, y, and z axes, and the sampling rate was set to 31 Hz. Moreover, the environmental sound was captured using the microphone of a smartphone. The audio sampling rate was set at 8000 Hz. The smartphone was placed on a table in the same room where the activity was taking place.

An in-house developed app was programmed to collect the data. The app runs on the Android operating system. The user interface consists of a dropdown list from which the subject can select the home-task. The wrist-band transfers the captured sensor data and timestamps over Bluetooth to the smartphone. All the inertial data is stored in a plain text format.

Fig. 1. Wrist-band watch.

3   Feature Extraction

In order to extract the accelerometer and audio features, the original raw signals were divided into non-overlapping 3 s segments. The segments are not overlapped. A three second window was chosen because, according to Banos *et al.* [2], this is a typical value for activity recognition systems. They did comprehensive tests by trying different segments sizes and they concluded that small segments produce better results compared to longer ones. From each segment, a set of features were calculated which are known as *feature vectors* or *instances*. Each *instance* is characterized by the audio and accelerometer features. In the following section, we provide details about how the features were extracted.

3.1 Accelerometer Features

From the inertial sensor readings, 16 measurements were computed including: The *mean*, *standard deviation*, *max* value for all the x, y and z axes, *pearson correlation* among pairs of axes (xy, xz, and yz), *mean magnitude*, *standard deviation of the magnitude*, the *magnitude area under the curve* (AUC, Eq. 1) , and *magnitude mean differences* between consecutive readings (Eq. 2). The *magnitude* of the signal characterizes the overall contribution of acceleration of x, y and z. (Eq. 3). Those features were selected based on previous related works [7,10,23].

$$AUC = \sum_{t=1}^{T} magnitude(t) \tag{1}$$

$$meandif = \frac{1}{T-1} \sum_{t=2}^{T} magnitude(t) - magnitude(t-1) \tag{2}$$

$$Magnitude(x,y,z,t) = \sqrt{a_x(t)^2 + a_y(t)^2 + a_z(t)^2} \tag{3}$$

where $a_x(t)^2$, $a_y(t)^2$ and $a_z(t)^2$ are the squared accelerations at time t.

Figure 2 shows violin plots for three of the accelerometer features: mean of the x-axis, mean of the y-axis, and mean of the z-axis. Here, we can see that overall, the mean acceleration in x was higher for the *brush teeth* and *eat chips* activities. On the other hand, the mean acceleration in the y-axis was higher for the *mop floor* and *sweep* activities.

3.2 Audio Features

The features extracted from the sound source were the Mel Frequency Cepstral Coefficients (MFCCs). These features have been shown to be suitable for activity classification tasks [1,8,12,18]. The 3 s sound signals were further split into 1 s windows. Then, 12 MFCCs were extracted from each of the 1 s windows. In total, each instance has 36 MFCCs. In total, this process resulted in the generation of 1,386 instances. The tuneR R package [14] was used to extract the audio features. Table 1 shows the percentage of instances per class. More or less, all classes are balanced in number.

4 Dataset Structure

The main folder contains directories for each user and a *features.csv* file. Within each users' directory, the accelerometer files can be found (*.txt* files). The file names are comprised of three parts with the following format: *timestamp-acc-label.txt*. *timestamp* is the timestamp in Unix format. *acc* stands for accelerometer and *label* is the activity's label. Each *.txt* file has four columns: timestamp and the acceleration for each of the x, y, and z axes. Figure 3 shows an example
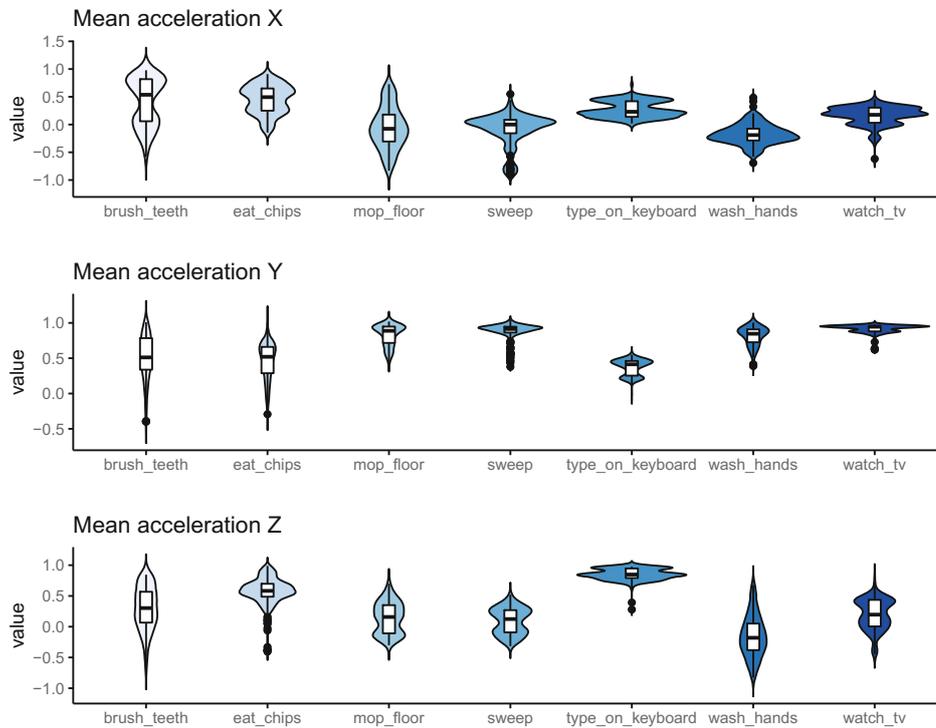
Fig. 2. Violin plots of mean acceleration of the x, y, and z axes.

Table 1. Distribution of activities by class.

Class	Proportion
Brush teeth	12.98%
Eat chips	20.34%
Mop floor	13.05%
Sweep	12.84%
Type on keyboard	12.91%
Wash hands	12.98%
Watch TV	14.90%

of the first rows of one of the files. The *features.csv* file contains the extracted features as described in Sect. 3. It contains 54 columns. *userid* is the user id. *label* represents the activity label and the remaining columns are the features. Columns with a prefix of *v1_* correspond to audio features whereas columns with a prefix of *v2_* correspond to accelerometer features. In total, there are 36 audio features that correspond to the 12 MFCCs for each second, with a total of 3 s and 16 accelerometer features.

```
1468360517664,-0.12915039,0.9797363,-0.21191406
1468360517693,-0.13500977,0.98168945,-0.21118164
1468360517743,-0.1496582,0.9819336,-0.20336914
1468360517763,-0.16894531,0.9892578,-0.21606445
1468360517788,-0.18847656,0.99658203,-0.20581055
1468360517818,-0.1850586,0.97998047,-0.21362305
1468360517857,-0.19140625,0.97216797,-0.21533203
1468360517904,-0.18066406,0.9692383,-0.21411133
1468360517921,-0.1730957,0.9560547,-0.21435547
1468360517962,-0.17871094,0.9626465,-0.2163086
```

Fig. 3. First rows of one of the accelerometer files.

5   Baseline Experiments

In this section, we present a series of baseline experiments that can serve as a starting point to develop more advanced methods and sensor fusion techniques. In total, 3 classification experiments were conducted with the HTAD dataset. For each experiment, different classifiers were employed, including ZeroR (baseline), a J48 tree, Naive Bayes, Support Vector Machine (SVM), a K-nearest neighbors (KNN) classifier with $k = 3$, logistic regression, and a multilayer perceptron. We used the WEKA software [22] version 3.8 to train the classifiers. In each experiment, we used different sets of features. For experiment 1, we trained the models using only *audio features*, that is, the MFCCs. The second experiment consisted of training the models with only the 16 *accelerometer features* described earlier. Finally, in experiment 3, we combined the *audio and accelerometer* features by aggregating them. 10-fold cross-validation was used to train and assess the classifier's performance. The reported performance is the weighted average of different metrics using a one-vs-all approach since this is a multi-class problem.

Table 2. Classification performance (weighted average) with audio features. The best performing classifier was KNN.

Classifier	False-Positive Rate	Precision	Recall	F1-Score	MCC
ZeroR	0.203	0.041	0.203	0.069	0.000
J48	0.065	0.625	0.623	0.624	0.559
Naive Bayes	0.049	0.720	0.714	0.713	0.667
SVM	0.054	0.699	0.686	0.686	0.637
KNN	0.037	0.812	0.788	0.793	0.761
Logistic regression	0.062	0.654	0.652	0.649	0.591
Multilayer perceptron	0.041	0.776	0.769	0.767	0.731

Tables 2, 3 and 4 show the final results. When using only audio features (Table 2), the best performing model was the KNN in terms of all performance

202   E. Garcia-Ceja et al.

Table 3. Classification performance (weighted average) with accelerometer features. The best performing classifier was KNN.

Classifier	False-Positive Rate	Precision	Recall	F1-Score	MCC
ZeroR	0.203	0.041	0.203	0.069	0.000
J48	0.036	0.778	0.780	0.779	0.743
Naive Bayes	0.080	0.452	0.442	0.447	0.365
SVM	0.042	0.743	0.740	0.740	0.698
KNN	0.030	0.820	0.820	0.818	0.790
Logistic regression	0.031	0.800	0.802	0.800	0.769
Multilayer perceptron	0.031	0.815	0.812	0.812	0.782

Table 4. Classification performance (weighted average) when combining all features. The best performing classifier was Multilayer perceptron.

Classifier	False-Positive Rate	Precision	Recall	F1-Score	MCC
ZeroR	0.203	0.041	0.203	0.069	0.000
J48	0.035	0.785	0.785	0.785	0.750
Naive Bayes	0.028	0.826	0.823	0.823	0.796
SVM	0.020	0.876	0.874	0.875	0.855
KNN	0.014	0.917	0.911	0.912	0.899
Logistic regression	0.022	0.859	0.859	0.859	0.837
Multilayer perceptron	0.014	0.915	0.914	0.914	0.901

metrics with a Mathews correlation coefficient (MCC) of 0.761. We report MCC instead of accuracy because MCC is more robust against class distributions. In the case when using only accelerometer features (Table 3), the best model was again KNN in terms of all performance metrics with an MCC of 0.790. From these tables, we observe that most classifiers performed better when using accelerometer features with the exception of Naive Bayes. Next, we trained the models using all features (accelerometer and audio). Table 4 shows the final results. In this case, the best model was the multilayer perceptron followed by KNN. Overall, all models benefited from the combination of features, of which some increased their performance by up to ≈0.15, like the SVM which went from an MCC of 0.698 to 0.855.

All in all, combining data sources provided enhanced performance. Here, we just aggregated the features from both data sources. However, other techniques can be used such as late fusion which consists of training independent models using each data source and then combining the results. Thus, the experiments show that machine learning systems can perform this type of automatic activity detection, but also that there is a large potential for improvements - where the

HTAD dataset can play an important role, not only as an enabling factor, but
also for reproducibility.

6   Conclusions

Reproducibility and comparability of results is an important factor of high-
quality research. In this paper, we presented a dataset in the field of activity
recognition supporting reproducibility in the field. The dataset was collected
using a wrist accelerometer and captured audio from a smartphone. We provided
baseline experiments and showed that combining the two sources of information
produced better results. Nowadays, there exist several datasets, however, most
of them focus on a single data source and on the traditional *walking, jogging,
standing, etc.* activities. Here, we employed two different sources (accelerome-
ter and audio) for home task activities. Our vision is that this dataset will allow
researchers to test different sensor data fusion methods to improve activity recog-
nition performance in home-task settings.

References

1. Al Masum Shaikh, M., Molla, M., Hirose, K.: Automatic life-logging: a novel app-
   roach to sense real-world activities by environmental sound cues and common
   sense. In: 11th International Conference on Computer and Information Technology,
   ICCIT 2008, pp. 294–299, December 2008. https://doi.org/10.1109/ICCITECHN.
   2008.4803018
2. Banos, O., Galvez, J.M., Damas, M., Pomares, H., Rojas, I.: Window size impact
   in human activity recognition. Sensors **14**(4), 6474–6499 (2014). https://doi.org/
   10.3390/s140406474. http://www.mdpi.com/1424-8220/14/4/6474
3. Bruno, B., Mastrogiovanni, F., Sgorbissa, A., Vernazza, T., Zaccaria, R.: Analysis
   of human behavior recognition algorithms based on acceleration data. In: 2013
   IEEE International Conference on Robotics and Automation, pp. 1602–1607. IEEE
   (2013)
4. Ceron, J.D., Lopez, D.M., Ramirez, G.A.: A mobile system for sedentary behaviors
   classification based on accelerometer and location data. Comput. Ind. **92**, 25–31
   (2017)
5. Ciucurel, C., Iconaru, E.I.: The importance of sedentarism in the development of
   depression in elderly people. Proc. - Soc. Behav. Sci. **33** (Supplement C), 722–726
   (2012). https://doi.org/10.1016/j.sbspro.2012.01.216. http://www.sciencedirect.
   com/science/article/pii/S1877042812002248. pSIWORLD 2011
6. Damen, D., et al.: Scaling egocentric vision: the dataset. In: Ferrari, V., Hebert,
   M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11208, pp. 753–771.
   Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01225-0_44
7. Dernbach, S., Das, B., Krishnan, N.C., Thomas, B.L., Cook, D.J.: Simple and
   complex activity recognition through smart phones. In: 2012 8th International
   Conference on Intelligent Environments (IE), pp. 214–221, June 2012. https://doi.
   org/10.1109/IE.2012.39
8. Galván-Tejada, C.E., et al.: An analysis of audio features to develop a human
   activity recognition model using genetic algorithms, random forests, and neural
   networks. Mob. Inf. Syst. **2016**, 1–10 (2016)

204   E. Garcia-Ceja et al.

9. Garcia, E.A., Brena, R.F.: Real time activity recognition using a cell phone's accelerometer and Wi-Fi. In: Workshop Proceedings of the 8th International Conference on Intelligent Environments. Ambient Intelligence and Smart Environments, vol. 13, pp. 94–103. IOS Press (2012). https://doi.org/10.3233/978-1-61499-080-2-94

10. Garcia-Ceja, E., Brena, R.: Building personalized activity recognition models with scarce labeled data based on class similarities. In: García-Chamizo, J.M., Fortino, G., Ochoa, S.F. (eds.) UCAmI 2015. LNCS, vol. 9454, pp. 265–276. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26401-1_25

11. Garcia-Ceja, E., Riegler, M., Nordgreen, T., Jakobsen, P., Oedegaard, K.J., Tørresen, J.: Mental health monitoring with multimodal sensing and machine learning: a survey. Pervasive Mob. Comput. **51**, 1–26 (2018). https://doi.org/10.1016/j.pmcj.2018.09.003. http://www.sciencedirect.com/science/article/pii/S1574119217305692

12. Hayashi, T., Nishida, M., Kitaoka, N., Takeda, K.: Daily activity recognition based on DNN using environmental sound and acceleration signals. In: 2015 23rd European Signal Processing Conference (EUSIPCO), pp. 2306–2310, August 2015. https://doi.org/10.1109/EUSIPCO.2015.7362796

13. Khaleghi, B., Khamis, A., Karray, F.O., Razavi, S.N.: Multisensor data fusion: a review of the state-of-the-art. Inf. Fusion **14**(1), 28–44 (2013). https://doi.org/10.1016/j.inffus.2011.08.001. http://www.sciencedirect.com/science/article/pii/S1566253511000558

14. Ligges, U., Krey, S., Mersmann, O., Schnackenberg, S.: tuneR: Analysis of music (2014). http://r-forge.r-project.org/projects/tuner/

15. Mannini, A., Sabatini, A.M.: Machine learning methods for classifying human physical activity from on-body accelerometers. Sensors **10**(2), 1154–1175 (2010). https://doi.org/10.3390/s100201154. http://www.mdpi.com/1424-8220/10/2/1154

16. Margarito, J., Helaoui, R., Bianchi, A.M., Sartor, F., Bonomi, A.G.: User-independent recognition of sports activities from a single wrist-worn accelerometer: a template-matching-based approach. IEEE Trans. Biomed. Eng. **63**(4), 788–796 (2016)

17. Mitchell, E., Monaghan, D., O'Connor, N.E.: Classification of sporting activities using smartphone accelerometers. Sensors **13**(4), 5317–5337 (2013)

18. Nishida, M., Kitaoka, N., Takeda, K.: Development and preliminary analysis of sensor signal database of continuous daily living activity over the long term. In: 2014 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), pp. 1–6. IEEE (2014)

19. Richter, J., Wiede, C., Dayangac, E., Shahenshah, A., Hirtz, G.: Activity recognition for elderly care by evaluating proximity to objects and human skeleton data. In: Fred, A., De Marsico, M., Sanniti di Baja, G. (eds.) ICPRAM 2016. LNCS, vol. 10163, pp. 139–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53375-9_8

20. Vaizman, Y., Ellis, K., Lanckriet, G., Weibel, N.: Extrasensory app: data collection in-the-wild with rich user interface to self-report behavior. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2018)

21. Wang, X., Rosenblum, D., Wang, Y.: Context-aware mobile music recommendation for daily activities. In: Proceedings of the 20th ACM International Conference on Multimedia, pp. 99–108. ACM (2012)

22. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems, 3rd edn. Morgan Kaufmann, Burlington (2011)
23. Zhang, M., Sawchuk, A.A.: Motion primitive-based human activity recognition using a bag-of-features approach. In: ACM SIGHIT International Health Informatics Symposium (IHI), Miami, Florida, USA, pp. 631–640, January 2012

A.7 Paper VII - Kvasir-Instrument: Diagnostic and Therapeutic tool Segmentation Dataset in Gastrointestinal Endoscopy

Authors: Debesh Jha, Sharib Ali, Krister Emanuelsen, Steven A. Hicks, **Vajira Thambawita**, Enrique Garcia-Ceja, Michael A. Riegler, Thomas de Lange, Peter T. Schmidt, Håvard D. Johansen, Dag Johansen, Pål Halvorsen

Abstract: Gastrointestinal (GI) pathologies are periodically screened, biopsied, and resected using surgical tools. Usually, the procedures and the treated or resected areas are not specifically tracked or analysed during or after colonoscopies. Information regarding disease borders, development, amount, and size of the resected area get lost. This can lead to poor follow-up and bothersome reassessment difficulties post-treatment. To improve the current standard and also to foster more research on the topic, we have released the "Kvasir-Instrument" dataset, which consists of 590 annotated frames containing GI procedure tools such as snares, balloons, and biopsy forceps, etc. Besides the images, the dataset includes ground truth masks and bounding boxes and has been verified by two expert GI endoscopists. Additionally, we provide a baseline for the segmentation of the GI tools to promote research and algorithm development. We obtained a dice coefficient score of 0.9158 and a Jaccard index of 0.8578 using a classical U-Net architecture. A similar dice coefficient score was observed for DoubleUNet. The qualitative results showed that the model did not work for the images with specularity and the frames with multiple tools, while the best result for both methods was observed on all other types of images. Both qualitative and quantitative results show that the model performs reasonably good, but there is potential for further improvements. Benchmarking using the dataset provides an opportunity for researchers to contribute to the field of automatic endoscopic diagnostic and therapeutic tool segmentation for GI endoscopy.

Candidate contributions: Vajira contributed to drafting and revising the paper.

Thesis objectives: Sub-objective I, Sub-objective II

Kvasir-Instrument: Diagnostic and Therapeutic Tool Segmentation Dataset in Gastrointestinal Endoscopy

Debesh Jha[1,2(✉)], Sharib Ali[9], Krister Emanuelsen[3], Steven A. Hicks[1,5],
Vajira Thambawita[1,5], Enrique Garcia-Ceja[10], Michael A. Riegler[1],
Thomas de Lange[4,6,7], Peter T. Schmidt[8], Håvard D. Johansen[2],
Dag Johansen[2], and Pål Halvorsen[1,5]

[1] SimulaMet, Oslo, Norway
debesh@simula.no
[2] UIT The Arctic University of Norway, Tromsø, Norway
[3] Simula Research Laboratory, Oslo, Norway
[4] Augere Medical AS, Oslo, Norway
[5] Oslo Metropolitan University, Oslo, Norway
[6] Medical Department, Sahlgrenska University Hospital-Mölndal,
Gothenburg, Sweden
[7] Department of Medical Research, Bærum Hospital, Gjettum, Norway
[8] Karolinska University Hospital, Solna, Sweden
[9] Department of Engineering Science, University of Oxford, Oxford, UK
[10] Sintef Digital, Oslo, Norway

Abstract. Gastrointestinal (GI) pathologies are periodically screened, biopsied, and resected using surgical tools. Usually, the procedures and the treated or resected areas are not specifically tracked or analysed during or after colonoscopies. Information regarding disease borders, development, amount, and size of the resected area get lost. This can lead to poor follow-up and bothersome reassessment difficulties post-treatment. To improve the current standard and also to foster more research on the topic, we have released the "Kvasir-Instrument" dataset, which consists of 590 annotated frames containing GI procedure tools such as snares, balloons, and biopsy forceps, etc. Besides the images, the dataset includes ground truth masks and bounding boxes and has been verified by two expert GI endoscopists. Additionally, we provide a baseline for the segmentation of the GI tools to promote research and algorithm development. We obtained a dice coefficient score of 0.9158 and a Jaccard index of 0.8578 using a classical U-Net architecture. A similar dice coefficient score was observed for DoubleUNet. The qualitative results showed that the model did not work for the images with specularity and the frames with multiple tools, while the best result for both methods was observed on all other types of images. Both qualitative and quantitative results show that the model performs reasonably good, but there is potential for further improvements. Benchmarking using the dataset provides an opportunity for researchers to contribute to the field of automatic endoscopic diagnostic and therapeutic tool segmentation for GI endoscopy.

Keywords: Gastrointestinal endoscopy · Tool segmentation · Endoscopic tools · Convolutional neural network · Benchmarking

1 Introduction

Minimally Invasive Surgery (MIS) is a commonly used technique in surgical procedures. The advantage of MIS is that small surgical incisions are made in the patient for endoscopy that causes less pain, reduced time of the hospital stay, fast recovery, reduced blood loss, and less scaring process as compared to the traditional open surgery. The nature of the operation is complex, and the surgeons have to precisely tackle hand-eye coordination, which may lead to restricted mobility and a narrow field of view [5].

However, unlike the treatment of accessory organs such as liver and pancreas, no incision is required forGastrointestinal (GI) tract organs (*oesophagus, stomach, duodenum, colon, and rectum*). GI procedures also include both minimally invasive surveillance and treatment (*including surgery*) procedures. A varied number of tools are used as per the requirement of these procedures. For example, balloon dilatation to help open the GI surface, biopsy forceps for tissue sample collection, polyp removal with snares, and submucosal injections.

A computer and robotic-assisted surgical system can enhance the capability of the surgeons [9]. It can provide the opportunity to gain additional information about the patient, which can be useful for decision making during surgery [6]. However, it is difficult to understand the spatial relationship between surgical instruments, cameras, and anatomy for the patient [11]. In GI endoscopy, it is vital to track and guide surgeons during tumor resection or biopsy collection from a defined site and help to correlate the biopsied samples and treatment locations post-diagnostic and therapeutic or surgical procedures. While most datasets and automated-algorithm developments for instrument segmentation are mostly focused on laparoscopy-based surgical removal, automatic guidance of tools for GI surgery has not been addressed before.

New developments in the area of robot-assisted systems show that there is potential for developing a fully automated robotic surgeon [14]. The da Vinci robot is a surgical system that is considered the de-facto standard-of-care for certain urological, gynecological, and general procedures [4]. Thus, it is critical to have information regarding intra-operative guidance, which plays an essential role in decision making. However, there are specific challenges, such as limited field of view and difficulties with the surgeons handling the instruments during surgery [13]. Therefore, image-based instrument segmentation and tracking are gaining more and more attention in both robotic and non-robotic minimally invasive surgery. Previous work targeting instrument segmentation, detection, and tracking on endoscopic video images failed on challenging images such as images with blood, smoke, and motion artifacts [13]. Other reasons that make semantic segmentation of surgical instruments a challenging task are the presence of images containing shadows, specular reflections, blood, camera lens fogging, and the complex background tissue [14]. The segmentation masks of these images can be useful for instrument detection and tracking.

Similarly, in the GI tract procedures, from tissue sample collection to surgical removal of pathologies is performed in low field-of-view areas. Visual clutter such as artifacts, moving objects, and fluid, hinders the localisation of the target site during surgical procedures. Additionally, currently, there is no way of correlating the tissue sample collection with biopsied location and assessing surgical procedure effectiveness or even post-treatment recovery analysis. Automated localisation and tracking of tools can help guide the endoscopists and surgeons to perform their tasks more effectively. Also, post-procedure video analysis can be done using these automated methods to track such tools, thus enabling improved surgical procedures or surveillance and their post-assessment. Currently, this is an open problem in the research community, where most procedures are not automated in GI tract endoscopy.

While there is an open research question for automated tool detection and guidance in GI procedures, there is a lack of available public datasets. We aim to initiate the development of automated systems for the segmentation of GI tract diagnostic and therapeutic endoscopy tools. This research direction will enable tracking and localisation of essential tools used in endoscopy and help to improve targeted biopsies and surgeries in complex GI tract organs. To accomplish this, and to address the lack of publicly available labeled datasets, we have publicly released 590 pixel-level annotated frames that comprise of tools such as balloon dilation for facilitating the opening of GI organs, biopsy forceps for tissue sample collection, polyp removal with snares, submucosal injections, radiofrequency ablation of dysplastic mucosa using probes and some other related surgical/diagnostic procedures. The released video frames will allow for building automated Machine Learning (ML) algorithms that can be applied during clinical procedures or post-analyses. To commence this effort, we provide a baseline benchmark on this dataset. U-Net [12] is a common semantic segmentation based architecture for medical image segmentation tasks. In this paper, we therefore present results utilising two U-Net based architectures. The provided dataset is open and can be used for research and development, and we invite medical imaging, computer vision, ML and multimedia researchers to develop novel algorithms on the provided dataset. The main contributions of this paper are:

– The release of 590 annotated images with bounding boxes and segmentation masks of GI diagnostic and surgical tool dataset. To the best of our knowledge, this is the first dataset of segmented tools used in the GI tract.
– A benchmark of the provided dataset using the U-Net [12] and Double-UNet [10] architectures for semantic segmentation is provided.

2   Related Work

Surgical vision is evolving as a promising technique to segment and track instruments using endoscopic images [6]. To gather researchers on a single platform, the *Endoscopic vision (EndoVis) challenge* has been organized since 2015 at Medical Image Computing and Computer Assisted Intervention Society (MICCAI)

Table 1. Similar available datasets

Dataset	Content	Task type	Procedure
Instrument segmentation and tracking (2015) [6]	Rigid and robotic instruments	Segmentation and tracking	Laparoscopy
Robotic Instrument Segmentation (2017) [4]	Robotic surgical instruments	Binary segmentation, part based segmentation, instrument segmentation	Abdominal porcine
Robotic Scene Segmentation (2018) [3]	Surgical instruments and other	Multi-instance segmentation	Robotic nephrectomy
Robust Medical instrument segmentation (2019) [13]	Laparoscopic instrument	Binary segmentation, multiple instance detection, multiple instance segmentation	Laparoscopy
Kvasir-Instrument (Ours)	Diagnostic and therapeutic tools in endoscopic images	Binary segmentation, detection and localization	Gastroscopy & colonoscopy

with an exception in 2016. The EndoVis challenge hosts different sub-challenges. The year-wise information about the hosted sub-challenge can be found on the challenge website[1].

Bodenstedt et al. [6] organized "EndoVis 2015 Instrument sub-challenge" for developing new techniques and benchmarking ML algorithms for segmentation and tracking of the instruments on a common dataset. The organizers challenged on two different tasks, i.e., (1) Segmentation and (2) Tracking. The goal of the challenge was to address the problem related to segmentation and tracking of articulated instruments in both laparoscopic and robotic surgery[2]. A comprehensive evaluation of the methods used in instrument segmentation and tracking task for minimally invasive surgery is summarized in this work [6]. The extensive evaluation showed that deep learning works well for instrument segmentation and tracking tasks.

In 2017, a follow up to the previous 2015 challenge was organized called "Robotic Instrument Segmentation Sub-Challenge"[3]. The challenge was part of the Endoscopic vision challenge that was organized at MICCAI 2017. This challenge offered three tasks: (1) Binary segmentation, (2) Parts based segmentation, and (3) Instrument type segmentation. The goal of the binary segmentation

---

[1] https://endovis.grand-challenge.org/.
[2] https://endovissub-instrument.grand-challenge.org/EndoVisSub-Instrument/.
[3] https://endovissub2017-roboticinstrumentsegmentation.grand-challenge.org/.

task was to separate the image into an instrument and background. Parts segmentation challenged the participants to divide the binary instrument into a shaft, wrist, and jaws. Type segmentation challenged the participants to identify different instrument types. A detailed description of the challenge tasks, dataset, methodologies used by ten participating teams in different tasks, challenge design, and limitation of the challenge can be found in the challenge summary paper [4].

In 2019, a similar challenge called "Robust Medical Instrument Segmentation Challenge 2019"[4] was organized by Roß et al. [13]. This challenge offered three tasks (1) Binary segmentation, (2) Multiple instance detection, and (3) Multiple instance segmentation. The challenge was focused on addressing two key issues in surgical instruments, *Robustness* and *Generalization*, and benchmark medical instrument segmentation and detection on the provided surgical instrument dataset. Endoscopic artefact detection challenge (EAD2019) challenge focused on endoscopic artifact detection primarily but also included instrument class in their detection, segmentation, and "out-of-sample" generalisation tasks. The challenge outcome revealed that most methods performed well for instrument detection and segmentation class [2]. However, this dataset mostly consisted of large biopsy forceps.

In Table 1, we present available instrument datasets in the field of tool segmentation. All of the datasets were designed for hosting challenges. The training dataset is released for all the datasets (except ROBUST-MIS); however, the test dataset is not provided by the challenge organizers. Thus, it makes it difficult to calculate and compare the results on the test dataset. However, experiments are still possible by splitting the training dataset into train, validation, and testing sets. The Robust Medical instrument segmentation dataset is yet not public. However, the participants who have participated in the challenge have the opportunity to download the training dataset. Usually, there are certain practicalities to download the dataset, such as signing the agreement and getting permission from the owner, which takes time, and it is inconvenient. Moreover, to participate in the challenge, the participants have to signup in a particular year, and usually, it often takes a very longtime before they publish the dataset. Thus, the significance of the datasets becomes less as the technology is changing rapidly. More information on available instrument datasets, contents, and offered tasks by the organizers and about the availability can be found from Table 1.

The literature review shows that there are only a few open-access datasets for MIS instrument segmentation. Moreover, to the best of our knowledge, GI tract tools have never been explored. This is the first attempt to provide the community with a curated and annotated public dataset that comprises diagnostic and therapeutic tools in the GI tract. We believe that the presented dataset and the widely used U-Net based algorithm benchmark will encourage the researchers to develop robust and efficient algorithms using the provided dataset that can help clinical procedures in endoscopy.

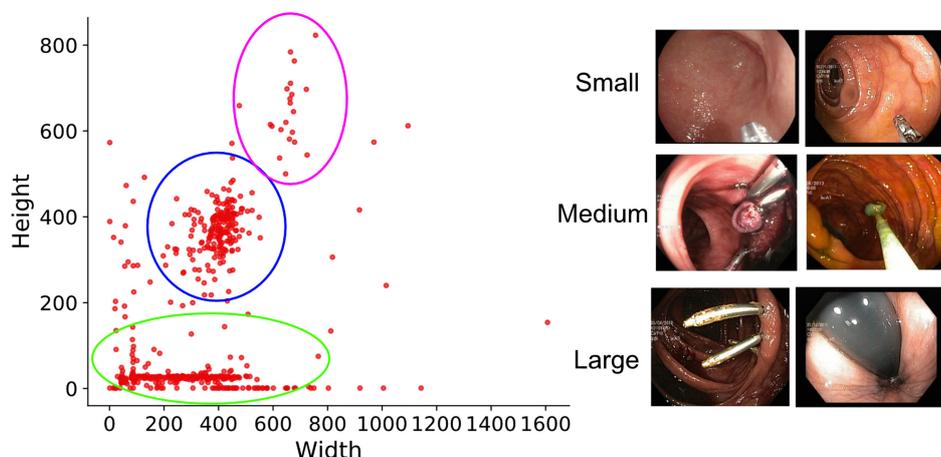[4] https://robustmis2019.grand-challenge.org/.

Fig. 1. Distribution of Kvasir-Instrument dataset. On left: Small (green), medium (blue) and large (pink) sized tool clusters. On right: sample images with variable tool size in images. (Color figure online)

3   Kvasir-Instrument Dataset

In this section, we introduce the Kvasir-Instrument dataset with details on how the data was collected, the annotation protocol, and the dataset's structure. The dataset was collected from endoscopic examinations performed at Bærum Hospital in Norway. The unlabelled images' frames are selected from the HyperKvasir dataset [7].

HyperKvasir provides frame-level annotations for 10,662 frames for 23 different classes. However, the majority of the images (99,417 frames) are not labeled. We trained a model using the labeled samples of this dataset and tried to predict the classes of the unlabeled samples. Although our algorithm [15,16] could not classify all the images correctly; however, we were able to classify the presence of instrument or tool out of thousands of provided image frames. However, in order to perform segmentation, pixel-wise masks and bounding boxes were missing. This is what is provided in the proposed dataset, and below, we present the acquisition and annotation protocols used in the data preparation:

3.1   Data Acquisition

The images and videos were collected using standard endoscopy equipment from Olympus (Olympus Europe, Germany) and Pentax (Pentax Medical Europe, Germany) at Bærum Hospital, Vestre Viken Hospital Trust, Norway. All the data used in this study were obtained from videos for procedures that had followed the patient consenting protocol of Bærum Hospital. Additionally, no patient information was available. We have performed a random naming for each publicly released image for further effective annonymisation.

224   D. Jha et al.
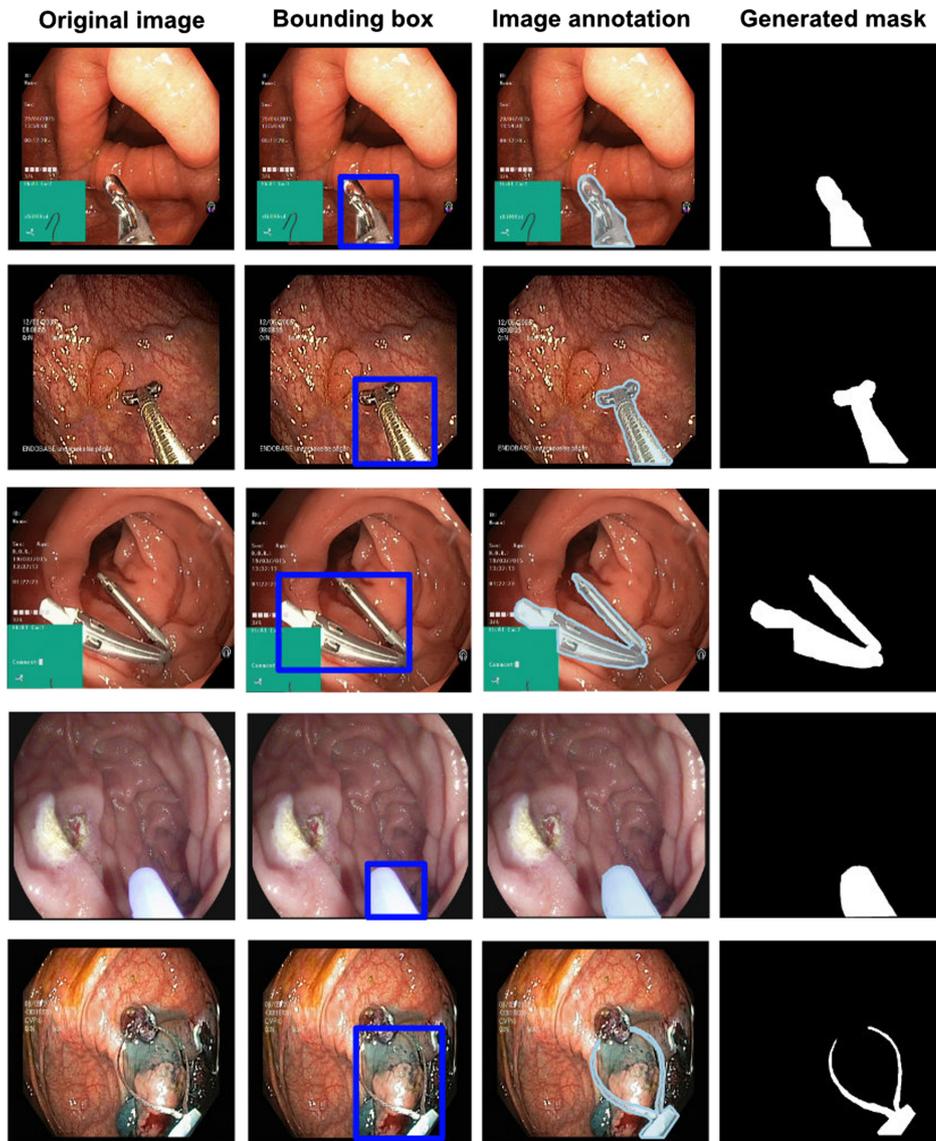
Fig. 2. Kvasir-Instrument dataset: first two rows represent frames with biopsy forceps, the middle row consist of metallic clip, the fourth row is a radio-frequency ablation probe and the last row depicts the crescent and hexagonal shaped snares for polyp removal.

3.2   Annotation Strategy

We have uploaded the Kvasir-Instrument dataset to labelbox[5] and labeled the Region of Interest (ROI) in the image frames, i.e., the ROI of diagnostic and

----

[5] https://www.labelbox.com/.

therapeutic tools in our cases, and generated all the ground truth masks. Figure 2 shows the example images, bounding box, image annotation, and generated masks for the Kvasir-Instrument dataset. All annotations were then exported in a JSON format, which was used to generate masks for each of the annotations. Related source codes and more information about the dataset can be found at https://github.com/DebeshJha/Kvasir-Instrument.

The exported file contained the information of the images along with the coordinate points that were used for mask and bounding box generation. All annotations were performed using a three-step strategy:

1. The selected samples were labeled by two experienced research assistants.
2. The annotated samples were cross-validated for their delineation quality by two experienced GI experts (more than 10 years of work experience in colonoscopy).
3. The suggested changes were incorporated using the comments from the experts.

The Kvasir-Instrument dataset includes 590 frames consisting of various GI endoscopy tools used during both endoscopic surveillance and therapeutic or surgical procedures. A thorough annotation strategy (detailed above) was used to create bounding boxes and segmentation masks. The dataset consists of variable tool size with respect to image height and width, as presented in Fig. 1. The majority of the tools are small and medium-sized. The sample bounding box annotation, precise area delineation, and extracted masks are shown in Fig. 2.

Our dataset is publicly available and can be accessed at https://datasets.simula.no/kvasir-instrument/. It consists of original image samples (in JPEG format), their corresponding masks (in PNG format), and bounding box information (in JSON format).

4   Benchmarking, Results and Discussion

In this section, we explore encoder-decoder based classical models for baseline algorithm benchmarking, their implementation details for reproducibility, details on evaluation metric used for quantitative analysis, and results and discussion.

4.1   Baseline Methods

U-Net [12] has been explored in the past through many biomedical segmentation challenges and has shown strength towards an effective supervised segmentation model. In this paper, we, therefore, use U-Net based architectures on our Kvasir-Instrument dataset to provide a baseline result for future comparisons. U-Net uses an encoder-decoder architecture, that is, a contractive feature extraction path and expansive path with a classifier to perform binary classification of each image pixel in an upsampled feature map. In our previous work, we have shown that the strength of supervised classification can be amplified by using the output mask from one U-Net [12] architecture to the other by proposing DoubleUNet [10]. In addition, the DoubleUNet architecture uses VGG-19 pretrained

226   D. Jha et al.

on ImageNet as one of the encoder blocks, squeeze and excite block, and Atrous spatial pyramid pooling (ASPP) block. All other components in the network remain the same as the U-Net. For both networks, dice loss gives a $1 - DSC$, where DSC is the dice similarity coefficient (see Eq. 1 below).

4.2   Implementation Details

We have implemented the U-Net-based and DoubleUNet based architectures using the Keras framework [8] with TensorFlow [1] as backend running on the Experimental Infrastructure for Exploration of Exascale Computing (eX3), NVIDIA DGX-2 machine. We have resized the training dataset into 512×512. We set the batch size of 8 for training. Both architectures are optimized by using Adam optimizer. We have made use of dice loss as the loss function. We split the dataset using 80% of the dataset for training and the remaining 20% for the testing (evaluation). The same split is also provided in the dataset for the further research. We performed basic augmentation, such as horizontal flip, vertical flip, and random rotation. Moreover, we have also provided the train-test split so that others can improve the methods on the same dataset.

4.3   Evaluation Metrics

In this medical image segmentation approach, each pixel of the diagnostic and therapeutic tool either belongs to a tool or non-tool region. The Dice similarity coefficient (DSC) is the mainly used for result evaluation in medical image segmentation. Additionally, we calculate other standard metrics such as Jaccard similarity coefficient (JC) (also known as the intersection over union (IoU)), precision, recall, overall accuracy, F2, and frames per second (FPS). Using tp, fp, tn, and fn to represent the true positives, false positives, true negatives, and false negatives, respectively, the mathematical formulas for them are as follows:

$$\text{DSC} = \frac{2 \cdot tp}{2 \cdot tp + fp + fn} \tag{1}$$

$$\text{JC or IoU} = \frac{tp}{tp + fp + fn} \tag{2}$$

$$\text{Recall } (r) = \frac{tp}{tp + fn} \tag{3}$$

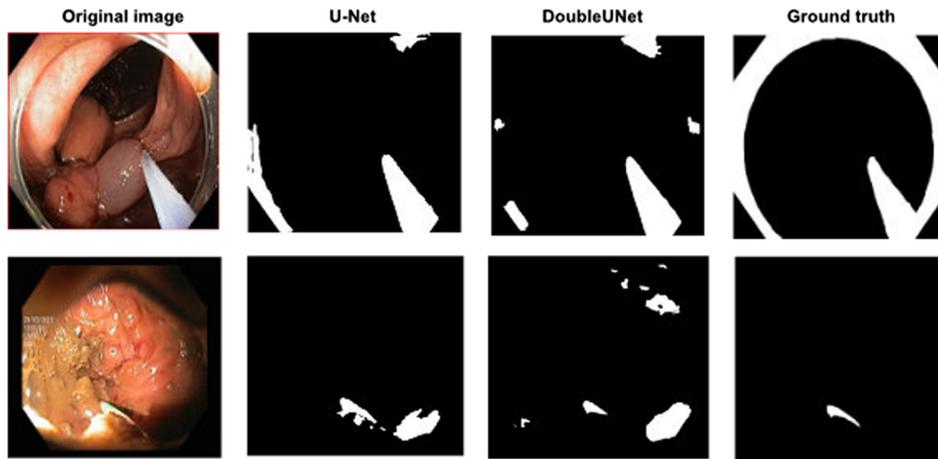$$\text{Precision } (p) = \frac{tp}{tp + fp} \tag{4}$$

$$\text{F2} = \frac{5p \times r}{4p + r} \tag{5}$$

$$\text{Overall accuracy } (Acc.) = \frac{tp + tn}{tp + tn + fp + fn} \tag{6}$$

$$\text{Frame Per Second } (FPS) = \frac{\#frames}{sec} \tag{7}$$

Table 2. Baseline results for tool segmentation

Method	JC	DSC	F2-score	Precision	Recall	Acc.	FPS
U-Net [12]	**0.8578**	**0.9158**	**0.9320**	**0.8998**	**0.9487**	**0.9864**	**20.4636**
DoubleUNet [10]	0.8430	0.9038	0.9147	0.8966	0.9275	0.9838	10.0000

Fig. 3. Failed cases: cap region (top) is under-segmented and small clip area is over-segmented and consist of large number of false positives (bottom).

4.4 Quantitative and Qualitative Results

Table 2 shows the results of the baseline methods for the tool segmentation on the proposed Kvasir-Instrument dataset. From the table, we can observe that the UNet achieved a high JC of 0.8578 and DSC of 0.9158, which is slightly above than the DoubleUNet that yielded JC of 0.8430 and DSC of 0.9038. Also, UNet achieved a speed of 20.4636 FPS, whereas computational time is double for DoubleUNet with only 10 FPS. Similarly, both the recall and precision scores are very comparable for both U-Net ($p = 0.8998, r = 0.9487$) and DoubleUNet ($p = 0.8966, r = 0.9275$).

Figure 3 shows the qualitative result on two challenging sample images. It can be observed that both UNet and DoubleUNet are under-segmenting the cap region (top) and over-segmenting the small clip area (bottom). Some parts of these images are confused because of the presence of saturation areas. However, both models were able to segment well with most endoscopic tool samples in the dataset. This is also evident from the quantitative results. However, even better models are still needed to motivate further research.

4.5 Discussion

From the experimental results in Table 2, we can validate that the classical U-Net architecture outperforms DoubleUNet model. Additionally, U-Net is 2×

faster than the DoubleUNet. This is because U-Net uses basic convolution
blocks, whereas DoubleUNet uses pre-trained encoders, ASPP, squeeze, and
excite blocks, all of which increase the inference latency. Here, the UNet is opti-
mized by dice loss instead of binary cross-entropy loss, which showed improved
performance during our experiments.

Further, fine-tuning on other similar datasets, rigorous data augmentation,
and applying more advanced Deep learning (DL) techniques can improve the
baseline results - eventually achieving the detection, localisation, and segmenta-
tion performance needed to make the technology useful in a clinical environment.
Additionally, the use of DL networks with fewer parameters could increase com-
putational efficiency, thereby enabling real-time systems that can be used in
clinical settings effectively.

5   Conclusion

We have curated, annotated, and publicly released a dataset that contains *endo-
scopic tools* used in GI examinations and surgical procedures. The dataset con-
sists of images, bounding boxes, and segmentation masks of endoscopy tools used
during different procedures in the GI tract. Additionally, we provided baseline
segmentation methods for the automatic delineation of these tools and have com-
pared them using standard computer vision metrics. In the future, we plan to
continuously increase the amount of data and also call for multimedia challenges
using the presented dataset.

References

1. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: Pro-
ceedings of USENIX Symposium on Operating Systems Design and Implementa-
tion, pp. 265–283 (2016)
2. Ali, S., et al.: An objective comparison of detection and segmentation algorithms
for artefacts in clinical endoscopy. Sci. Rep. **10**(1), 1–15 (2020)
3. Allan, M., Azizian, M.: Robotic scene segmentation sub-challenge. arXiv preprint
arXiv:1902.06426 (2019)
4. Allan, M., et al.: 2017 robotic instrument segmentation challenge. arXiv preprint
arXiv:1902.06426 (2019)
5. Bernhardt, S., Nicolau, S.A., Soler, L., Doignon, C.: The status of augmented
reality in laparoscopic surgery as of 2016. Med. Image Anal. **37**, 66–90 (2017)
6. Bodenstedt, S., et al.: Comparative evaluation of instrument segmentation and
tracking methods in minimally invasive surgery. arXiv preprint arXiv:1805.02475
(2018)

7. Borgli, H., et al.: Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. Sci. Data **7**(1), 1–14 (2020)
8. Chollet, F., et al.: Keras (2015)
9. Cleary, K., Peters, T.M.: Image-guided interventions: technology review and clinical applications. Annu. Rev. Biomed. Eng. **12**, 119–142 (2010)
10. Jha, D., Riegler, M., Johansen, D., Halvorsen, P., Håvard, J.: DoubleU-net: a deep convolutional neural network for medical image segmentation. In: Proceedings of 33rd International Symposium on Computer-Based Medical Systems, pp. 558–564 (2020)
11. Pakhomov, D., Premachandran, V., Allan, M., Azizian, M., Navab, N.: Deep residual learning for instrument segmentation in robotic surgery. In: Suk, H.-I., Liu, M., Yan, P., Lian, C. (eds.) MLMI 2019. LNCS, vol. 11861, pp. 566–573. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32692-0_65
12. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
13. Ross, T., et al.: Robust medical instrument segmentation challenge 2019. arXiv preprint arXiv:2003.10299 (2020)
14. Shvets, A.A., Rakhlin, A., Kalinin, A.A., Iglovikov, V.I.: Automatic instrument segmentation in robot-assisted surgery using deep learning. In: Proceedings of International Conference on Machine Learning and Applications, pp. 624–628 (2018)
15. Thambawita, V., et al.: The medico-task 2018: disease detection in the gastrointestinal tract using global features and deep learning. arXiv preprint arXiv:1810.13278 (2018)
16. Thambawita, V., et al.: An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification. arXiv preprint arXiv:2005.03912 (2020)

A.8 Paper VIII - The Medico-Task 2018: Disease Detection in the Gastrointestinal Tract using Global Features and Deep Learning

Authors: Vajira Thambawita, Debesh Jha, Michael Riegler, Pål Halvorsen, Hugo Lewi Hammer, Håvard D. Johansen, Dag Johansen

Abstract: In this paper, we present our approach for the 2018 Medico Task classifying diseases in the gastrointestinal tract. We have proposed a system based on global features and deep neural networks. The best approach combines two neural networks, and the reproducible experimental results signify the efficiency of the proposed model with an accuracy rate of 95.80%, a precision of 95.87%, and an F1-score of 95.80%.

Candidate contributions: In this working notepaper, Vajira is the first author and the corresponding author. He contributed to the main conception and design of three experiments (out of five) using deep learning approaches which use Resnet-152, Densenet-161, and a combination of these. Vajira's experiments achieved the best performance of this paper. He developed and analyzed the results of the three experiments. Vajira contributed to draft the article and revise it.

Thesis objectives: Sub-objective I, Sub-objective III

The Medico-Task 2018: Disease Detection in the Gastrointestinal Tract using Global Features and Deep Learning

Vajira Thambawita[1,3], Debesh Jha[1,4], Michael Riegler[1,3,5], Pål Halvorsen[1,3,5],
Hugo Lewi Hammer[2], Håvard D. Johansen[4], and Dag Johansen[4]

[1]Simula Research Laboratory, Norway   [2]Oslo Metropolitan University, Norway   [3]Simula Metropolitan, Norway
[4]University of Tromsø, Norway   [5]University of Oslo, Norway
Contact:vajira@simula.no,debesh@simula.no

ABSTRACT

In this paper, we present our approach for the 2018 Medico Task classifying diseases in the gastrointestinal tract. We have proposed a system based on global features and deep neural networks. The best approach combines two neural networks, and the reproducible experimental results signify the efficiency of the proposed model with an accuracy rate of 95.80%, a precision of 95.87%, and an F1-score of 95.80%.

1 INTRODUCTION

Our main goal for the Medico Task [15] is to classify findings in images from the Gastrointestinal (GI) tract. This task provides two types of input data: Global Features (GFs) and original images. The 2017 Medico Task consisted of a balanced dataset with only 8 classes [12] whereas the current task consists of a highly imbalanced dataset with 16 classes [11, 12], i.e., making this years task more complicated. Different approaches have been used in the last year medico task [5, 7, 9, 10, 14, 17] based on GFs extractions and Convolutional Neural Networks (CNN) methods. We extend upon these solutions and present our solutions based on both GFs and transfer learning mechanisms using CNN. We achieve best results combining two CNNs and using an extra multilayer perceptron to combine the outputs of the two networks.

2 APPROACHES

We approach the problem of GI tract disease detection with small training datasets using five different methods: two based on GF extractions, and three based on CNN with transfer learning described below.

2.1 Global-feature-based approaches

Method 1 and **Method 2** use the concept of GFs. For the extraction of GFs, we use Lucence Image Retrieveal (LIRE) [6]. GFs are easy and fast to calculate, and can also be used for image comparison, image collection search and distance computing [14]. Based on [13, 16], we use Joint Composite feature (JCD), Tamura, Color layout, Edge Histogram, Auto Color Correlogram and Pyramid Histogram of Oriented Gradients (PHOG). These features represent the overall properties of the images. Adding more GFs is possible, but it may increase the redundant information which can reduce the overall classification performance.

The extracted features are sent to the different machine learning classifier for the multi-class classification. **Method 1** makes the use

of extracted GFs that are sent to SimpleLogistic (SL) classifier. We input the same selected set of features to the logistic model tree (LMT) classifier in **Method 2**.

2.2 Transfer learning based approaches

Our CNN approaches use transfer learning mechanism with pre-trained models using the ImageNet dataset [18]. Resnet-152 [3] and Densenet-161 [4] have been selected, and this selection is based on top 1-error and top-5-errors rate of pre-trained networks in the Pytorch [8] deep learning framework.

One of the main problems of the given dataset is the "out of patient"-category which has only four images while other classes have a considerable number. The colour distribution of this class shows a completely different colour domain compared to the other categories. We identified this difference via manual investigations of the dataset and moved all four images of this category into the corresponding validation set folder. Then, the training set folder is filled with random Google images which are not related to the GI tract. To overcome the problems of stopping training in a local minima, we use the stochastic gradient descent [1] method with dynamic learning rate scheduling. The losses (loss 1 and loss 2 in Figure 1) of CNN methods were calculated for each network separately. Additionally, horizontal flips, vertical flips, rotations and re-sizing data augmentations have been applied to overcome the problem of over-fitting.

Method 3 uses transfer learning with Resnet-152 which has the top-1-error and top-5-error rates. The last fully connected layer of Resnet-152, which is originally designed to classify 1000 classes of the ImageNet dataset, has been changed to classify the 16 classes in the MEdico task. Usually, the transfer learning freezes pre-trained layers to avoid back propagation of large errors. This is because of newly added layers with random weights. However, we did not freeze the pre-trained layers, because modifying only the last layer cannot propagate huge errors backwards in transfer learning. The network was trained until it reached to the maximum validation accuracy of the validation dataset.

Method 4 extends Method 3 by using two parallel pre-trained models, Resnet-152 and Densenet-161, to get a cumulative decision at the end as depicted in Figure 1. The classification is based on an average of the two output probability vectors. Finally, one loss value was calculated and propagated for updating weights. However, this yields a restriction of updating weights of networks Resnet-152 and Densenet-161 separately as they required. Therefore, we calculated two different loss values (loss 1 and loss 2 in Figure 1) from each network to update their weights separately. Both
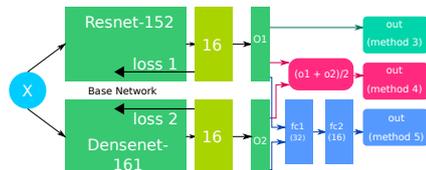
Figure 1: Block diagram of the CNN methods

networks were trained simultaneously until it reached to the best validation accuracy by changing hyper-parameters manually.

Method 5 was constructed to overcome the limitation of calculating the average of the probabilistic output of the two networks used in Method 4. Instead of calculating the average using the simple mathematical formula, another multilayer perceptron (MLP) has been merged with the above network to identify complex mathematical formula to get the cumulative decision as illustrated in Figure 1. Therefore, we passed the probability output of two networks (16 probabilities from each network) to a new MLP with 32 inputs, 16 outputs (via sigmoid layer) and one hidden layer with 32 units. In this, we used pre-trained Resnet-152 and Densenet-161 using the dataset and froze them before training the MLP. Then, we trained only the MLP to identify the best mathematical formula to get the cumulative decision.

3 RESULTS AND ANALYSIS

We have divided the development dataset into a training set (70%) and a validation set (30%). For the GFs based approach, ensembles of six extracted GFs were fetched to all the available machine learning classifiers (with different parameters) using WEKA[2] library. The SL and LMT classifiers outperform all other available classifiers for the dataset. The other promising classifier were Sequential minimal optimization (RBF kernel), and a combination of PCA with LibSVM (RBF) classifier.

On validation set, all the CNN methods (3-5) show accuracies of around 95% and specificities of around 99%. These are always better than the GFs based extraction methods (1,2) which have accuracies of around 82% and specificities of around 98%. According to the task organizers' evaluation results of the test dataset, Methods 3 to 5 show accuracies and specificities of around 99% again,which demonstrates our CNN methods are not overfitted with validation dataset.

Method 5 and 4 with Resnet-152 and Densenet-161 performs better compared to the Method 3 which has only Resnet-152 because of the capability of deciding the final answer based on two answers generated from two deep learning networks. However, getting a cumulative decision based on simple averaging function (Method 4) shows poor performance than the decision taken from a MLP (Method 5). As a result, Method 5 shows better results than method 4 by increasing the accuracy from 0.955 to 0.958. Therefore, Method 5 has been selected as our best method and confusion matrix represented in Table 1 was generated. An overview of the individual results obtained from five different experiments along with their performance metrics is presented in Table 2. Results obtained from the organizers for the test dataset is presented in the Table 3.

Table 1: The Confusion Matrix of Method 5 in our study

A:blurry-nothing, **B**:colon-clear, **C**:dyed-lifted-polyps, **D**:dyed-resection-margins, **E**:esophagitis,**F**:instruments, **G**:normal-cecum, **H**:normal-pylorus, **I**:normal-z-line, **J**:out-of-patient, **K**:polyps, **L**:retroflex-rectum, **M**:retroflex-stomach, **N**:stool-inclusions, **O**:stool-plenty, **P**:ulcerative-colitis

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	A	53	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	B	–	81	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	C	–	–	130	7	–	–	–	–	–	–	–	–	–	–	–	1
	D	–	–	3	122	–	–	–	–	–	–	–	–	–	–	–	–
	E	–	–	–	–	115	–	–	–	19	–	–	–	–	–	–	–
	F	–	–	–	–	–	10	–	–	–	–	1	–	–	–	–	–
Actual class	G	–	–	–	–	–	–	125	–	–	–	–	–	–	–	–	–
	H	–	–	–	–	–	–	–	132	–	–	–	–	–	–	–	–
	I	–	–	–	–	11	–	–	–	121	–	–	–	–	–	–	–
	J	–	–	–	–	–	1	–	–	–	3	–	–	–	–	–	–
	K	–	1	–	–	–	–	6	2	–	–	172	–	–	–	–	–
	L	–	–	–	–	–	–	1	–	–	–	–	71	–	–	–	–
	M	–	–	–	–	–	–	–	–	–	–	–	2	118	–	–	–
	N	–	–	–	–	–	–	–	–	–	–	–	–	–	39	–	–
	O	–	–	–	–	–	–	–	–	–	–	–	–	–	–	110	–
	P	–	–	–	–	1	1	2	–	–	–	4	1	–	–	–	129

Table 2: Validation results

Method	REC	PREC	SPEC	ACC	MCC	F1	FPS
1	0.855	0.793	0.989	0.816	0.814	0.823	79
2	0.816	0.817	0.984	0.816	0.800	0.815	12
3	0.9536	0.9543	0.9968	0.9536	0.9498	0.9535	64
4	0.9555	0.9563	0.9969	0.9555	0.9519	0.9554	29
5	0.9580	0.9587	0.9971	0.9580	0.9546	0.9580	29

Table 3: Official results

Method	REC	PREC	SPEC	ACC	MCC	F1
1	0.8457	0.8457	0.9897	0.9807	0.8353	0.8456
2	0.8457	0.8457	0.9897	0.9807	0.8350	0.8457
3	0.9376	0.9376	0.9958	0.9922	0.9335	0.9376
4	0.9400	0.9400	0.9960	0.9925	0.9360	0.9400
5	0.9458	0.9458	0.9964	0.9932	0.9421	0.9458

The main considerable point in the confusion matrix in Table 1 is misclassification between categories E: esophagitis and I: normal-z-line. A large number of misclassifications like 30 images from the validation set occurred and a manual investigation was done to identify the reason. We notice that the images of these two categories were very similar to each other because of the close location in the GI tract, and identifying these is also a challeng for physicians.

4 CONCLUSION

In this paper, we presented five different methods for the multi-class classification of GI tract diseases. The proposed approach are based on the GFs, and pre-trained CNN with transfer learning mechanism. The combination of Resnet-152 and Densenet-161 with an additional MLP achieved the highest performance with both the validation dataset and the test dataset provided by the task organizers. We show that a combination of pre-trained deep neural models on ImageNet has better capabilities to classify images into the correct classes because of cumulative decision-making capabilities. For future work, we will combine deeper CNNs parallelly to add more cumulative decision taking capabilities for classifying multi-class objects. In addition to that, Generative Adversarial Network (GAN) methods can be utilized to handle imbalance dataset by generating more data to train deep neural networks.

Appendix A. Published Articles

REFERENCES

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter (SIGKDD Explor. Newsl.)* 11, 1 (2009), 10–18.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 770–778.

[4] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely Connected Convolutional Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2261–2269.

[5] Yang Liu, Zhonglei Gu, and William K Cheung. 2017. HKBU at MediaEval 2017 Medico: Medical multimedia task. In *Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017)*.

[6] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. 2016. LIRE: open source visual information retrieval. In *Proceedings of the 7th International Conference on Multimedia Systems (MMSys)*. ACM, 30.

[7] Syed Sadiq Ali Naqvi, Shees Nadeem, Muhammad Zaid, and Muhammad Atif Tahir. 2017. Ensemble of Texture Features for Finding Abnormalities in the Gastro-Intestinal Tract. *Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017)*.

[8] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Proceedings of 31st Conference on Neural Information Processing Systems (NIPS)*.

[9] Stefan Petscharnig and Klaus Schöffmann. 2018. Learning laparoscopic video shot classification for gynecological surgery. *An International Journal of Multimedia Tools and Applications* 77, 7 (2018), 8061–8079.

[10] Stefan Petscharnig, Klaus Schöffmann, and Mathias Lux. 2017. An Inception-like CNN Architecture for GI Disease and Anatomical Landmark Classification. In *Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017)*.

[11] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, Mario Taschwer, Mathias Lux, Peter Thelin Schmidt,

Michael Riegler, and Pål Halvorsen. 2017. Nerthus: A Bowel Preparation Quality Video Dataset. In *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSYS)*. ACM, 170–174.

[12] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, and others. 2017. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSYS)*. ACM, 164–169.

[13] Konstantin Pogorelov, Michael Riegler, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Carsten Griwodz, Peter Thelin Schmidt, and Pål Halvorsen. 2017. Efficient disease detection in gastrointestinal videos–global features versus neural networks. *An International Journal Multimedia Tools and Applications* 76, 21 (2017), 22493–22525.

[14] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Carsten Griwodz, Thomas de Lange, Kristin Ranheim Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, Olga Ostroukhova, and others. 2017. A comparison of deep learning with global features for gastrointestinal disease detection. In *Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017)*.

[15] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico Multimedia Task at MediaEval 2018. In *Working Notes Proceedings of the MediaEval 2018 Workshop*.

[16] Michael Riegler, Konstantin Pogorelov, Sigrun Losada Eskeland, Peter Thelin Schmidt, Zeno Albisser, Dag Johansen, Carsten Griwodz, Pål Halvorsen, and Thomas De Lange. 2017. From annotation to computer-aided diagnosis: Detailed evaluation of a medical multimedia system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 13, 3 (2017), 26.

[17] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Carsten Griwodz, Thomas Lange, Kristin Ranheim Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, Mathias Lux, and others. 2017. Multimedia for medicine: the medico Task at mediaEval 2017. In *Working Notes Proceedings of the MediaEval 2017 Workshop (MediaEval 2017)*.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and others. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* (2015).

A.9 Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

Authors: **Vajira Thambawita**, Debesh Jha, Hugo Lewi Hammer, Håvard D. Johansen, Dag Johansen, Pål Halvorsen, and Michael A. Riegler

Abstract: Precise and efficient automated identification of gastrointestinal (GI) tract diseases can help doctors treat more patients and improve the rate of disease detection and identification. Currently, automatic analysis of diseases in the GI tract is a hot topic in both computer science and medical-related journals. Nevertheless, the evaluation of such an automatic analysis is often incomplete or simply wrong. Algorithms are often only tested on small and biased datasets, and cross-dataset evaluations are rarely performed. A clear understanding of evaluation metrics and machine learning models with cross datasets is crucial to bring research in the field to a new quality level. Toward this goal, we present comprehensive evaluations of five distinct machine learning models using global features and deep neural networks that can classify 16 different key types of GI tract conditions, including pathological findings, anatomical landmarks, polyp removal conditions, and normal findings from images captured by common GI tract examination instruments. In our evaluation, we introduce performance hexagons using six performance metrics, such as recall, precision, specificity, accuracy, F1-score, and the Matthews correlation coefficient to demonstrate how to determine the real capabilities of models rather than evaluating them shallowly. Furthermore, we perform cross-dataset evaluations using different datasets for training and testing. With these cross-dataset evaluations, we demonstrate the challenge of actually building a generalizable model that could be used across different hospitals. Our experiments clearly show that more sophisticated performance metrics and evaluation methods need to be applied to get reliable models rather than depending on evaluations of the splits of the same dataset—that is, the performance metrics should always be interpreted together rather than relying on a single metric.

Appendix A. Published Articles

Published: ACM Transactions on Computing for Healthcare, 2020-2021

Candidate contributions: Vajira is the first author and the corresponding author of this journal paper. He contributed to the main conception and design of the experiments in this manuscript. Vajira developed and analyzed the three different deep neural networks critically in this study. Additionally, he analyzed several Gastrointestinal (GI) tract datasets to use in the experiments and evaluated his models using those cross datasets to measure the generalizability of the deep learning solutions in real-world applications. He contributed to drafting the manuscript and revising it. This journal paper is the extended version of "The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning".

Thesis objectives: Sub-objective I, Sub-objective III

An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

VAJIRA THAMBAWITA, SimulaMet and Oslo Metropolitan University
DEBESH JHA, SimulaMet and UiT—The Arctic University of Norway
HUGO LEWI HAMMER, Oslo Metropolitan University and SimulaMet
HÅVARD D. JOHANSEN and DAG JOHANSEN, UiT—The Arctic University of Norway
PÅL HALVORSEN, SimulaMet and Oslo Metropolitan University
MICHAEL A. RIEGLER, SimulaMet

Precise and efficient automated identification of gastrointestinal (GI) tract diseases can help doctors treat more patients and improve the rate of disease detection and identification. Currently, automatic analysis of diseases in the GI tract is a hot topic in both computer science and medical-related journals. Nevertheless, the evaluation of such an automatic analysis is often incomplete or simply wrong. Algorithms are often only tested on small and biased datasets, and cross-dataset evaluations are rarely performed. A clear understanding of evaluation metrics and machine learning models with cross datasets is crucial to bring research in the field to a new quality level. Toward this goal, we present comprehensive evaluations of five distinct machine learning models using global features and deep neural networks that can classify 16 different key types of GI tract conditions, including pathological findings, anatomical landmarks, polyp removal conditions, and normal findings from images captured by common GI tract examination instruments. In our evaluation, we introduce performance hexagons using six performance metrics, such as recall, precision, specificity, accuracy, F1-score, and the Matthews correlation coefficient to demonstrate how to determine the real capabilities of models rather than evaluating them shallowly. Furthermore, we perform cross-dataset evaluations using different datasets for training and testing. With these cross-dataset evaluations, we demonstrate the challenge of actually building a generalizable model that could be used across different hospitals. Our experiments clearly show that more sophisticated performance metrics and evaluation methods need to be applied to get reliable models rather than depending on evaluations of the splits of the same dataset—that is, the performance metrics should always be interpreted together rather than relying on a single metric.

CCS Concepts: • **Computing methodologies** → **Cross-validation**; **Supervised learning by classification**; **Machine learning approaches**; • **Applied computing** → **Life and medical sciences**;

17

Appendix A. Published Articles

1 INTRODUCTION

Cancer is one of the leading causes of death worldwide and a significant barrier to life expectancy [12]. In particular, the gastrointestinal (GI) tract can be affected by a variety of diseases and abnormalities [52]. Using data from the Global Cancer Observatory,[1] Bray et al. [12] estimated that, for 2018, there would be around 5 million new luminal GI cancer incidences and about 3.6 million deaths due to GI cancer.[2] The most frequently diagnosed GI cancers in 2018 for new cases were colorectal cancer (CRC) (6.1%), stomach cancer (5.7%), liver cancer (4.7%), rectum cancer (3.9%), and esophageal cancer (3.2%) out of 36 types of cancers [12].

Gastroscopy and colonoscopy are the most successful medical procedures for GI endoscopy examinations. Among both, colonoscopy has been proven to be an effective preventative method by improving declination in the occurrence of Colorectal Cancer (CRC) by 30% [41]. During a colonoscopic procedure, an endoscopist inserts a colonoscope carefully through the anus to examine the rectum and colon. A tiny wide-angle video camera mounted at the end of the colonoscope captures a live video signal of the internal mucosa of the patient's colon. The endoscopist uses the video signal for real-time diagnosis of the patient, where one of the primary goals is to identify and remove abnormalities such as polyps [77].

The current EU guidelines [74] recommend GI tract screening for all people older than 50 years. Such regular screenings can be of great significance for early detection and prevention of cancer inside the GI tract, but they are challenging due to many factors. Moreover, a colonoscopy examination is entirely an operator-dependent screening procedure [63]. The detection rate of GI tract lesions mostly relies on the clinical experience of the gastroenterologist. The shortage of experienced gastroenterologists, and the clinicians' tiredness and lack of concentration during the colonoscopic examination, can lead to missing polyps that otherwise would be detected [68]. The estimated miss rate for the subject undergoing a colonoscopy examination is 25% [39].

Although considerable work has been done to develop and improve systems for automatic polyp detection, the performance of existing solutions is still behind that of an expert endoscopist [7, 16, 44, 75, 76]. Most of the published works in the field use non-public datasets or develop models from too-small training, validation, and test sets [7, 75, 76]. The performance metrics used to measure the performance of methods are also not sufficient (e.g., see the first part of Table 1). Thus, it is difficult for researchers to compare and reproduce some of the present related works. Moreover, the state-of-the-art research in this field does not present the generalizability of their solutions using cross-dataset evaluations. As a result, it creates a distrust for applying these machine learning (ML) solutions in practice.

An automatic and efficient computer-aided diagnosis (CAD) system in a clinic could assist medical experts during the endoscopic and colonoscopy procedure to improve the detection rate by finding unrecognized lesions and act as a second observer by providing better insights to the gastroenterologist concerning the presence and types of lesions. With this inspiration, we conducted five experiments to classify 16 classes of GI tract conditions

---

[1]https://gco.iarc.fr.
[2]We have considered the statistic of esophagus, stomach, colon, rectum, anus, gallbladder, and pancreas.

Table 1. Overview of the Related Work

Reference	Year	REC	PREC	SPEC	ACC	MCC	F1	Rk	FPS
Hwang et al. [27]	2007	0.9600	0.8300	—	—	—	—	—	15
Li & Meng [40]	2012	0.8860	—	0.9620	0.9240	—	—	—	—
Zhou et al. [83]	2014	0.7500	—	0.9592	0.9077	—	—	—	—
Wang et al. [76]	2014	0.8140	—	—	—	—	—	—	0.14
Mamonov et al. [43]	2014	0.4700	—	0.9000	—	—	—	—	—
Wang et al. [77]	2015	0.9770	—	—	0.9570	—	—	—	10
Riegler et al. [57]	2016	0.9850	0.9388	0.7250	0.8770	—	—	—	~300
Shin & Balasingham [63]	2017	0.9082	0.9271	0.9176	0.9126	—	—	—	—
Riegler et al. [58]	2017	0.9850	0.9390	0.7250	0.8770	—	—	—	~75
Yu et al. [78]	2017	0.5005	0.4917	—	0.9471	—	0.4830	0.5357	—
Pogorelov et al. [54]	2017	0.8260	0.8290	0.9750	0.9570	—	0.8260	0.8020	46
Agrawal et al. [1]	2017	—	—	—	0.9610	0.8260	0.8470	—	—
Naqvi et al. [45]	2017	—	0.7665	0.9660	0.9420	0.7360	0.7670	—	—
Petscharnig et al. [48]	2017	0.7550	0.7550	0.9650	0.9390	0.7200	0.7550	0.7240	—
Pogorelov et al. [52]	2017	0.9060	0.9060	0.9810	0.9690	—	—	—	30
Yuan et al. [79]	2018	0.8180	0.7232	—	—	—	0.7431	—	—
Wang et al. [75]	2018	0.9438	—	0.9592		—	—	—	—
Mori & Kudo [44]	2018	>0.9000	—	>0.9000	—	—	—	—	—
MediaEval 2018 Medico Task [53] (The following experiments were done using the 2018 Medico dataset.)										
Hoang et al. [25]	2018	0.9281	**0.9426**	**0.9963**	**0.9932**	**0.9312**	**0.9342**	**0.9398**	23
Hicks et al. [24]	2018	0.9218	0.9378	0.9959	0.9924	0.9228	0.9236	0.9325	624
Borgli et al. [10]	2018	0.8572	0.8708	0.9956	0.9918	0.8555	0.8555	0.9280	—
Kirkerød et al. [36]	2018	0.8433	0.8514	0.9944	0.9896	0.8366	0.8367	0.9082	—
Dias & Dias [18]	2018	0.8205	0.8414	0.9938	0.9885	0.8146	0.8114	0.8983	8.61
Taschwer et al. [70]	2018	0.8673	0.8826	0.9933	0.9876	0.8641	0.8662	0.8897	—
Ostroukhova et al. [46]	2018	0.8236	0.8281	0.9911	0.9835	0.8115	0.8145	0.8539	1E-100
Khan & Tahir [33]	2018	0.6203	0.7173	0.9767	0.957	0.6025	0.5868	0.6302	43329
Steiner et al. [64]	2018	0.4219	0.5146	0.9717	0.9469	0.3901	0.3913	0.5368	—
Ko et al. [37]	2018	0.5005	0.4916	0.9715	0.9471	0.4608	0.4829	0.5357	0.5357
Thambawita et al. (Ours) [71]	2018	**0.9361**	0.9319	**0.9963**	**0.9932**	0.9283	0.9297	0.9397	—

REC, recall (sensitivity); ACC, accuracy; MCC, Matthews correlation coefficient; F1, F1-score; Rk, Rk correlation coefficient; FPS, frames per second.

The results of the Medico Task may slightly vary compared to the proceeding note papers because of different ways of calculating the multi-class performance metrics by the organizers. The highest score for the MediaEval 2018 Medico Task is marked in bold.

for the Medico Multimedia Task at MediaEval 2018 [53]. One example for each of the 16 classes is depicted in Figure 1.

In this work, we focus on identifying the limitations of generalizing ML models across different datasets and how to interpret the evaluation metrics in that context. For this, we are using global feature (GF)-based and deep learning (DL)-based methods that performed well at the 2018 Medico Task [53], where one specific dataset was used. In addition, here we explore the different performance metrics of both methods (GF and deep learning (DL)) to identify the limitations of each. We show that combined complex deep neural network (DNN) models outperform other methods. Finally, we explore how multi-class models perform on polyp and non-polyp detection with and without retraining the model for the two specific classes. The effects of retraining for classifying the sub-categories of the same dataset and using them in other datasets are analyzed in detail to identify
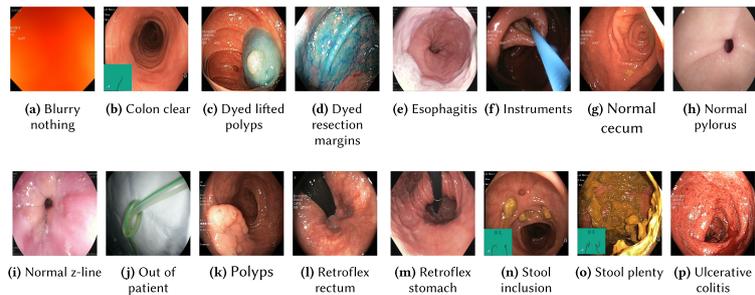
Fig. 1. Sample images of GI findings. Each image represents one of the 16 classes from the dataset used for the Medico 2018 Challenge [50, 51].

the cross-dataset generalization capabilities of our models. We emphasize that a large number of performance measures do not show the real performance of ML models. We also highlight the necessity of having cross-dataset evaluations to determine the real capabilities of ML models before using them in clinical settings.

To study cross-dataset bias and metrics interpretation, our contributions are as follows:

(1) We present five ML classification models to classify multi-class findings (anatomical landmark, pathological findings, polyp removal conditions, and normal findings) of the GI tract. Using a limited imbalanced dataset, we experiment with approaches ranging from Global Feature (GF) approaches to simple Deep Neural Network (DNN) and complex DNN approaches with transfer learning. Moreover, we present a detailed evaluation using six performance metrics to show the real classification performance of ML models. In addition, we analyze and present detailed evaluation results of using multi-class classification ML models for classifying binary classes (sub-classes of the multi-class categories) with and without retraining to evaluate the generalizability of our models. We emphasize the difficulties of using well-performing ML methods in cross-datasets as a result of the reluctance of ML models to cross-dataset generalization. We present this negative impact with the aid of another evaluation using the receiver operating characteristic (ROC) curve and the precision-recall (PR) curve of the best model. We also demonstrate when a Receiver Operating Characteristic (ROC) curve is good to use and when it is better to use a PR curve.

(2) With the preceding point, we emphasize the requirement of detailed cross-dataset evaluations to identify generalizability of ML models before using them as universal models in live applications. Because good performance measures with a single dataset do not necessarily imply good real-world performance, we argue that researchers should present cross-dataset evaluations for building a generalizable model rather than presenting performance values for the test datasets, which is separated from the same training data source.

Moreover, with respect to the 2018 Medico Task [53], our best DNN method achieved the highest recall, specificity, and accuracy for multi-class classification of the GI tract findings. We achieved a Matthews correlation coefficient (MCC) (0.0029 less) and an Rk correlation coefficient[3] (0.0001 less) nearly equal to the winning team. With this achievement, we demonstrate all of the steps, from designing to training and testing, for reaching such performance using this model and its expandability using different pre-trained networks.

---

[3]The Rk correlation coefficient and the MCC were the most important considered metrics for winning the 2018 Medico Task.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

In Section 2, we present related work and the performance of relevant existing solutions. Section 3 discusses the methodology used for our GF-based approaches and the theoretical foundation for our work. The DNN-based approaches are similarly described in Section 4. Our experimental results are presented and analyzed in Section 5, followed by a discussion in Section 6 on how our results can be helpful to other researchers. In Section 7, we conclude our findings.

2 RELATED WORK

Many methods and algorithms have been proposed for GI tract disease detection/classification using videos and images from colonoscopy and gastroscopy as input. The problem of polyp detection has by far received the most attention by researchers. Images and videos of polyps and other abnormalities inside the GI tract are usually collected using a specific-purpose camera and imaging system, like ScopeGuide from Olympus. The information gathered from these types of devices may be of great significance for later examination and must be handled with great care. Polyps generally have characteristics different from the normal surrounding healthy tissue and are often easy for clinicians to detect. There are several good datasets available for training and testing on polyps (the details about the available polyp dataset can be found in other works [16, 30]), and binary classification methods are relatively straightforward to implement.

The other active research efforts include developing an automatic and real-time detection system for GI bleeding, ulcerative lesion, blood-based abnormality, tumor, and angiectasia, and for multi-class data of the GI tract that comprise anatomical landmarks (e.g., z-line, pylorus, and cecum), pathological findings (e.g., esophagitis and ulcerative colitis), and normality and regular findings (e.g., normal colon mucosa and stool). Suitable datasets for research in these areas are less developed and lack adequate content. Similarly, presented performance measures in these areas are not adequate because of not presenting enough performance metrics or not presenting cross-dataset evaluations.

Table 1 presents an overview of important works related to GI disease detection/classification and the 2018 Medico Task [53] using Computer-Aided Diagnosis (CAD), from automatic polyp detection to multi-class disease detection and classification systems. The dataset used for the experiments in the first half of the Table 1 is different. Therefore, the results cannot be directly compared; however, the results in the lower half of the table can be compared, as the algorithms are tested on the same dataset.

Most of the research in the medical field only focuses on designing an automated disease detection system for detecting or classifying specific disease or abnormality, such as polyp detection or ulcer detection. Because patients may suffer from more than one type of disease at the time, a working multi-class disease detection system will help treatment. The performance of existing multi-abnormality detection systems is, however, not satisfactory and cannot assist doctors in CAD in real time while undergoing colonoscopies. Furthermore, these research works have not evaluated all performance metrics at once to analyze the real behavior of their classification models. Yet none of the preceding methods have performed cross-dataset evaluations to prove the capabilities for using the ML models in real CAD systems.

For handcrafted (HC) feature-based methods, image descriptors like global or local image features (e.g., color, texture, and edges) are extracted, and later on, various ML classifiers (e.g., logistic model tree (LMT) [71], random forest classifier [43], or support vector machine (SVM) [76]) are employed to perform analysis using these features. HC descriptors (manually designed features) are useful for the gastroenterologist while identifying specific abnormality regions inside the GI tract. For instance, as blood has a particular range of chromaticity, we can specify a specific chromaticity range where features of bleeding abnormality seem to be concentrated [31]. Riegler et al. [58] achieved an F1-score of 0.909 with a GF-based approach and an F1-score of 0.875 with a DL-based approach with a multi-class GI tract dataset. With the ASU-Mayo polyp dataset, the GF-based approach achieves an F1-score of 0.961, whereas the DL-based approach could obtain 0.936. They further suggested that the combination of both approaches may lead to improved performance. In addition, previous work by Riegler

213

Appendix A. Published Articles

et al. [56] reveals that although only detecting whether a frame contains an irregularity or not, GFs can beat local features—for instance, they can at least reach the same results with regard to detection/classification and perform better than local features with regard to processing speed. In all of these works, researchers presented performance metrics using a test dataset selected from the same dataset used for the training data. Therefore, these results do not reflect the actual practical performance of the proposed methods.

A few past studies used information such as the color and texture of polyps to sketch HC descriptors [2, 3, 13, 28, 29, 32, 68]. The other category of methods for automated polyp detection used shape, intensity, edge, and spatio-temporal information. For instance, Hwang et al. [27] appropriated elliptical shape features to detect the occurrence of polyps in colonoscopy videos. Bernal et al. [7] proposed a polyp detection technique by utilizing a polyp region descriptor, which is dependent on the depth of the valley image and introduced a region growing method to detect polyps in colonoscopy images. Bernal et al. [8] additionally used valley information and enhanced their approach by improving the polyp localization results to almost 30%. Bernal et al. [6] also performed additional evaluations using valley information and demonstrated better performance, especially for smaller polyps and decreased the polyp miss rate. Park et al. [47] utilized spatio-temporal features for automatic polyp detection. The recently completed related work that uses the cross-sectional profile to detect protruding polyps automatically is the polyp detection system Polyp-Alert [77], which can provide near real-time feedback during colonoscopies. However, the system is limited to polyp detection and is slow for live examinations. Tajbakhsh et al. [68] proposed a method for automatic polyp detection from colonoscopy videos that uses context information to remove non-polyp and shape information to localize polyp reliably. Riegler et al. [58] utilized various GFs and achieved high precision and recall above 90%. Yuan et al. [79] employed a bottom-up and top-down saliency approach for automated polyp detection. Although these research works discuss improving the performance of ML models, they have not evaluated the performance of the ML models with cross-datasets.

As convolutional neural network (CNN) architectures have achieved exceptional gains in medical image and video analysis tasks, more recent work on polyp detection is mainly based on Convolutional Neural Networks (CNNs). Tajbakhsh et al. [67] proposed a 2D-CNN method for polyp detection by learning discriminative spatial and temporal features. Yu et al. [78] proposed a 3D fully convolutional network to deal with the challenges related to automatic polyp detection for colonoscopy videos. Zhang et al. [81] suggested an enhanced single-shot multi-box detector (SSD) called *SSD-GPNet* for detecting gastric polyps, which have the potential for achieving real-time detection up to 50 FPS using Nvidia Titan V. Furthermore, they use GPDNet [82] to classify three classes of pre-cancerous gastric disease.

Researchers have also compared HC and DL methods. For instance, Pogorelov et al. [52] and Riegler et al. [58] compared several (HC- and DL-based) localization methods. Pogorelov et al. [49] evaluated their approach utilizing HC and DL methods on different available datasets for real-time polyp detection. Their best model with a generative adversarial network (GAN) obtained detection specificity of 94% and accuracy of 90.9%. The preceding research works presented good performance for predicting polyps, whereas Pogorelov et al. [49] presented evaluation results of the models with cross-datasets. However, having overlapped data sources in the cross-datasets, the shown results do not reveal the real performance in cross-dataset evaluations.

The pre-trained models, along with transfer learning mechanisms, are also becoming popular because of their capability to outperform state-of-the-art algorithms even with less training data, where the limited size of the medical dataset for experiments has always been a problem to yield better results. For the detection and localization of the polyps [9, 69], the pre-trained models with a CNN mechanism also achieve promising results. A comparison of DL with GFs for GI tract disease detection has also been presented. Pogorelov et al. [54] presented 17 different methods for multi-class classification of GI tract data with the limited number of the training dataset. They used both GFs and DL approaches in their work. They achieved the best result with modified ResNet50 features using the LMT classifier. They reached an Rk value of 80.2% and an F1-score of 82.6% with 2,000 training and 2,000 test datasets.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

Comparing with the polyp detection approaches, the research on multi-class disease detection/classification on a complete GI tract system is minimal. However, for multi-class disease detection/classification (including polyp detection) inside the GI tract, we note a few contributions made in this area. For example, the authors of numerous works [1, 10, 18, 24, 25, 33, 36, 37, 46, 48, 64, 70] presented their approach in classifying disease inside the GI tract utilizing the Kvasir dataset and the MediaEval Medico 2018 dataset. The latter is a combination of the Nerthus [50] and Kvasir [51] datasets.

Hicks et al. [24] show how fine tuning a CNN model using transfer learning with data from different source domains affects classification performance. In their case, extending the generic ImageNet dataset with medical images from the LapGyn4 and Cataract-101 dataset, they obtained a high Matthews Correlation Coefficient (MCC) score of 0.9228. For the 2018 Medico Task, we proposed solutions based on GFs and DL-based methods for multi-class classification of GI tract findings [71]. Our best model was a combination of two pre-trained networks, ResNet-152 and Densenet-161, along with a multi-layer perceptron (MLP). Here, we obtained an MCC of 94.21%, an F1-score of 94.58%, and an accuracy of 99.32%. This was one of the best results in the MediEval 2018 Medico Task Challenge. We discuss the model introduced by Thambawita et al. [71] in detail in this article and reproduce similar results. Based on those models, we provide and discuss the requirement of detailed evaluations using multiple performance metrics and cross-dataset evaluations.

Recent related works show promising results in terms of evaluation metrics, such as both sensitivity and specificity despite various challenges (e.g., difficulties arise due to a dataset obtained from different modalities). The limitation with most of the recent approaches is that they target only specific problems, like bleeding detection or polyp detection. Current systems are either (i) too narrow for a flexible, multi-disease detection/classification system; (ii) tested only on a limited datasets, too small to show whether the systems would work well in hospitals, (iii) provide low processing performance for a real-time system or ignore the system performance entirely; (iv) problematic with regard to overfitting of the specific dataset and lead to unreliable results; or (v) tested using datasets that are not publicly available, making it difficult to compare the approaches with others.

In some cases, GF-based approaches produce better results. For some methods, DL performs better. The CNN approaches and pre-trained network with transfer learning mechanism approaches have the best results in most of the cases. Reusing already existing DL architectures and pre-trained models leads to excellent results in, for example, the ImageNet classification tasks. For example, the HC feature-based approach works well for true negative (TN) detection/classification tasks.

To reduce the damage of the dataset bias problem, Khosla et al. [34] directed their experiments for both classification tasks and detection problems. They used different datasets from different domains in the training stage to generalize the features extracted from their ML model. However, SVM was used as the main algorithm, and the DNN dataset bias problem was not addressed.

With the goal of making researchers aware of the dataset bias problems, Torralba and Efros [72] did informative research using basic datasets and basic ML models with the classification and detection task of computer vision. Initially, they trained a simple linear Support Vector Machine (SVM) to make a simple classifier to name a given dataset from 12 different datasets, which have nearly the same categories. They were inspired by the research done by Dollár et al. [20] to detect pedestrians. The result of the experiment for dataset classification shows a clear diagonal in the confusion matrix (CM). This implies that there are clear dataset bias features, and that these datasets have the same categories. Therefore, researchers want to apply cross-dataset generalization for avoiding dataset bias behavior of ML models. Moreover, they discussed selection bias, capture bias, category or label bias, and negative bias as the main factors for the dataset bias. This directs our research to do additional experiments to identify the significant factors of the cross-domain data generalization in the medical domain, which is more critical than the general image classification.

The classification of GI diseases is more complicated than a simple real-world object classification task where one detects faces or recognizes characters. Typical GI tract datasets are heavily imbalanced—for example, the 2018 Medico Task dataset consists of 16 classes of anatomical landmarks, pathological findings, polyp removal
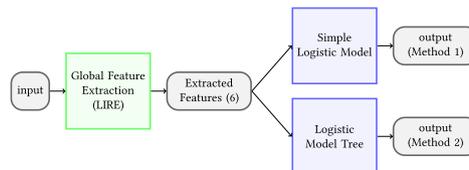
Fig. 2. Block diagram of the proposed method 1 and method 2. The pipeline starts with the input of images. GFs are extracted using the LIRE framework. These features are then used for two different classification algorithms (the SL model for method 1 and LMT for method 2).

cases, and normal and regular findings, where the polyp class has a maximum of 613 images, and the instrument class has a minimum of only 4 images. Additionally, medical datasets are captured using different endoscopic instruments, and some of the images can be noisy, blurry, over- or under-exposed, and interleaved, and can have superfluous information within the image, contain borders, and be affected by specular reflections caused by the instrument light source. Some of the images may have bleeding, whereas other images can be partially covered by stool or mucus. Moreover, the organs from mouth to anus can have multiple lesions showing different diseases, abnormalities, and internal injuries. Thus, the preceding situation leads to the necessity of distinguishing between various classes of GI tract findings. In this scenario, not only high precision and recall but also high accuracy and MCC become essential for developing an automated generalizable multi-class classification system. This implies the real requirement of measuring and analyzing all performance metrics at once. Furthermore, to prove the generalizability of models, cross-dataset evaluations are required.

3 GF-BASED APPROACHES

GFs or descriptors are features computed over the whole image or covering a regular sub-section of an image. GFs represent the overall properties of an image and are often used in image retrieval, image compression, image classification, object detection, and image collection search and distance computing [54]. Examples of GFs are shape matrix, histogram-oriented gradients (HOGs), Co-HOG, and invariant moments (Hu, Zernike). The LIRE [42] framework can be used to extract HC GFs such as texture, color distribution, and the histogram of brightness. The most commonly used GFs include joint composite descriptor (JCD), Tamura, color layout (CL), edge histogram (EH), autocolor correlogram, pyramid histogram of oriented gradients (PHOGs), color and edge directivity descriptor (CEDD, local binary patterns, and scalable color (SC). Figure 2 shows the architecture of the proposed GF-based methods (1 and 2). These methods use six selected GFs and the best ML classifiers for the provided dataset.

Feature engineering is among the most crucial and challenging parts for approaching any ML and computer vision problem. Based on the findings of Pogorelov et al. [54] and Riegler et al. [59], we choose to used JCD, Tamura, CL, EH, autocolor correlogram, and PHOG. The combinations of these features represent the overall properties of the images. We can even add more GFs, but doing so may increase the noise to the image features, which again would hurt the classification performance. Moreover, we have formulated the problem of GI tract anomaly classification as a multi-class (16-class) classification of different findings including anomalies, landmarks, and clinical markings. With the provided dataset, we computed the GFs of each image. A multi-class classification problem is a general and well-studied ML problem, and there is a variety of methods available to solve this issue with higher performance. Therefore, we sent the extracted GFs to many available ML classifiers. The whole experiment was completed with the development dataset. The 2018 Medico Task [53] shows the best classification rates with SimpleLogistic (SL) [38] and LMT [38] classifiers.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics
Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality
Classification

3.1   Method 1: The SL Classifier

In method 1, we combine the SL classifier from the Weka software [22] to build a linear logistic regression (LR) model with the LogitBoost [21] utility for determining attributes. The SimpleLogistic (SL) classifier can deal with binary class classification, multi-class classification, missing class, and nominal class. It can handle different types of attributes, such as binary attributes, nominal attributes, date attributes, missing values, unary attributes, and empty nominal attributes [38]. In a linear LR classifier, a simple (linear) model fits the data, and the method of model fitting is pretty stable, leading to low variances.

LogitBoost is utilized for determination of the most appropriate attributes in the data at the time of executing LR, which is done by performing a simple regression in every iteration before it converges to a solution of maximum likelihood. Therefore, LogitBoost, with a simple regression function that acts as a base learner, is utilized for fitting the logistic models. The optimum number of iterations associated with the LogitBoost algorithm to function is cross validated, which leads to the automatic selection of the attribute [65]. The SL classifier has a built-in attribute selection (if the default parameter is not changed): it stops computing simple linear regression models (i.e., performing LogitBoost iterations) when the cross-validated classification error no longer decreases. With the extracted features using LIRE, the SL classifier has not only the highest classification accuracy but also takes the lowest classification time (i.e., lowest computational complexity) when compared with other ML classification algorithms.

3.2   Method 2: The LMT

In method 2, we use the Logistic Model Tree (LMT) classifier from the Weka software. The LMT is a classification model related to a supervised training algorithm, which is a combination of LR and decision tree learning techniques [38, 62]. Thus, the LMT is considered an analogue model for solving classification problems. In the logistic variant, information gain is utilized for splitting, the LogitBoost algorithm generates an LR model at each node in the tree, and the CART algorithm [62] is utilized for pruning the tree.

The LMT uses a cross-validation (CV) technique to find several LogitBoost iterations to prevent overfitting of the training data. The LogitBoost algorithm accomplishes additive LR, which is achieved by least-square fits for every class M [19], which is shown in Equation (1):

$$L_M(x) = \sum_{i=1}^{n} \beta_i + \beta_0. \tag{1}$$

Here, β_i denotes the coefficient of the ith component of the vector x, and n denotes number of features. The LMT model uses the linear LR method to calculate the posterior probabilities of the leaf nodes [38], which is shown in Equation (2):

$$L_M(X) = -\frac{\exp(L_M(X))}{\sum_{M=1}^{D} \exp(L_M(X))}. \tag{2}$$

Here, D denotes the number of classes, and $L_M(X)$ stands for the least-square fits. The least-square fits $L_M(X)$ are transformed in such a way that $\sum_{M=1}^{D} \exp(L_M(X))$ is equal to zero.

4   DL APPROACHES

For our transfer learning approaches, we selected two DNNs: ResNet-152 [23] and DenseNet-161 [26] based on the top-1 error rate and top-5 error rate for the ImageNet [17, 61] classification as given in the PyTorch documentation [14]. Then, we chose ResNet-152 as the base model of the first DL approach, and this base model experiment was done under method 3 (the model is illustrated in Figure 3). This selection was made based on preliminary experiments. In the preliminary experiments, ResNet-152 showed better performance than DenseNet-161. This DenseNet-161 was in second place in the performance ranking when we compared stand-alone pre-trained DL models.
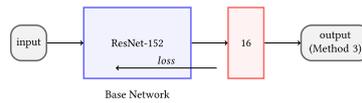
Appendix A. Published Articles

Fig. 3. Block diagram of method 3. The input is an image that is passed to a ResNet-152 neural network. A final softmax layer outputs the scores for the 16 classes.
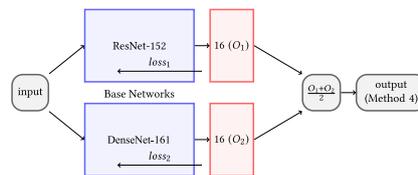

Fig. 4. Block diagram of method 4. The input image is in parallel passed to a ResNet-152 and a DenseNet-161 neural network. Two separate softmax layers calculate separate 16-class scores, which are finally combined.
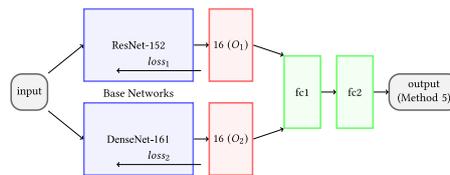

Fig. 5. Block diagram of method 5. It is similar to method 4, but instead of a single step to combine the output scores of the two neural networks, two fully connected layers are utilized.

In DL methods 4 and 5 (as illustrated in Figures 4 and 5), we used both pre-trained ResNet-152 and DenseNet-161 using the ImageNet dataset. In the following sections, we discuss data pre-processing mechanisms and training mechanisms used for all three DL methods. In later sections, we discuss these methods one by one with their fine-tuning mechanisms with more comprehensive explanations.

For the transfer learning methods, we use the data pre-processing tool of the PyTorch library to (i) resize input images, (ii) crop marginal annotations of the medical images, (iii) normalize the pixel values of input images, and (iv) apply random image transformations. Regarding image resizing, all images of the dataset were resized into 224×224 because ResNet-152 and DenseNet-161 accept images with these dimensions. By applying the central-cropping transformation of PyTorch, we minimized unnecessary effects for the final predictions of DNNs affected from annotated marks (green boxes) of the medical images as shown in Figure 1(b), (n), and (o). Center cropping did not remove important information from the images because we cropped down to 224×224 from 256×256. Our experiments show that removing the whole green box, such as those in Figure 1(b), (n), and (o), from the images by applying a larger crop size is not advisable, because for some images, too much content of the finding is lost with a large crop size. When applying the normalization function to the input images, a standard deviation (σ) of 0.5 and a mean (μ) of 0.5 were used with the normalization function in PyTorch. The mathematical equation used in this function is given in Equation (3), and c represents the three channels R, G, and B of input images. The *input* represents a tensor of pixel values of each layer. We used random transformations, random horizontal

flips, random vertical flips, and random rotations from PyTorch as data augmentation techniques.

$$input_c = \frac{input_c - \mu_c}{\sigma_c}; \quad where \quad c = [0, 1, 2] \tag{3}$$

For training all DNNs, the transfer learning mechanism was used. Then, we used cross-entropy loss [15] with weighted classes as given in Equation (4) to calculate the loss values of the DNNs:

$$loss(x, class) = weight[class] \times \left(-x[class] + \ln\left(\sum_j \exp(x[j]) \right) \right). \tag{4}$$

In this equation, the weight parameter value is calculated inversely proportional to the image count in the corresponding class. In other words, class weight values are high when the classes have fewer images. However, the inbuilt cross-entropy function given in PyTorch is used instead of implementing it from scratch. While doing preliminary experiments, we observed that there was not any effect from weighted cross-entropy loss. Then, we used the normal cross-entropy loss (Equation (5)) function for calculating the loss of the DNNs:

$$loss(x, class) = -\ln\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])} \right) = -x[class] + \ln\left(\sum_j \exp(x[j]) \right). \tag{5}$$

As the optimizer of all DNNs, the stochastic gradient descent (SGD) [11] method with a momentum [66] was applied. We selected this optimizer because of its stable learning mechanism in contrast to the highly unstable learning pattern of other methods [35, 60, 80], as they show fast convergence.

During the training procedure, we changed the learning rate manually based on the progress of learning curves rather than using the inbuilt learning rate schedulers of PyTorch. Initially, we began with a high learning rate. Then, the learning rate was reduced by a factor of 10 if the training process did not show good progress in the learning curves. Finally, model weights of the best epoch based on the best validation accuracy were saved to use in the inference stage.

4.1 Method 3: DNN Approach Based on ResNet-152

Method 3 is the base method that uses only ResNet-152. A block diagram of this is illustrated in Figure 3. In this method, the last layer of ResNet-152 is modified to output 16 classes of the 2018 Medico Task from 1,000 classes of ImageNet. Usually, we freeze first layers (there is not a logical way to select the number of layers to freeze) of pre-trained networks when we do transfer learning. Then, we train the last and the new layers using the new domain data. Finally, the entire network is trained after unfreezing all parameters of the network (a method known as fine tuning).

We performed preliminary experiments to identify the influence of the preceding freezing-unfreezing technique compared to using simple fine tuning. Both techniques showed the same performance at the end of the training process, and we could not gain any performance benefit from the freezing-unfreezing method, as using the simple fine-tuning method was faster. Therefore, we decided to use the simple fine-tuning method for all experiments.

In method 3, we started the training process with a learning rate of 0.001. Then, the learning rate was decreased by a factor of 10 if we could not see any performance improvement for the validation dataset. We repeated this change of learning rate until the model came to a good stable position. In this experiment, the SGD method was used as the optimization method with a momentum of 0.9.

4.2 Method 4: DNN Approach Based on ResNet-152 and DenseNet-161

In method 4, as illustrated in Figure 4, we used two pre-trained networks on ImageNet: ResNet-152 and DenseNet-161. These networks were retrained separately into the Medico dataset using the same procedure used in
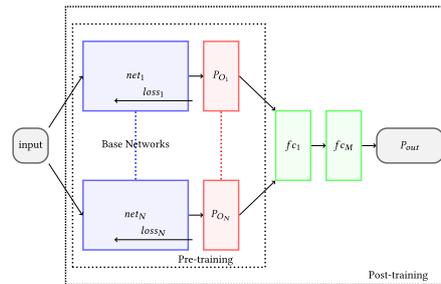
Fig. 6. Block diagram of the proposed parallel DNN merging. The training process is split into a pre-training (pre-training of individual models) and post-training step (training the whole network architecture).

method 3. Before this retraining, the networks were modified to classify the 16 classes. Then, we calculated an average probability of the two probability vectors (V_{Resnet_152} and $V_{Densenet\_161}$) output by the two separate networks: ResNet-152 and DenseNet-161. By calculating the average of these two probability vectors ($V_{answer} = \frac{1}{2} (V_{Resnet\_152} + V_{Densenet\_161})$), we accepted the cumulative probability decision rather than the individual decision. Using the average from these two networks, we expected to have a good decision with high confidence. For example, if the two networks return high probability values for the same class, the class probability value (confidence of classifying to that class) is high. However, when one network has a high probability and the other network has a low probability for a specific class, then the final probability value is around 0.5. This value infers that confidence about the particular class is not good enough for the final decision.

In this model, the probability of the final answer depends on the average values rather than the highest probability value returned from one of the two models. Here, the problem is that the prediction suggested from the highest probability value of one model may be the correct class compared to the selected category from the average. Finally, we trained the model using a learning rate of 0.001. In addition, we decreased the learning rate by a factor of 10 when the model did not show convergence. A momentum of 0.9 with SGD was used as in method 3.

4.3 Method 5: DNN Approach Based on ResNet-152, DenseNet-161, and MLP

Method 5 was designed to overcome the problem of method 4. The block diagram of this method is illustrated in Figure 5. The simple averaging method was not enough to make a final decision when the two networks provided two different answers. As a solution, an Multi-Layer Perceptron (MLP) was introduced instead of the simple averaging method. Then, we trained only this MLP with the pre-trained ResNet-152 and DenseNet-161 for the Medico dataset to decide the final prediction based on the probabilities that come from two networks. More details about designing this complex model are discussed in Sections 4.3.1 and 4.3.2.

4.3.1 Extendable Method 5. In this section, we show how we can improve accuracy using multiple cumulative probabilistic decisions by extending method 5 into $N \geq 2$ DNNs. In this general model, as illustrated in Figure 6, we divide the whole training process into the following four steps: (1) pre-training of individual models, (2) model selection for merging, (3) merging models with an MLP, and (4) post-training and fine tuning. Let $NETS = \{net_1, net_2, \ldots, net_N\}$ be the set of pre-trainer networks using the ImageNet dataset and P_{O_i} be the returned probability vector for model net_i.

In step 1 (pretraining), we train each DNN $net_i \in NETS$ as much as possible using the transfer learning mechanism until it gives the best predictions as described in method 3 (using different loss functions; $loss_1$ to $loss_N$).

The DNNs have their unique prediction capabilities within the given classification problem. Then, we analyze the CM of the best outcome of each DNN.

In step 2 (selection), we select networks that give different diagonals of CMs (the diagonal of a CM represents correct classifications) compared to other CMs of selected DNNs. If the diagonal of CM of network $net_i = CM_i$, then we select networks that have $CM_i \neq CM_j; j = [1, 2, \ldots, i - 1, i + 1, \ldots, N]$. The goal of this comparison is to identify DNN models that have different classification performances compared to each other. Equal diagonals of CMs do not imply that the networks are identical for their classifications, because there might be models that give the same diagonal numbers but lead to different classifications for a given image. If the case of equal diagonals occurs, we have to compare correctly classified images to identify the differences. The number of DNNs selected for the final training may or may not be equal to the initial number of pre-trained DNNs depending on similarities in some of the CMs.

In step 3 (merging), we use an MLP to merge all outputs of the selected DNNs. The MLP consists of M layers that take $\sum_{i=1}^{N} length\_of(P_{O_i})$ number of inputs and output P_{out} probability vector according to the given classification problem. Then, step 4 can be started by freezing all the pre-trained DNNs and training only the new MLP until it shows a good validation performance. Optionally, we can retrain the whole model without freezing any layer if we cannot achieve a performance improvement by training only the new MLP.

4.3.2 Method 5 Used by This Research Work. According to the procedure discussed in Section 4.3.1, our implementations of method 5 were designed using two parallel networks ($N = 2$): ResNet-152 and DenseNet-161. Then, we analyzed two CMs, which came from ResNet-152 and DenseNet-161. These two networks were pre-trained according to the given classification problem. Because $CM_{Resnet\_152} \neq CM_{Densenet\_161}$, we combined the two networks with an MLP. This comparison of CMs was done visually using colormaps. However, if the visual inspection of CMs is hard, mathematical operations can be used. Moreover, if the CMs are equal completely, a manual inspection of the classified images is required to identify the differences of model classifications. After combining, we froze two DNNs to proceed to the post-training step. In our experiments, the input layer of the MLP consisted of 32 input nodes. The output of the MLP was a probability vector with 16 values, which is equal to the number of classes of the Medico dataset. We used two fully connected layers, with 32 neurons and 16 neurons. In the post-training step, we started training only the MLP with a learning rate of 0.01. To do the post-training, multi-class cross-entropy loss and Stochastic Gradient Descent (SGD) were used.

5 RESULTS

In this section, we discuss the experimental setup, datasets, and results obtained from our experiments. Using these presented results, we emphasize that high scores for performance metrics do not always show the actual performance of ML methods. To show this, we present well-performing ML models that achieved good results for their performance values. Using cross-dataset testing, we present a detailed analysis of evaluation metrics to emphasize that they are not always representative to identify the real performance of models.

For all experiments, we used the same hardware platform with an Intel Core i7 eighth-generation processor with 16 GB of DDR4 RAM and an 8-GB NVIDIA GeForce 1080 GPU. However, we practiced two different software frameworks for implementing our methods. To implement the GF-based methods (1 and 2), we used the Weka framework [22]. We used the PyTorch framework for the DNN-based methods (3, 4, and 5).

5.1 Datasets

For the work performed in this article, we used the following four datasets: the 2018 Medico dataset [55], CVC-356-plus (a modified version of CVC-356 [6, 7, 73]), CVC-612-plus (a modified version of CVC-612 [6, 7, 73]), and CVC-12k [4, 5]. The training and testing datasets of the 2018 Medico Task were derived from the Kvasir dataset [51] and Nerthus dataset [50], consisting of 16 classes as shown in Table 2. These images consist of different anatomical landmarks (z-line, pylorus, cecum), pathological findings (esophagitis, polyps, ulcerative

Table 2. Summary of the 2018 Medico Dataset

Type	Images in the Development Set (#)	Images in the Test Set (#)
Blurry-nothing	176	39
Colon-clear	267	1,070
Dyed-lifted-polyps	457	590
Dyed-resection-margins	416	583
Esophagitis	444	483
Instruments	36	165
Normal-cecum	416	604
Normal-pylorus	439	569
Normal-z-line	437	636
Out-of-patient	4	6
Polyps	613	423
Retroflex-rectum	237	194
Retroflex-stomach	398	399
Stool-inclusions	130	508
Stool-plenty	366	1,920
Ulcerative-colitis	457	551

The first column shows the names of the different findings. The second and third columns show the number of images in the development and test sets.

Table 3. Overview of the Datasets Used for Our Experiments

Dataset	Training	Testing	Images (#)	Polyps (#)	Non-Polyps (#)
2018 Medico—Development	X	—	5,906	613	5,293
2018 Medico—Testing	—	X	8,740	423	8,317
CVC-356-plus	X	X	2,285	356	1,929*
CVC-612-plus	X	X	1,316	612	704
CVC-12k	—	X	11,954	10,025	1,929

*We replaced this image set with a new image set (with 1,171 images) extracted from a clear colon video collected from the Bærum Hospital, Norway, in the second stage of this research to avoid the overlap between the training data and the testing data.

In total, we have five different datasets, but the Medico dataset is split into a development part and a test part for the challenge. The training and testing columns indicate how the dataset was used in the experiments. Polyps and non-polyps indicate the number of findings. Medico and CVC-356-plus represent a bias toward non-findings. CVC-612-plus is a quite balanced dataset, and CVC-12k presents a bias toward findings. Datasets were chosen based on these distributions to represent common cases in medical imaging datasets.

colitis), endoscopic polyp removal cases (dyed and lifted polyp, dyed resection margin), and normal findings (normal colon mucosa, stool) in the GI tract. The dataset also contains images with different degrees of the Boston Bowel Preparation Scale (BBPS), ranging from 0 to 3. Some of the original images contain the endoscope position marking probe. These are seen as a small green box located in the bottom corners, showing its configuration and location of the image frame. The images used in the study were captured using an electromagnetic imaging system (Scopeguide, Olympus, Europe) [51]. In Table 3, we present a summary of the uses of the 2018 Medico dataset and other datasets for polyp and non-polyp classifications.

The Medico development dataset was used to train our ML models in the first stage. However, this dataset consists of a highly imbalanced number of images, as summarized in Table 2. Within this, the out-of-patient class had only 4 images to train our models. Therefore, only in the first stage, we used an additional 30 images
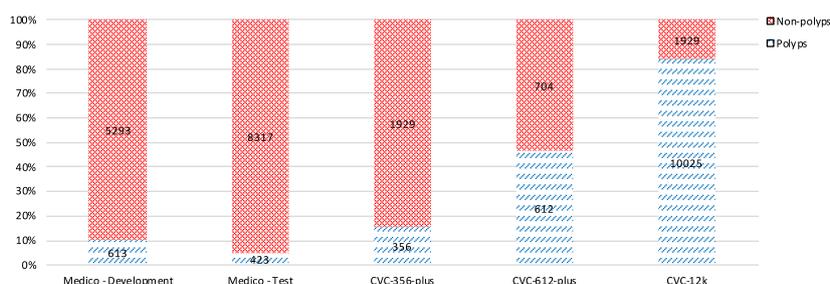
Fig. 7. Ratios of findings to non-findings in the datasets (polyp/non-polyp). The *X* axis represents the different datasets used for the binary classification. The *Y* axis represents the percentage of polyps and non-polyps. The numbers inside the bars show the actual number of polyp and non-polyp images.

that were selected randomly from the Internet to fill this class in the training dataset. These were images of flowers, vehicles, and other general stuff in our everyday life and did not have any relationship with this class. The advantage of this technique is discussed in the discussion of Section 6.

When we discussed the ML models' generalizability in the second part of the article, we used the CVC datasets to retrain and test our models. The CVC-356-plus dataset is the modified version of the CVC-356 [6, 7, 73] dataset that has only polyp images. In that modification, we added 1,929 non-polyp images from the CVC-12k [4, 5] dataset to the CVC-356 dataset and created a new dataset called *CVC-356-plus*. Similarly, the CVC-612-plus dataset was created by extending the CVC-612 dataset [6, 7, 73]. For this CVC-612-plus dataset, we added 704 non-polyp images extracted from new GI tract videos collected by the Bærum Hospital, which is part of the Vestre Viken Hospital Trust in Norway. The content of the CVC-12k dataset underwent a minor reorganization by filtering and grouping polyp and non-polyp images into two separate folders. However, the content and number of images in CVC-12k were not otherwise changed. Therefore, we refer to it by its common name.

In the second part of our research, we used the CVC-356-plus and CVC-612-plus datasets for retraining our models to classify polyps and non-polyps. In only this part of the research, we replaced 1,929 non-polyp images of the CVC-356-plus dataset with 1,171 newly extracted images from a clean and healthy colon video collected from the same hospital. We did this modification to avoid the overlap between the non-polyp images of the CVC-356-plus training dataset and the CVC-12k testing dataset.

For the dataset preparation stage, we focused on the number of polyp and non-polyp images in each dataset to analyze the correlation between the data distribution and the model performance. A bar graph of this data distribution is illustrated in Figure 7. We chose to include different proportions for the number of polyps and non-polyps to keep diversity of data percentages in each test case. In the CVC-356-plus dataset, the polyp percentage is low compared to the non-polyp percentage. In the CVC-612-plus dataset, percentages of polyps and non-polyps are around 50%. In contrast, the CVC-12k dataset has a higher polyp percentage than the non-polyp percentage. Due to this, we can study the effects of data imbalance in the training and testing datasets on the performance and interpretability of the metrics.

5.2 Analyzing Results

We discuss our results in two main sections: (i) the 16-class classification task based on the 2018 Medico Task and (ii) the polyp and non-polyp classification task to analyze generalizability of ML models.

Table 4. Evaluation Results of the 2018 Medico Task (as Provided by the Organizers of the 2018 Medico Task) [71] for the Five Methods Used in This Article

Method	REC	PREC	SPEC	ACC	MCC	F1
1	0.8457	0.8457	0.9897	0.9807	0.8353	0.8456
2	0.8457	0.8457	0.9897	0.9807	0.8350	0.8457
3	0.9376	0.9376	0.9958	0.9922	0.9335	0.9376
4	0.9400	0.9400	0.9960	0.9925	0.9360	0.9400
5	0.9458	0.9458	0.9964	0.9932	0.9421	0.9458

Based on the official results, method 5 was the best one based on the MCC score.
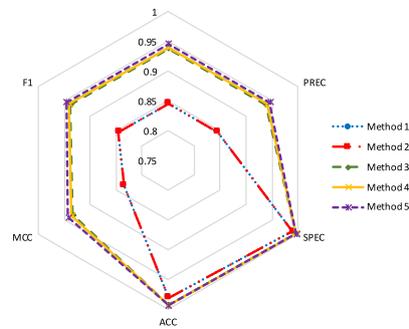
Fig. 8. Performance comparison of all five classification models for the 16 classes of the 2018 Medico test dataset. Methods 1 and 2 are similar in results but different from the other three methods (note that measurements start at 0.75).

5.2.1 16-Class Classification. In this 16-class classification task, the training dataset of the 2018 Medico Task was split into a 70% training dataset and a 30% validation dataset. Then, the test data given by the organizers was used to test the performance of five methods for classifying 16 classes of the GI tract findings.

We evaluated our five models based on the results collected by the organizers. The evaluated results of the main five models are tabulated in Table 4. With an MCC score of 0.9421, method 5 showed the best performance for classifying the 16 classes of GI tract findings. However, our GF-based approaches did not show results competitive with the DNN methods. The GF model introduced in method 1 could reach an MCC score of 0.8353. This result showed the best performance record for a GF-based method. A clear performance difference between the GF-based methods and the DNN-based methods can be seen in Figure 8. In this plot, we compared this performance difference using six performance measures: recall (REC), precision (PREC), specificity (SPEC), accuracy (ACC), MCC, and F-score (F1). According to this plot, it is clear that the areas of the hexagons covered by the GF methods are smaller than the areas covered by DNN methods. These results imply that three DL methods outperform two GF methods.

The CM of method 5 collected from the organizers of the 2018 Medico Task is tabulated in Table 5 for the in-depth investigation. According to the CM, we can identify two main bottlenecks to improve the performance of method 5. The first one is misclassification between esophagitis and normal-z-line, and the second one is misclassification between dyed-lifted-polyps and dyed-resection-margins. Therefore, images from these classes were manually examined to identify the reasons for these misclassifications. For the conflict between esophagitis

Table 5. CM of Method 5 (Our Best Model) Based on the Medico Test Dataset

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
										Actual Class							
	Ulcerative-colitis (A)	500	—	—	—	—	—	—	—	39	—	3	—	1	1	—	7
	Esophagitis (B)	3	432	48	—	—	—	—	—	—	—	—	—	—	—	—	—
	Normal-z-line (C)	1	121	513	—	—	—	—	—	—	—	—	—	1	—	—	—
	Dyed-lifted-polyps (D)	1	—	—	522	31	—	—	—	—	—	2	—	—	—	—	34
	Dyed-resection-margins (E)	—	—	—	33	532	—	—	—	—	—	1	—	—	—	—	17
Predicted Class	Out-of-patient (F)	—	—	—	—	1	5	—	—	—	—	—	—	—	—	—	—
	Normal-pylorus (G)	3	3	2	—	—	—	559	—	—	—	2	—	—	—	—	—
	Stool-inclusions (H)	—	—	—	—	—	—	—	501	7	—	—	—	—	—	—	—
	Stool-plenty (I)	1	—	—	—	—	—	—	—	1,918	—	—	—	—	—	—	1
	Blurry-nothing (J)	1	—	—	—	—	—	—	—	1	37	—	—	—	—	—	—
	Polyps (K)	10	—	—	1	—	—	1	—	—	—	358	6	—	1	—	46
	Normal-cecum (L)	18	—	—	—	—	—	—	—	—	—	6	578	—	—	—	2
	Colon-clear (M)	1	—	—	—	—	—	—	5	—	—	—	—	1,063	—	1	—
	Retroflex-rectum (N)	3	—	—	—	—	—	—	—	—	—	2	—	—	188	1	—
	Retroflex-stomach (O)	—	—	—	—	—	—	1	—	—	—	—	—	—	2	395	1
	Instruments (P)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	165

The diagonal value represents true predictions (number of images) of the model. A through P are the classes corresponding to the class names in the first column. The most confusion can be observed between classes B and C, and classes D and E. Looking at the images, we can see that they are quite similar in their visual features (colors, texture, etc.).

and normal-z-line, the reason is the very close locations of these two landmarks in the GI tract. However, the confusion between dyed-lifted-polyps and dyed-restrictions is caused because of the same color patterns and the same texture structures of both types of images. With these limitations, method 5 showed the best performance with an MCC of 0.9421, which was the important measurement to win the 2018 Medico Task. Based on the MCC value, we won second place in the 2018 Medico Task. The winning team [25] relabeled the development dataset and also generated more images out of the provided instruments class by placing the instrument as a foreground over the images of dyed-lifted-polyps, dyed-resection-margins, and ulcerative colitis to balance the instrument class for improving performance. However, we developed the model by only using the images provided by the task organizers for a fair comparison of the approaches with the limited dataset. Then, our next experiments were conducted to find the reusability of these well-performed models in different datasets with polyp and non-polyp categories (sub-categories of the 16 classes of primary tasks).

5.2.2 Polyp and Non-Polyp Classification Using the Pre-Trained Models. The following analysis was performed to identify the polyp classification ability of our five models on the same test dataset and different CVC datasets. The 16-class classification results collected from the Medico Task organizers were analyzed to calculate polyp detection performance in the Medico test data. Moreover, our models were tested with the CVC-356-plus, CVC-612-plus, and CVC-12k datasets without any modifications to the five models to compare the performance of polyp detection.

According to the correct and incorrect classifications of polyps and non-polyps in the test datasets, the first large column of Table 6 was calculated to measure the polyp detection performance of five models. In this evaluation process, all 15 classes except the polyp class were considered as the non-polyp classification because the number of outputs is 16 in the first models. For comparison, the MCC values of these tests are plotted in Figure 9. This graph shows that the polyp detection performance of the same dataset (the testing dataset of the Medico Task) is higher than on the completely new datasets (CVC-356-plus, CVC-612-plus, and CVC-12k) for both the

Table 6. Polyp Classification Results with and without Retraining for All Datasets and Methods

	M	Without Retraining						With Retraining to 2-Class Classification					
		REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Test Dataset	1	0.7834	0.4899	0.9635	0.9558	0.5987	0.6028	0.9550	0.9630	0.6740	0.9553	0.5430	0.9590
	2	0.7834	0.4899	0.9635	0.9558	0.5987	0.6028	0.9540	0.9630	0.6840	0.9537	0.5400	0.9580
	3	0.9733	0.8088	0.9897	0.9890	0.8819	0.8835	0.9813	0.6577	0.9772	0.9773	0.7934	0.7876
	4	0.9599	0.8467	0.9922	0.9908	0.8969	0.8997	0.9813	0.7384	0.9845	0.9843	0.8440	0.8427
	5	0.9572	0.8463	0.9922	0.9907	0.8954	0.8984	0.9706	0.7516	0.9857	0.9850	0.8470	0.8471
CVC-356-plus	1	0.3089	0.1053	0.5158	0.4835	−0.127	0.1571	0.8450	0.7990	0.1700	0.8446	0.0750	0.7780
	2	0.3089	0.1053	0.5158	0.4835	−0.127	0.1571	0.8510	0.8420	0.2070	0.8512	0.1930	0.7930
	3	0.7865	0.3738	0.7569	0.7615	0.4198	0.5068	0.8118	0.5547	0.8797	0.8691	0.5978	0.6591
	4	0.6713	0.4003	0.8144	0.7921	0.4010	0.5016	0.6517	0.4150	0.8305	0.8026	0.4068	0.5071
	5	0.6685	0.4837	0.8683	0.8372	0.4737	0.5613	0.6713	0.6408	0.9305	0.8902	0.5906	0.6557
CVC-612-plus	1	0.7696	0.7969	0.8295	0.8016	0.6008	0.7830	0.6980	0.8070	0.6530	0.6983	0.4740	0.6590
	2	0.7696	0.7969	0.8295	0.8016	0.6008	0.7830	0.7220	0.8170	0.6800	0.7218	0.5140	0.6910
	3	0.8415	0.6242	0.5597	0.6907	0.4137	0.7168	0.8382	0.6136	0.5412	0.6793	0.3932	0.7086
	4	0.8627	0.6559	0.6065	0.7257	0.4803	0.7452	0.8578	0.6890	0.6634	0.7538	0.5265	0.7642
	5	0.8137	0.6501	0.6193	0.7097	0.4379	0.7228	0.8007	0.7061	0.7102	0.7523	0.5104	0.7504
CVC-12k	1	0.4858	0.8391	0.5158	0.4907	0.0012	0.6154	0.1650	0.7880	0.8370	0.1651	0.0130	0.0530
	2	0.4858	0.8391	0.5158	0.4907	0.0012	0.6154	0.1650	0.8210	0.8380	0.1699	0.0290	0.0630
	3	0.6112	0.9289	0.7569	0.6347	0.2722	0.7373	0.6033	0.9631	0.8797	0.6479	0.3558	0.7419
	4	0.6236	0.9458	0.8144	0.6544	0.3241	0.7517	0.6459	0.9519	0.8305	0.6757	0.3539	0.7696
	5	0.5936	0.9591	0.8683	0.6379	0.3401	0.7333	0.5576	0.9766	0.9305	0.6178	0.3595	0.7099

M, method.

For training, 2018 Medico development data was used. We can observe that for some datasets, retraining seems to improve performance.
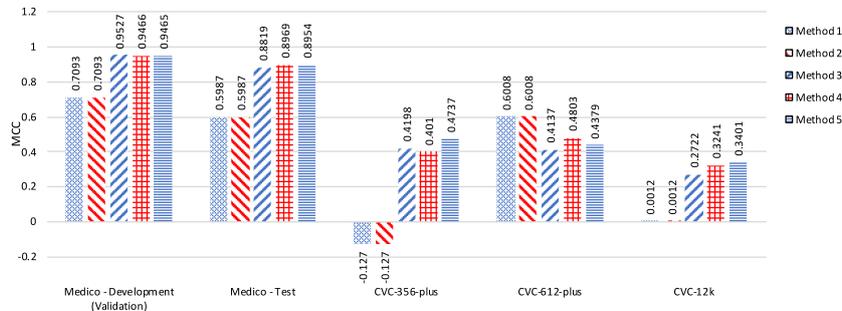

Fig. 9. Polyp and non-polyp classification capabilities (based on MCC) of all five methods that were trained using 2018 Medico development data to classify 16 classes. For most cases, methods 3 through 5 perform best. For the CVC-612-plus test data, methods 1 and 2 perform best.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics
Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality
Classification

GF-based approaches and the DNN approaches. This is the first analysis, and we emphasize that it shows that researchers need to do cross-dataset evaluations to prove the real capabilities of ML models.

From the first column of Table 6 and Figure 9, it is clear that the performance of the GF methods for different datasets (CVC-356plus, CVC-612-plus, and CVC-(356+612) dataset) is unpredictable because it presents huge value fluctuations in the graph with a negative MCC value. This shows the incapability of GF methods to make predictions on different datasets. The negative values of MCC in this experiment, such as −0.127 for the CVC-356-plus dataset, indicate that there is no agreement or only a non-relevant relationship between target and prediction. An MCC around zero would mean that the classifier is deciding randomly, and MCCs above zero would indicate correct classification. The closer to −1 or 1, the stronger the indication for being wrong or correct, respectively. However, the polyp detection performance of the GF-based methods in the CVC-612-plus dataset outperforms the DNN methods with an MCC value of 0.6008, whereas the best DNN method shows an MCC value of 0.4803. This prediction accuracy of the GF methods can be identified as an erroneous prediction, because the performance of this method for the other two CVC datasets shows poorer MCC scores than those of DNN-based approaches. Moreover, the DNN-based approaches show considerable steady MCC values for all new datasets, implying that the DNN methods are more generalizable than the GF methods.

Because the performance gap between the 16-class classification and polyp classification showed differences, we retrained our models to classify only the polyp and non-polyp classes. Therefore, our next experiments were performed to test how retraining our five ML models to classify polyps and non-polyps will influence performance.

For the retraining experiments, we first retrained the two GF methods with new ARFF files generated for polyp and non-polyp categories. Second, in the retraining stage of the three DNN methods, we changed only the last layer into two outputs. However, we did not change the loss function from categorical cross-entropy into binary cross-entropy because two-class categorical cross-entropy is equal to binary cross-entropy. Moreover, we retained the original optimization functions. Then, we retrained all five models using the same Medico dataset, which has only polyp and non-polyp classes. The results of these experiments are tabulated in the right columns of Table 6.

The results in Table 6 show that it can be difficult to evaluate the models and interpret the results after retraining for two-class classification. All MCC values of the five methods tested on the CVC-356-plus data show improvements. Similarly, for the CVC-612-plus test data, methods 4 and 5 show performance improvements from MCC values of 0.4803 and 0.4379 to 0.5265 and 0.5104, respectively. In contrast, methods 1, 2, and 3 show a performance drop, which is indicated by MCC values 0.6008, 0.6008, and 0.4137 reduced to 0.4740, 0.5140, and, 0.3932, respectively. Therefore, we extended our experiment by introducing additional retraining options with the CVC-356-plus and CVC-612-plus datasets. After that, the retraining process can be categorized as retraining the models to classify polyps and non-polyps using (i) only the same Medico training dataset (as tabulated in Table 6), (ii) the Medico dataset with the CVC-356-plus dataset, (iii) the Medico dataset with the CVC-612-plus dataset, and (iv) the Medico dataset with the CVC-356-plus and CVC-612-plus datasets. Then, our testing datasets are limited to two datasets: the Medico test dataset and the CVC-12k dataset. Results related to these new retraining processes can be seen in Table 7. When the models are trained using the balanced CVC-612-plus dataset in combination with the 2018 Medico development data, the DNN models show better MCC values (0.8189, 0.8555, and 0.8606) for methods 3, 4, and 5, respectively. This is true for the Medico test data and the two smaller CVC datasets. Moreover, the MCC values for the CVC-12k test data also achieve the best MCC values of 0.1421, 0.1418, and 0.1802 for methods 3, 4, and 5. An important observation from the CVC-12k dataset is also that looking at all other metrics but MCC and specificity could mislead to the assumption that the results are good—for example, scores above 0.8 for accuracy, which is often used as the only indicator for performance in similar studies.

In the first comparison, we plotted performance changes for the retraining with the different training datasets and tested them on the Medico test dataset. The changes in the Recall (REC), Precision (PREC), Specificity (SPEC), Accuracy (ACC), MCC, and F-score (F1) values can be seen as hexagon plots in Figure 10(a), (c), (e), (g), and (i),
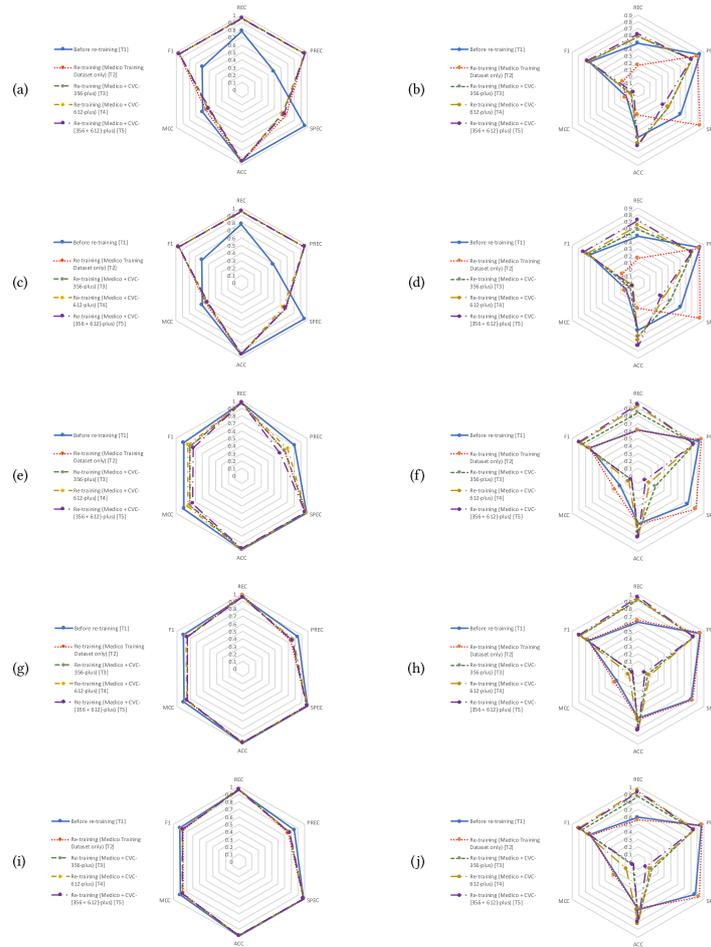
Appendix A. Published Articles

Fig. 10. Polyp and non-polyp classification using the proposed ML methods: 1, 2, 3, 4, and 5. The first column (sub-figures (a, c, e, g, i)) shows the results of the Medico test dataset, and the second column (sub-figures (b, d, f, h, j)) shows the results of the CVC-12k dataset. The methods are represented as follows: (a) and (b) for method 1, (c) and (d) for method 2, (e) and (f) for method 3, (g) and (h) for method 4, and (i) and (j) for method 5, respectively.

Table 7. Evaluation Results on Using CVC-356-plus and CVC-612-plus Combined as Training Data with Retraining to Classify Polyps and Non-Polyps

			MedicoTest Data						CVC-12k					
		M	REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Retraining Datasets with Medico Data	CVC-356-plus	1	0.9550	0.9610	0.6230	0.9549	0.5160	0.9570	0.5840	0.7040	0.3090	0.5836	−0.084	0.6320
		2	0.9520	0.9620	0.6710	0.9521	0.5260	0.9560	0.5810	0.7100	0.3360	0.5807	−0.065	0.6310
		3	0.9626	0.6630	0.9781	0.9775	0.7887	0.7852	0.8423	0.8565	0.2665	0.7494	0.1052	0.8493
		4	0.9599	0.7526	0.9859	0.9848	0.8427	0.8437	0.9192	0.8481	0.1441	0.7941	0.0810	0.8822
		5	0.9706	0.7773	0.9876	0.9868	0.8623	0.8633	0.8694	0.8507	0.2068	0.7625	0.0802	0.8599
	CVC-612-plus	1	0.9510	0.9590	0.6270	0.9508	0.4970	0.9540	0.5840	0.7030	0.3040	0.5842	−0.087	0.6320
		2	0.9530	0.9610	0.6430	0.9530	0.5160	0.9560	0.6400	0.6970	0.2240	0.6395	−0.117	0.6660
		3	0.9652	0.7092	0.9823	0.9816	0.8189	0.8177	0.9325	0.8546	0.1752	0.8103	0.1421	0.8918
		4	0.9572	0.7766	0.9877	0.9864	0.8555	0.8575	0.9336	0.8544	0.1731	0.8109	0.1418	0.8922
		5	0.9626	0.7809	0.9879	0.9868	0.8606	0.8623	0.9486	0.8571	0.1778	0.8242	0.1802	0.9005
	CVC-(356+612)	1	0.9500	0.9600	0.6480	0.9503	0.5050	0.9540	0.6180	0.6930	0.2280	0.6179	−0.129	0.6520
		2	0.9500	0.9610	0.6710	0.9503	0.5170	0.9550	0.7200	0.7010	0.1820	0.7199	−0.105	0.7100
		3	0.9733	0.5909	0.9699	0.9700	0.7458	0.7354	0.9537	0.8479	0.1109	0.8177	0.1028	0.8977
		4	0.9545	0.7596	0.9865	0.9851	0.8443	0.8460	0.9543	0.8463	0.0995	0.8164	0.0874	0.8971
		5	0.9599	0.7771	0.9877	0.9865	0.8571	0.8589	0.9278	0.8462	0.1239	0.7981	0.0699	0.8851

The 2018 Medico test dataset and the CVC-12k dataset are the test datasets. Using the balanced CVC-612-plus as training data, we achieve the best results. Combining CVC-356-plus and the CVC-612-plus does not improve performance. Overall, the performance is better on the Medico test dataset.

which correspond to methods 1, 2 ,3, 4, and 5, respectively. In these plots, T1 is used to present performance values before retraining the ML models into 2-class classification (binary classification). In this case, 15 classes except for the polyp class of the 16 classes were considered as the non-polyp class, and the polyp class is counted as the same polyp class. Furthermore, from T2 to T5, lines are used to present models with only two outputs. The T2 plot represents models' performance for the retraining using the Medico training dataset. Similarly, T3, T4, and T5 represent the retraining process using the Medico dataset and the CVC-356-plus dataset, the Medico dataset, and the CVC-612-plus dataset, and the Medico dataset, the CVC-356-plus dataset, and the CVC-612 dataset, respectively.

In the second series of experiments in this session, the same experiments were performed and tested on the CVC-12k dataset. The results obtained from these experiments are tabulated in Tables 6 and 7. Then, relevant results from these tables are plotted in Figure 10(b), (d), (f), (h), and (j). These plots use line notations similar to the preceding experiments.

Using the plot series in Figure 10, we can examine the reusability of ML models to classify polyps and non-polyps, which are sub-classes of the primary classes on the task. For example, if we compare plots in Figure 10(a) and (b), then we can know how method 1 performs to classify polyps and non-polyps within the test dataset the same as the training dataset and within an entirely new dataset. While investigating these plots, the proportion of the number of polyps and non-polyps is an important factor in explaining the shape of these hexagon plots.

If we compare the GF methods (Figures 10(a) through (d)) and the DL methods (Figures 10(e) through (j)), it is clear that the DL methods outperform the GF methods in both the Medico Task and polyp classification task introduced in this article. This implies that the DL methods are capable of extracting deep features that cannot be extracted by manual feature extraction methods used by the GF methods. With the retraining process in the GF methods, we can see performance differences between the Medico dataset and the CVC-12k dataset. The main conclusion that we make is that GF-based methods are not able to capture the underlying patterns that would allow for efficient classification; thus, their performance is low.

Appendix A. Published Articles

(a) ROC and PR curves for the Medico test dataset.

(b) ROC and PR curves for the CVC-12k dataset.

(c) PRC curves for the Medico test dataset.
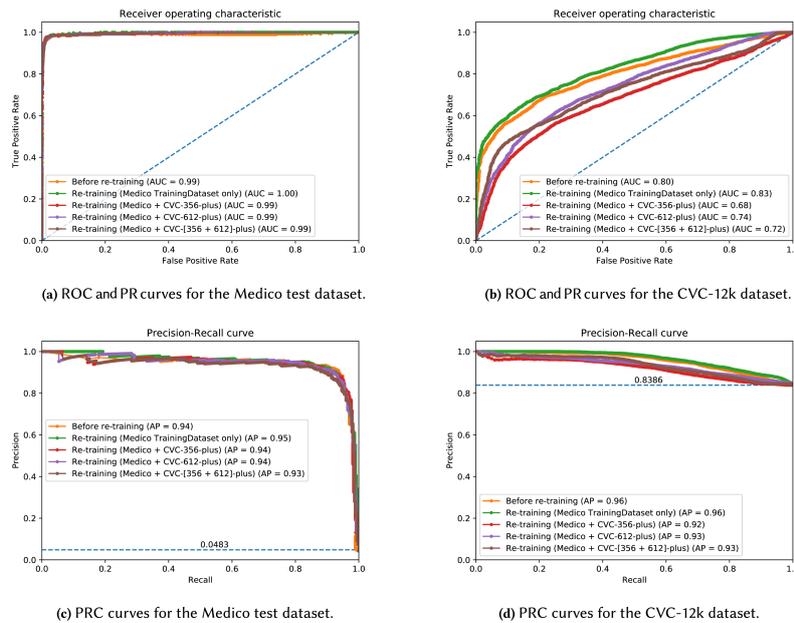
(d) PRC curves for the CVC-12k dataset.

Fig. 11. ROC and Precision-Recall Curve (PRC) curves for method 5 trained on the CVC-356-plus and CVC-612-plus datasets as mentioned in the legends. Testing datasets are the CVC-12k and Medico test datasets. Overall, good performance can be observed in both ROC and PR curves. For CVC-12k, the PR curve shows the interesting case of a high random baseline for a biased dataset.

Plots in the first and second columns in Figure 10 show completely different behaviors for the same retraining process when we use different test datasets. The test dataset for the first column comes from the same domain as the training data, and the test dataset for the second column comes from the completely new domain, such as the CVC-12k dataset. To investigate these unusual performance changes, we generated and examined ROC and PR curves for the best DNN model (method 5). The ROC and PR curves for method 5 with the Medico test data (for the plot in Figure 10) are depicted in Figure 11(a) and (c). Similarly, the ROC and PR curves for method 5 with CVC-12k data (for the plot in Figure 10) are plotted in Figure 11(b) and (d).

Analysis of ROC curves is more robust for ML models that are used with balanced datasets, whereas PR curves are more valuable for ML methods when the methods engage with imbalanced datasets. However, we have used both curves in this paper to investigate the behavior of these curves while we are using highly imbalanced datasets. Consequently, the PR curves show completely different baseline values of 0.0483 for the Medico test dataset and 0.8386 for the CVC-12k dataset. The small baseline value arises in the plot in Figure 11(c) as a result of small polyps to the non-polyp proportion in the Medico test dataset. Conversely, the high baseline value in Figure 11(d) appears there as an effect on a high ratio of polyps to non-polyps.

To get a better understanding of the above plots, we selected the plots in Figure 10(i) and (j), and ROC and PR curves in Figure 11. With this selection, first, we analyzed T1 and T2 from the hexagon plots and the

Table 8. Method 5: Training Only the MLP vs. the Complete DNN

Test		Training Only the MLP						Training the Whole DNN					
Data	T	REC	PREC	SPEC	ACC	MCC	F1	REC	PREC	SPEC	ACC	MCC	F1
Medico Test Data	T1	0.9572	0.5859	0.9698	0.9692	0.7357	0.7269	0.9706	0.7773	0.9876	0.9868	0.8623	0.8633
	T2	0.9599	0.7804	0.9879	0.9867	0.8591	0.8609	0.9626	0.7809	0.9879	0.9868	0.8606	0.8623
	T3	0.9626	0.6316	0.9749	0.9744	0.7684	0.7627	0.9599	0.7771	0.9877	0.9865	0.8571	0.8589
CVC-12k	T1	0.6984	0.8972	0.5842	0.6799	0.2184	0.7854	0.8694	0.8507	0.2068	0.7625	0.0802	0.8599
	T2	0.7588	0.8993	0.5583	0.7265	0.2565	0.8231	0.9486	0.8571	0.1778	0.8242	0.1802	0.9005
	T3	0.7614	0.8933	0.5272	0.7236	0.2352	0.8221	0.9278	0.8462	0.1239	0.7981	0.0699	0.8851

T, the additional training dataset that was added to the Medico dataset; T1, Medico dataset + CVC-356-plus; T2, Medico dataset + CVC-612-plus; T3, Medico dataset + CVC-356-plus + CVC-612-plus.

corresponding ROC and PR curves. Although T2 shows a performance loss compared to T1 in Figure 10(i), Figure 10(j) shows that T2 achieves a performance improvement over T1. Next, we look for the reasons for these performance changes.

In method 5, the model with the 16 outputs corresponding to T1 has 15 choices to classify non-polyp images. Similarly, the Medico test dataset has more non-polyp images than polyp images. However, the model corresponding to T2 has a 50% chance to classify both polyps and non-polyps. As a result, the model of T1 shows better performance than the model of T2 in Figure 10(i). Because this shows a slight performance change, we cannot see the same difference in ROC and PR curves in Figure 11(a) and (c). In contrast, T2 in the plot in Figure 10(j) shows performance improvement when the model has a 50:50 chance for classifying polyps and non-polyps. This improvement occurred as a result of a large number of polyps in the CVC-12k dataset. The ROC and PR curves in plots in Figure 11(b) and (d) show this performance difference precisely. In other words, the model of T2 has a better chance of classifying polyps compared to the 1/16th chance in the model of T1.

The retrained models corresponding to T3, T4, and T5 do not show considerable performance changes for the Medico test dataset, as we can see from plots in Figure 10(i), (a), and (c). Conversely, the retraining method used in T3, T4, and T5 for the CVC-12k dataset shows large performance changes in the plots in Figure 10(j), (b), and (d). However, these methods show an overall performance loss. More comparisons on these plots are discussed in Section 6.

For the following experiments, we analyzed method 5 even further. The main focus of this analysis is to understand the behavior of the best model for training only the MLP versus training the whole DNN. In this experiment, we collected results for two main test datasets: the Medico test dataset and the CVC-12k dataset. Then, we collected performance measures from the two training mechanisms: training only the MLP and training the whole DNN. Furthermore, results were tabulated in Table 8, and corresponding graphs were depicted in Figure 12 to analyze them.

The first row of Figure 12 shows the differences in the performance of testing with the Medico test data. In the second row, it presents the performance changes for the CVC-12k dataset. The dotted lines in plots in Figure 12 represent training MLP. Similarly, the dashed lines represent training the whole DNN. The three plots of each row represent results of retraining the model with the Medico training data and CVC-356-plus dataset, the Medico training data and CVC-612-plus dataset, and the Medico training data and both CVC-356-plus and CVC-612-plus datasets, respectively.

According to the plots in Figure 12(a) through (c), it is clear that retraining the whole DNN can be used to improve the overall performance of the DNN model because we can see performance improvement in these plots except in Figure 12(b), which shows closely equal performance metrics. However, in test cases with the CVC-12k dataset, it shows a completely new behavior for retraining the whole DNN as depicted in Figure 12(d) through (f). These plots show large changes in the performance hexagons with considerable positive improvements for the

(a) Medico test data - (CVC-356-plus).

(b) Medico test data - (CVC-612-plus).

(c) Medico test data - (CVC-356-plus + CVC-612-plus).

(d) CVC-12k - (CVC-356-plus.)

(e) CVC-12k - (CVC-612-plus).
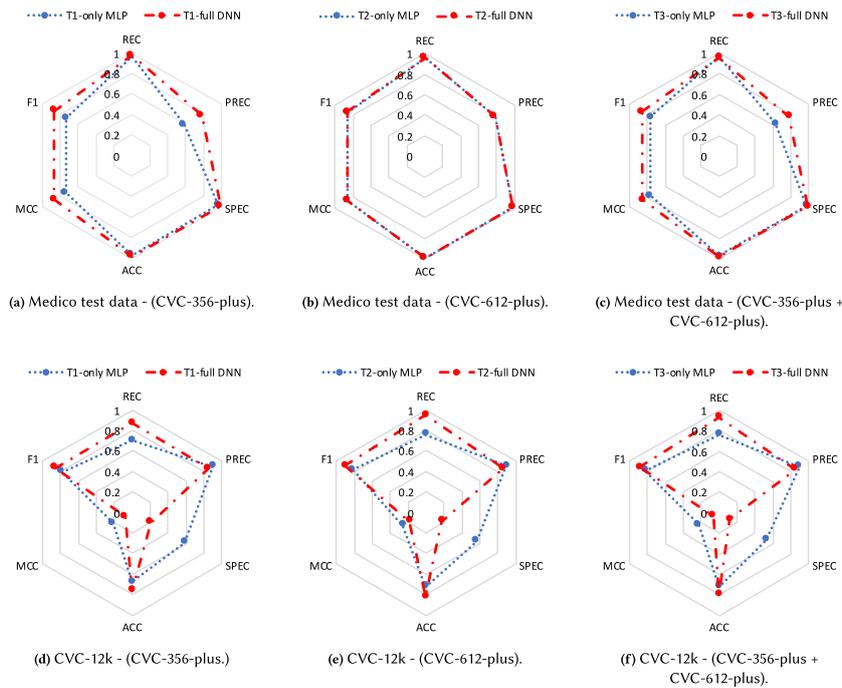
(f) CVC-12k - (CVC-356-plus + CVC-612-plus).

Fig. 12. Behavior of the complex DNN method (method 5) while training only MLP compared to training the whole DNN. The first row shows the effects for both cases when the test dataset is the Medico test data, and the second row shows the result when the test dataset is the CVC-12k dataset. T1, T2, and T3 represent the training dataset used for the model. (T1, Medico training dataset + CVC-356; T2, Medico training dataset + CVC-612; T3, Medico training dataset + CVC-356 + CVC-612.)

recall and considerable performance loss for the specificity values. This experiment also shows that researchers could be misled by the performance monitoring process of DNN methods using a single dataset. In other words, according to the first row of the figure, researchers may conclude that retraining the whole DNN is a positive factor. However, the results of the second row prove that it is not always true by showing performance losses for the same technique.

The results presented in plots in Figure 12 show difficulties in adapting ML models for cross-dataset general-ization with a different perspective. In that experiment, the performance loss in specificity, which is a parameter of reflecting True Negative (TN) detection, shows that method 5 is affected by imbalanced data in the CVC-12k dataset. The main reason for the effect is that the CVC-12k dataset contains a lower percentage of negative images compared to positive ones. This reflects an important factor to take into account when developing gen-eralizable ML models, which is that the ratio of negative and positive findings needs to be taken into account when looking at metrics. Metrics such as MCC are better suited to interpret results. In terms of ROC compared

to the PR curve, the results show that the PR curve reflects the performance of the model more realistically than ROC.

6 DISCUSSION

In this section, we present our findings and point out several important considerations for future research. Our discussion follows the same sequence as our contributions in this article.

In our experiments, combinations of ResNet-152, DenseNet-161, and an additional MLP produced the best result for the Medico 2018 dataset. The reported results from this model for the Medico Task led us to hold second place based on the MCC values calculated by organizers, and there was only a tiny gap of around 0.0029. Furthermore, the winning team of this competition used additional data items that were made by photo-editing tools for the imbalanced classes, such as the out-of-patients class. In contrast to this, our method 5 works well without using manually annotated data items because of the procedure we followed to implement and train that model. The procedure of implementing such a complex model is described step by step in Section 4.3, and anyone can follow these steps to get a well-performing DL model in a classification task.

In addition to the implementation and the procedure used in method 5, the data-filling mechanism used to fill the out-of-patient imbalance class shows impressive performance gain. This method is preferred when one class has a small number of data items in a multi-class classification task. In our work, without annotating more data ourself, which also requires the help of medical experts, we prefer to use random images from the Internet, as described in Section 5.1. This is an efficient way to add more data items without spending more time on manual annotation or creating synthetic data items. The preceding method works because the random images influence the ML models to make a wider range of possibilities to classify images into a particular class.

Dyed-lifted-polyps, dyed-resection-margins, esophagitis, and normal-z-line raised classification conflicts in our best method (method 5). If we could overcome these conflicts, then the model would perform better than the current recorded performance in the 2018 Medico Task. To identify the reasons for these classification conflicts, we manually investigated the images of these classes. If we compare sample images of dyed-lifted-polyps (Figure 1(c)), dyed-resection-margins (Figure 1(d)), esophagitis (Figure 1(e)), and normal-z-line (Figure 1(i)), then we can identify that this conflict was caused as a result of similar texture and shapes of these images. To overcome this problem, researchers can select only the images that made the conflict and train a new DL model to classify them into the correct classes. Then, this model can be added to the model introduced in method 5 using the property of its expandability.

Can we use our best DL model for real systems in hospitals to classify GI findings? Or can we use the state-of-the-art ML classification models introduced by researchers in real applications? Toward answering this question, this article focuses on deep evaluations of the proposed methods as one of the main contributions. Regularly, researchers present the performance of their classification models using only a test dataset, which was reserved from the dataset used to produce the training data. In addition, they measure the performance by selecting only a few measurements out of the REC, PREC, SPEC, ACC, MCC, and F1. However, we emphasize the requirements of an in-depth analysis of all of these six parameters at once to identify the real performance of ML models. Several of the works listed in Table 1 do not use this methodology as part of their evaluations. This makes it difficult to reason about the real-world performance of the proposed methods and how they compare with other methods. In this article, we also consider the importance of evaluating ML models with cross-datasets.

Why do we need cross-dataset evaluations? To explain this requirement, we consider the research work done by Wang et al. [75]. They presented an area under the ROC curve of 0.984 and a per-image sensitivity of 94.38 for polyp detection. In our first look, these results show a good DL model. Similarly, our results in Figure 10(i) and 11(a) and (c) reflect the same impression in the first look because it shows excellent performance as a DL model. However, after analyzing cross-dataset performance for polyp detection with a completely new dataset like CVC-12k, we recognized that performance gain is not enough for applying it in real applications. Therefore, from this work, we emphasize that researchers want to consider cross-dataset evaluations thoroughly before applying

Appendix A. Published Articles

their solutions in real-world applications. Otherwise, the selection bias, the capture bias, and the category bias (label bias) problems may appear in the results. Then, we may end up with the wrong conclusion about research works. All of these facts imply that more research must be performed to improve the generalizability along with the performance improvement on a single dataset or single data source.

7 CONCLUSION

We studied cross-dataset bias and evaluation metrics interpretation in ML using five methods and four different datasets within the field of GI endoscopy as respective use case. In particular, we performed an extensive study of ML models in the context of medical applications based on a use case of GI tract abnormality classification across different datasets. The main conclusion and resulting recommendation is that a multi-center or cross-dataset evaluation is important, if not essential, for ML models in the medical field to obtain a realistic understanding of the performance of such models in real-world settings.

We found that the combination of DNN ResNet-152 and DenseNet-161 with an additional MLP performed best on both the validation and test datasets. This model shows that a combination of multiple pre-trained DNN models can have better capabilities to classify images into the correct classes because of their cumulative decision-making capabilities. We also proposed an evaluation method using six measures: REC, PREC, SPEC, ACC, MCC, and F1. Moreover, we suggest that these measures should be presented all at once using hexagon plots that convey a complete view of real performance. It is our hope that these tools can enable a more realistic evaluation and comparison of ML methods.

Furthermore, we presented cross-dataset evaluations to identify the generalizability of our ML models, emphasizing the fact that achieving high scores for evaluation metrics does not always represent the real performance of ML models and should be interpreted with care. By evaluating the ML models with cross-datasets experiments, we showed the complexity of understanding the real functional performance of the models. The state-of-the-art research works that perform classification cannot be used in practical applications because of their lack of generalizability. Based on the experimental results, we conclude that researchers should focus on implementing and researching generalizable ML models with cross-dataset evaluations. Rather than presenting metrics calculated from a simple training and testing split of the data, we suggest to always rely on cross-dataset evaluation to obtain a real-world representative indication of model performance. This is especially important in a medical context because one has to make sure that the obtained models are reliable and not just perform well on a specific dataset.

Finally, we want to point out that the lack of generalization, as evidenced by the poor result for cross-dataset evaluation presented in this article, rises a very important question: in the context of cross-dataset or multi-center studies, is it really possible to have generalizable ML models? This is something that we ourselves plan to investigate further in future work, and it is ou hope that other researchers in computer science and medicine will do the same or at least have the question in their mind when performing similar studies.

REFERENCES

[1] Taruna Agrawal, Rahul Gupta, Saurabh Sahu, and Carol Y. Espy-Wilson. 2017. SCL-UMD at the Medico Task-MediaEval 2017: Transfer learning based classification of medical images. In *Proceedings of MediaEval 2017*.

[2] Luís A. Alexandre, Nuno Nobre, and João Casteleiro. 2008. Color and position versus texture features for endoscopic polyp detection. In *Proceedings of IEEE BMEI2008*, Vol. 2. 38–42.

[3] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. 2009. Texture-based polyp detection in colonoscopy. In *Bildverarbeitung für die Medizin 2009*. 346–350.

[4] Quentin Angermann, Jorge Bernal, Cristina Sánchez-Montes, Maroua Hammami, Gloria Fernández-Esparrach, Xavier Dray, Olivier Romain, F. Javier Sánchez, and Aymeric Histace. 2017. Towards real-time polyp detection in colonoscopy videos: Adapting still frame-based methodologies for video sequences analysis. In *Proceedings of CARE and CLIP 2017*. 29–41.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification

[5] Jorge Bernal, Aymeric Histace, Marc Masana, Quentin Angermann, Cristina Sánchez-Montes, Cristina Rodriguez, Maroua Hammami, et al. 2018. Polyp detection benchmark in colonoscopy videos using GTCreator: A novel fully configurable tool for easy and fast annotation of image databases. In *Proceedings of CARS 2018*.

[6] Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. 2015. WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics* 43 (2015), 99–111.

[7] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. 2012. Towards automatic polyp detection with a polyp appearance model. *Pattern Recognition* 45, 9 (2012), 3166–3182.

[8] Jorge Bernal, Javier Sánchez, and Fernando Vilarino. 2013. Impact of image preprocessing methods on polyp localization in colonoscopy frames. In *Proceedings of IEEE EMBC 2013*. 7350–7354.

[9] Jorge Bernal, Nima Tajkbaksh, Francisco Javier Sánchez, Bogdan J. Matuszewski, Hao Chen, Lequan Yu, Quentin Angermann, et al. 2017. Comparative validation of polyp detection methods in video colonoscopy: Results from the MICCAI 2015 Endoscopic Vision Challenge. *IEEE Transactions on Medical Imaging* 36, 6 (2017), 1231–1249.

[10] Rune Johan Borgli, Pål Halvorsen, Michael Riegler, and Håkon Kvale Stensland. 2018. Automatic hyperparameter optimization in Keras for the MediaEval 2018 Medico Multimedia Task. In *Proceedings of MediaEval 2018*.

[11] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT 2010*. 177–186.

[12] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L. Siegel, Lindsey A. Torre, and Ahmedin Jemal. 2018. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians* 68, 6 (2018), 394–424.

[13] Da-Chuan Cheng, Wen-Chien Ting, Yung-Fu Chen, and Xiaoyi Jiang. 2011. Automatic detection of colorectal polyps in static images. *Biomedical Engineering: Applications, Basis and Communications* 23, 05 (2011), 357–367.

[14] Torch Contributors. 2018. Torchvision Models. Retrieved May 7, 2020 from https://pytorch.org/docs/stable/torchvision/models.html.

[15] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. A tutorial on the cross-entropy method. *Annals of Operations Research* 134 (2005), 19–67.

[16] Thomas de Lange, Pål Halvorsen, and Michael Riegler. 2018. Methodology to develop machine learning algorithms to improve performance in gastrointestinal endoscopy. *World Journal of Gastroenterology* 24, 45 (2018), 5057.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE CVPR 2009*. 248–255.

[18] Danielle Dias and Ulisses Dias. 2018. Transfer learning with CNN architectures for classifying gastrointestinal diseases and anatomical landmarks. In *Proceedings of MediaEval 2018*.

[19] Patrick Doetsch, Christian Buck, Pavlo Golik, Niklas Hoppe, Michael Kramp, Johannes Laudenberg, Christian Oberdörfer, Pascal Steingrube, Jens Forster, and Arne Mauser. 2009. Logistic model trees with AUC split criterion for the KDD Cup 2009 Small Challenge. In *Proceedings of KDD-Cup '09*. 77–88.

[20] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. 2009. Pedestrian detection: A benchmark. In *Proceedings of IEEE CVPR 2009*.

[21] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2000. Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors). *Annals of Statistics* 28, 2 (2000), 337–407.

[22] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11, 1 (2009), 10–18.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of IEEE CVPR 2016*. 770–778.

[24] Steven A. Hicks, Pia H. Smedsrud, Pål Halvorsen, and Michael Riegler. 2018. Deep learning based disease detection using domain specific transfer learning. In *Proceedings of MediaEval 2018*.

[25] Trung-Hieu Hoang, Hai-Dang Nguyen, and Thanh-An Nguyen. 2018. An application of residual network and faster - RCNN for Medico: Multimedia Task at MediaEval 2018. In *Proceedings of MediaEval 2018*.

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of IEEE CVPR 2017*. 2261–2269.

[27] Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C. De Groen. 2007. Polyp detection in colonoscopy video using elliptical shape feature. In *Proceedings of IEEE ICIP 2007*, Vol. 2. 465–468.

[28] Dimitrios K. Iakovidis, Dimitrios E. Maroulis, Stavros A. Karkanis, and A. Brokos. 2005. A comparative study of texture features for the discrimination of gastric polyps in endoscopic video. In *Proceedings of IEEE CBMS 2005*. 575–580.

[29] Yuji Iwahori, Takayuki Shinohara, Akira Hattori, Robert J. Woodham, Shinji Fukui, Manas Kamal Bhuyan, and Kunio Kasugai. 2013. Automatic polyp detection in endoscope images using a Hessian filter. In *Proceedings of MVA 2013*, Vol. 13. 21–24.

[30] Debesh Jha, Pia Smedsrud, Michael Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard Johansen. 2020. Kvasir-SEG: A segmented polyp dataset. In *Proceedings of MMM 2020*. 1–12.

[31] Xiao Jia and Max Q.-H. Meng. 2017. Gastrointestinal bleeding detection in wireless capsule endoscopy images using handcrafted and CNN features. In *Proceedings of IEEE EMBC 2017*. 3154–3157.

Appendix A. Published Articles

[32] Stavros A. Karkanis, Dimitrios K. Iakovidis, Dimitrios E. Maroulis, Dimitris A. Karras, and M. Tzivras. 2003. Computer-aided tumor detection in endoscopic video using color wavelet features. *IEEE Transactions on Information Technology in Biomedicine* 7, 3 (2003), 141–152.

[33] Zeshan Khan and Muhammad Atif Tahir. 2018. Majority voting of heterogeneous classifiers for finding abnormalities in the gastro-intestinal tract. In *Proceedings of MediaEval 2018*.

[34] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei Efros, and Antonio Torralba. 2012. Undoing the damage of dataset bias. In *Proceedings of ECCV 2012*.

[35] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.

[36] Mathias Kirkerød, Vajira Thambawita, Michael Riegler, and Pål Halvorsen. 2018. Using preprocessing as a tool in medical image detection. In *Proceedings of MediaEval 2018*.

[37] Tobey H. Ko, Zhonglei Gu, and Yang Liu. 2018. Weighted discriminant embedding: Discriminant subspace learning for imbalanced medical data classification. In *Proceedings of MediaEval 2018*.

[38] Niels Landwehr, Mark Hall, and Eibe Frank. 2005. Logistic model trees. *Machine Learning* 59, 1–2 (2005), 161–205.

[39] A. M. Leufkens, M. G. H. van Oijen, F. P. Vleggaar, and P. D. Siersema. 2012. Factors influencing the miss rate of polyps in a back-to-back colonoscopy study. *Endoscopy* 44, 05 (2012), 470–475.

[40] Baopu Li and Max Q.-H. Meng. 2012. Tumor recognition in wireless capsule endoscopy images using textural features and SVM-based feature selection. *IEEE Transactions on Information Technology in Biomedicine* 16, 3 (2012), 323–329.

[41] David Lieberman. 2005. Quality and colonoscopy: A new imperative. *Gastrointestinal Endoscopy* 61, 3 (2005), 392–394.

[42] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. 2016. LIRE: Open source visual information retrieval. In *Proceedings of ACM MMSys 2016*. 30.

[43] Alexander V. Mamonov, Isabel N. Figueiredo, Pedro N. Figueiredo, and Yen-Hsi Richard Tsai. 2014. Automated polyp detection in colon capsule endoscopy. *IEEE Transactions on Medical Imaging* 33, 7 (2014), 1488–1502.

[44] Yuichi Mori and Shin-Ei Kudo. 2018. Detecting colorectal polyps via machine learning. *Nature Biomedical Engineering* 2, 10 (2018), 713.

[45] Syed Sadiq Ali Naqvi, Shees Nadeem, Muhammad Zaid, and Muhammad Atif Tahir. 2017. Ensemble of texture features for finding abnormalities in the gastro-intestinal tract. In *Proceedings of MediaEval 2017*.

[46] Olga Ostroukhova, Konstantin Pogorelov, Michael Riegler, Duc-Tien Dang-Nguyen, and Pål Halvorsen. 2018. Transfer learning with prioritized classification and training dataset equalization for medical objects detection. In *Proceedings of MediaEval 2018*.

[47] Sun Young Park, Dustin Sargent, Inbar Spofford, Kirby G. Vosburgh, and Y. A-Rahim. 2012. A colon video analysis framework for polyp detection. *IEEE Transactions on Biomedical Engineering* 59, 5 (2012), 1408.

[48] Stefan Petscharnig, Klaus Schöffmann, and Mathias Lux. 2017. An inception-like CNN architecture for GI disease and anatomical landmark classification. In *Proceedings of MediaEval 2017*.

[49] Konstantin Pogorelov, Olga Ostroukhova, Mattis Jeppsson, Håvard Espeland, Carsten Griwodz, Thomas de Lange, Dag Johansen, Michael Riegler, and Pål Halvorsen. 2018. Deep learning and hand-crafted feature based approaches for polyp detection in medical videos. In *Proceedings of IEEE CBMS 2018*. 381–386.

[50] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Carsten Griwodz, Dag Johansen, Concetto Spampinato, et al. 2017. Nerthus: A bowel preparation quality video dataset. In *Proceedings of ACM MMSys 2017*. 170–174.

[51] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, et al. 2017. Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection. In *Proceedings of ACM MMSys 2017*. 164–169.

[52] Konstantin Pogorelov, Michael Riegler, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Carsten Griwodz, Peter Thelin Schmidt, and Pål Halvorsen. 2017. Efficient disease detection in gastrointestinal videos—Global features versus neural networks. *International Journal of Multimedia Tools and Applications* 76, 21 (2017), 22493–22525.

[53] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico multimedia task at MediaEval 2018. In *Proceedings of MediaEval 2018*.

[54] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Carsten Griwodz, Thomas de Lange, Kristin Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, and Olga Ostroukhova. 2017. A comparison of deep learning with global features for gastrointestinal disease detection. In *Proceedings of MediaEval 2017*.

[55] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico multimedia task at MediaEval 2018. In *Proceedings of MediaEval 2018*.

[56] Michael Riegler, Martha Larson, Mathias Lux, and Christoph Kofler. 2014. How 'how' reflects what's what: Content-based exploitation of how users frame social images. In *Proceedings of ACM MM 2014*. 397–406.

[57] Michael Riegler, Mathias Lux, Carsten Griwodz, Concetto Spampinato, Thomas de Lange, Sigrun L. Eskeland, Konstantin Pogorelov, et al. 2016. Multimedia and medicine: Teammates for better disease detection and survival. In *Proceedings of ACM MM 2016*. 968–977.

[58] Michael Riegler, Konstantin Pogorelov, Sigrun Losada Eskeland, Peter Thelin Schmidt, Zeno Albisser, Dag Johansen, Carsten Griwodz, Pål Halvorsen, and Thomas De Lange. 2017. From annotation to computer-aided diagnosis: Detailed evaluation of a medical multimedia system. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 3 (2017), 26.

A.9. Paper IX - An Extensive Study on Cross-Dataset Bias and Evaluation Metrics
Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality
Classification

[59] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Kristin Randel, Sigrun Losada Eskeland, Duc-Tien Dang-Nguyen, Mathias Lux, Carsten Griwodz, Concetto Spampinato, and Thomas Lange. 2017. Multimedia for medicine: The Medico Task at MediaEval 2017. In *Proceedings of MediaEval 2017*.

[60] Sebastian Ruder. 2016. An overview of gradient descent optimization algorithms. arXiv:1609.04747.

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision* 115 (2015), 211–252.

[62] Steven L. Salzberg. 1994. C4. 5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning* 16, 3 (1994), 235–240.

[63] Younghak Shin and Ilangko Balasingham. 2017. Comparison of hand-craft feature based SVM and CNN based deep learning framework for automatic polyp classification. In *Proceedings of IEEE EMBC 2017*. 3277–3280.

[64] Michael Steiner, Mathias Lux, and Pål Halvorsen. 2018. The 2018 Medico Multimedia Task submission of Team NOAT using neural network features and search-based classification. In *Proceedings of MediaEval 2018*.

[65] Marc Sumner, Eibe Frank, and Mark Hall. 2005. Speeding up logistic model tree induction. In *Proceedings of PKDD 2005*. 675–683.

[66] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of ICML 2013*. 1139–1147.

[67] Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. 2015. Automatic polyp detection in colonoscopy videos using an ensemble of convolutional neural networks. In *Proceedings of IEEE ISBI 2015*. 79–83.

[68] Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. 2016. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging* 35, 2 (2016), 630–644.

[69] Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. 2016. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE Transactions on Medical Imaging* 35, 5 (2016), 1299–1312.

[70] Mario Taschwer, Manfred Jürgen Primus, Klaus Schoeffmann, and Oge Marques. 2018. Early and late fusion of classifiers for the MediaEval Medico Task. In *Proceedings of MediaEval 2018*.

[71] Vajira Thambawita, Debesh Jha, Michael Riegler, Pål Halvorsen, Hugo Lewi Hammer, Håvard D. Johansen, and Dag Johansen. 2018. The Medico-Task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. In *Proceedings of MediaEval 2018*.

[72] A. Torralba and A. A. Efros. 2011. Unbiased look at dataset bias. In *Proceedings of IEEE CVPR 2011*. 1521–1528.

[73] David Vázquez, Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Antonio M. López, Adriana Romero, Michal Drozdzal, and Aaron C. Courville. 2017. A benchmark for endoluminal scene segmentation of colonoscopy images. *Journal of Healthcare Engineering* 2017 (2017), 4037190.

[74] L. Von Karsa, J. Patnick, and N. Segnan. 2012. European guidelines for quality assurance in colorectal cancer screening and diagnosis. First edition—Executive summary. *Endoscopy* 44, Suppl. 3 (2012), SE1–SE8.

[75] Pu Wang, Xiao Xiao, Jeremy R. Glissen Brown, Tyler M. Berzin, Mengtian Tu, Fei Xiong, Xiao Hu, et al. 2018. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. *Nature Biomedical Engineering* 2, 10 (2018), 741.

[76] Yi Wang, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C. De Groen. 2014. Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy. *IEEE Journal of Biomedical and Health Informatics* 18, 4 (2014), 1379–1389.

[77] Yi Wang, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C. De Groen. 2015. Polyp-Alert: Near real-time feedback during colonoscopy. *International Journal of Computer Methods and Programs in Biomedicine* 120, 3 (2015), 164–179.

[78] Lequan Yu, Hao Chen, Qi Dou, Jing Qin, and Pheng Ann Heng. 2017. Integrating online and offline three-dimensional deep learning for automated polyp detection in colonoscopy videos. *IEEE Journal of Biomedical and Health Informatics* 21, 1 (2017), 65–75.

[79] Yixuan Yuan, Dengwang Li, and Max Q.-H. Meng. 2018. Automatic polyp detection via a novel unified bottom-up and top-down saliency approach. *IEEE Journal of Biomedical and Health Informatics* 22, 4 (2018), 1250–1260.

[80] Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. arXiv:1212.5701.

[81] Xu Zhang, Fei Chen, Tao Yu, Jiye An, Zhengxing Huang, Jiquan Liu, Weiling Hu, Liangjing Wang, Huilong Duan, and Jianmin Si. 2019. Real-time gastric polyp detection using convolutional neural networks. *PloS One* 14, 3 (2019), e0214133.

[82] Xu Zhang, Weiling Hu, Fei Chen, Jiquan Liu, Yuanhang Yang, Liangjing Wang, Huilong Duan, and Jianmin Si. 2017. Gastric precancerous diseases classification using CNN with a concise model. *PloS One* 12, 9 (2017), e0185508.

[83] Mingda Zhou, Guanqun Bao, Yishuang Geng, Bader Alkandari, and Xiaoxi Li. 2014. Polyp detection and radius measurement in small intestine using video capsule endoscopy. In *Proceedings of IEEE BMEI2014*. 237–241.

A.10 Paper X - Machine Learning-Based Analysis of Sperm Videos and Participant Data for Male Fertility Prediction

Authors: Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, **Vajira Thambawita**, Pål Halvorsen, Hugo L. Hammer, Trine B. Haugen, Michael A. Riegler

Abstract: Methods for automatic analysis of clinical data are usually targeted towards a specific modality and do not make use of all relevant data available. In the field of male human reproduction, clinical and biological data are not used to its fullest potential. Manual evaluation of a semen sample using a microscope is time-consuming and requires extensive training. Furthermore, the validity of manual semen analysis has been questioned due to limited reproducibility, and often high inter-personnel variation. The existing computer-aided sperm analyzer systems are not recommended for routine clinical use due to methodological challenges caused by the consistency of the semen sample. Thus, there is a need for an improved methodology. We use modern and classical machine learning techniques together with a dataset consisting of 85 videos of human semen samples and related participant data to automatically predict sperm motility. Used techniques include simple linear regression and more sophisticated methods using convolutional neural networks. Our results indicate that sperm motility prediction based on deep learning using sperm motility videos is rapid to perform and consistent. Adding participant data did not improve the algorithms performance. In conclusion, machine learning-based automatic analysis may become a valuable tool in male infertility investigation and research.

Published: Nature scientific reports, 2019

Candidate contributions: Vajira contributed to the conception and design of this article. He experimented with two different deep learning methods (out of four) which are based on dense optical flow and a novel preprocessing technique called "vertical frame matrix" to predict motility values of sperm samples. He performed his experiments with different input types such as pre-processed video frames and participant

data. He contributed to analyzing the results of his methods, drafting the article, and revising it.

Thesis objectives: Sub-objective I, Sub-objective III

Appendix A. Published Articles

OPEN

Machine Learning-Based Analysis of Sperm Videos and Participant Data for Male Fertility Prediction

Steven A. Hicks[1,2*], Jorunn M. Andersen[3,5], Oliwia Witczak[3,5], Vajira Thambawita[1,2], Pål Halvorsen[1,2], Hugo L. Hammer[1,2], Trine B. Haugen[3,6] & Michael A. Riegler[1,4,6]

Methods for automatic analysis of clinical data are usually targeted towards a specific modality and do not make use of all relevant data available. In the field of male human reproduction, clinical and biological data are not used to its fullest potential. Manual evaluation of a semen sample using a microscope is time-consuming and requires extensive training. Furthermore, the validity of manual semen analysis has been questioned due to limited reproducibility, and often high inter-personnel variation. The existing computer-aided sperm analyzer systems are not recommended for routine clinical use due to methodological challenges caused by the consistency of the semen sample. Thus, there is a need for an improved methodology. We use modern and classical machine learning techniques together with a dataset consisting of 85 videos of human semen samples and related participant data to automatically predict sperm motility. Used techniques include simple linear regression and more sophisticated methods using convolutional neural networks. Our results indicate that sperm motility prediction based on deep learning using sperm motility videos is rapid to perform and consistent. Adding participant data did not improve the algorithms performance. In conclusion, machine learning-based automatic analysis may become a valuable tool in male infertility investigation and research.

Automatic analysis of clinical data may open new avenues in medicine, though often limited to one modality, usually images[1]. Recently, however, trends have shifted to include data from other modalities, including sensor data and participant data[2,3]. Furthermore, advancements in artificial intelligence, specifically deep learning, have shown its potential in becoming an essential tool for health professionals through its promising results on numerous use-cases[1,4–6].

Male reproduction is a medical field that is gaining increased attention due to several studies indicating a global decline in semen quality during the last decades[7,8] as well as geographical differences[9]. Semen analysis is a central part of infertility investigation, but the clinical value in predicting male fertility is uncertain[10]. Standard semen analysis should be performed according to the recommendations made by the WHO, which includes methods of assessing semen volume, sperm concentration, total sperm count, sperm motility, sperm morphology, and sperm vitality[11]. Sperm motility is categorized into the percentage of progressive, non-progressive, and immotile spermatozoa. Sperm morphology is classified according to the presence of head defects, neck and midpiece defects, principal piece (main part of the tail) defects, and excess residual cytoplasm in a stained preparation of cells. Figure 1 shows an example of a frame extracted from a video of a wet human semen sample. The WHO has established reference ranges for various semen parameters based on the semen quality of fertile men whose partners had a time to pregnancy up to and including 12 months[12]. However, these ranges can not be used to distinguish fertile from infertile men. Manual semen analysis requires trained laboratory personnel, and even when performed in agreement with the WHOs guidelines, it may be prone to high intra- and inter-laboratory variability.

Attempts to develop automatic systems for semen analysis have been carried out for several decades[13]. CASA was introduced during the 1980s after the digitization of images made it possible to analyze images using a computer. A more rapid and objective assessment of sperm concentration and sperm motility was expected by using CASA, but it has been challenging to obtain accurate and reproducible results[13]. The results may be unreliable

[1]Holistic Systems Department, Simula Metropolitan Center for Digital Engineering, Oslo, Norway. [2]Faculty of Technology, Art and Design, OsloMet – Oslo Metropolitan University, Oslo, Norway. [3]Faculty of Health Sciences, OsloMet – Oslo Metropolitan University, Oslo, Norway. [4]Department of Technology, Kristiania University College, Oslo, Norway. [5]These authors contributed equally: Jorunn M. Andersen and Oliwia Witczak. [6]These authors jointly supervised this work: Trine B. Haugen and Michael A. Riegler. *email: steven@simula.no
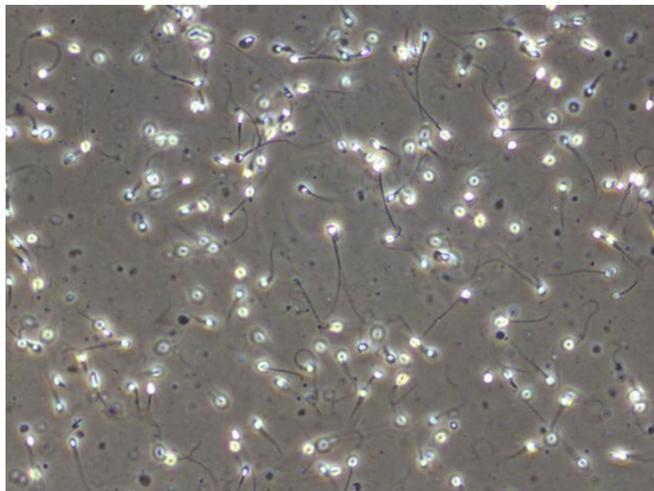
240

Figure 1. Frame from a microscopic video of a human semen sample showing several spermatozoa (Olympus CX31 phase contrast microscope with heated stage, UEye UI-2210C camera, 400x magnification).

due to particles and other cells than spermatozoa in the sample as well as the occurrence of sperm collisions and crossing sperm trajectories. Better results are obtained when analyzing spermatozoa separated from seminal plasma and re-suspended in a medium. CASA was also developed for assessment of sperm morphology and DNA fragmentation in the sperm. It is claimed that new models can also assess vitality and that some functional tests of a semen sample are possible[13]. However, the assessments require special staining or preparation procedures. Despite its long history as a digitized sperm analyzer, CASA is not recommended for clinical use[11,13]. The technology, however, has been improved, and it has been suggested that using CASA for sperm counting and motility assessment can be a useful tool with less analytical variance than the manual methods[14,15].

Concerning automatic semen analysis in general, Urbano et al.[16] present a fully automated multi-sperm tracking algorithm, which can track hundreds of individual spermatozoa simultaneously. Additionally, it is also able to measure motility parameters over time with minimal operator intervention. The method works by applying a modified version of the jpdaf to microscopic semen recordings, allowing them to track individual spermatozoa at proximities and during head collisions (a common issue with existing CASA instruments). The main contribution made by Urbano et al. is the modified jpdaf algorithm for tracking individual spermatozoa, but by only evaluating the proposed approach on two samples, the generalizability of the method to a larger population is difficult to determine.

Dewan et al.[17] present a similar method, tracking spermatozoa by generating trajectories of the cells across microscopic video sequences. Similar to CASA, object proposals are generated through a greyscale edge detection algorithm, which is then tracked to generate object trajectories. These trajectories are then classified into "sperm" or "non-sperm" entities using a CNN, of which the "sperm" entities are used to estimate three quality measurements for motility (progressive, non-progressive, and immotile), and the concentration of spermatozoa per unit volume of semen. The results seem promising but since the method was evaluated on a closed dataset, it is not possible to directly compare this approach with other methods.

Although not the focus in our work, another essential attribute for semen quality is measuring the number of abnormal spermatozoa present in a semen sample. Ghasemian et al.[18] tried to detect abnormal spermatozoa by individually classifying human spermatozoa into normal or abnormal groups. Shaker et al.[19] did a similar study to predict sperm heads as normal or abnormal by splitting images of sperm heads into square patches and using them as training data for a dictionary-based classifier. A common theme is that all automatic approaches, for both motility and morphology assessment, focus on one modality and do not incorporate other data into the analysis. Additionally, the evaluation is performed on a rather limited or closed data which hinders reproducibility and comparability of the results. In the presented work, we aim to contribute to the field of automated semen analysis in the following three ways: (i) to develop a rapid and consistent method for analyzing sperm motility automatically, (ii) to explore the potential of multimodal analysis methods combining video data with participant data to improve the results of the automatic analysis, and (iii) to compare different methods for predicting sperm motility using algorithms based on deep learning and classical machine learning.

To the best our knowledge, no study has been performed on how deep learning and multimodal data analysis may be used to directly analyze semen recordings in combination with participant/patient data for the automated prediction of motility parameters. Using data from 85 participants and three-fold cross-validation, we observe that the initial results are promising. Thus, machine learning-based automatic analysis may become a valuable tool for the future of male infertility investigation.

Appendix A. Published Articles

Methods

Experimental design. Our main approach is the use of CNNs to analyze sequences of frames from video recordings of human semen under a microscope to predict sperm motility in terms of progressive, non-progressive, and immotile spermatozoa. The video recordings are then combined with participant data to see how it may improve our methods using the multiple modalities available in our dataset. As there are no related works for which to compare directly, we first trained a series of machine learning algorithms to set a baseline for how well we can expect our deep learning-based algorithms to perform.

The presentation of our methods is divided into three parts. Firstly, we provide a description of the dataset used for both training and evaluation of the presented methods and the statistical analysis. Secondly, we detail how we trained and evaluated the methods based on classical machine learning algorithms. Lastly, we describe our primary approach of using deep learning-based algorithms to predict sperm motility in terms of progressive, non-progressive, and immotile spermatozoa. All experiments were performed following the relevant guidelines and regulations of the Regional Committee for Medical and Health Research Ethics - South East Norway, and the General Data Protection Regulation(GDPR).

Dataset. For all experiments, we used videos and several variables from the VISEM-dataset[20] [https://data-sets.simula.no/visem/], a fully open and multimodal dataset with anonymized data and videos of semen samples from 85 different participants. In addition to the videos, the selected variables for the analysis included manual assessment of sperm concentration and sperm motility for each semen sample and participant data. Participant data consisted of age, BMI, and days of sexual abstinence. In the experiments, the videos and participant data were used as independent variables whereas the sperm motility values (percentage of progressive, non-progressive sperm motility, and immotile spermatozoa) were used as the dependent variables. We also performed an additional experiment to test the effect of sperm concentration if added as an independent variable to the analysis.

Details on the collection and handling of semen samples have previously been described by Andersen *et al.*[21]. Briefly, the semen samples were collected at a room near the laboratory or at home and handled according to the WHO guidelines[11]. Samples collected at home, were transported close to the body to avoid cooling and analyzed within two hours. Assessment of sperm concentration and sperm motility was performed as described in the WHO 2010 manual[11]. Sperm motility was evaluated using videos of the semen sample, and all samples were assessed by one experienced laboratory technician. $10 \, \mu l$ of semen were placed on a glass slide, covered with a 22×22 mm cover slip and placed under the microscope. Videos were recorded using an Olympus CX31 microscope with phase contrast optics, heated stage (37°C), and a microscope mounted camera (UEye UI-2210C, IDS Imaging Development Systems, Germany). Videos for sperm motility assessment were captured using 400× magnification and stored as AVI files. The recordings vary in length between two to seven minutes with a frame rate of 50 frames-per-second.

Statistical analysis. For all experiments, we report the MAE calculated over three-fold cross-validation to get a more robust and generalizable evaluation. Furthermore, statistical significance was tested by a corrected paired t-test, where a p-value below or equal to 0.05 was considered significant. Usually, t-test is based on the assumption that samples are independent. However, samples in the folds of cross-validation are not independent. Therefore, a fudge factor is needed to compensate for the not independent samples[22]. The significance test showed that all results with an average MAE below 11 are significant improvements compared to the ZeroR baseline. For ZeroR, which is also commonly known as the null model, the cross-validation coefficient is defined with a $Q2$ value of 0. This means that the ZeroR predictions are equal to the average calculated over the entire training dataset.

Baseline machine learning approach. For the machine learning baseline, we relied on a combination of well-known algorithms and handcrafted features. To extract features from the video frames, we used the open-source library Lucene Image Retrieval (LIRE)[23]. LIRE is a Java library that offers a simple way to retrieve images and photos based on color and texture characteristics. We tested all available features (more than 30 different ones) with all machine learning algorithms (more than 40 different ones), but in this work, we only report the features that worked best with our machine learning algorithms, which were the Tamura features. Tamura features (coarseness, contrast, directionality, line-likeness, regularity, and roughness) are based on human visual perception, which makes them very important in image representation. Using the Tamura image features, participant data and a combination of both, we trained different algorithms to perform prediction on the motility variables. We performed a total of three experiments per tested algorithm; one using only Tamura features, one using only participant data, and one combining the Tamura features with the participant data through early fusion.

Since the Tamura features are sparse compared to deep features, we used a slightly different approach for selecting frames from the videos. Each video was represented by a feature vector containing the Tamura features of two frames per second (the first and the middle frame) for the first 60 seconds. In total, we had 120 frames per video and a visual feature space consisting of 2160 feature points. These features were then used to train multiple machine learning algorithms using the WEKA machine learning library[24]. We conducted experiments with all available algorithms, but report only the six best performing ones. The reported algorithms are Simple Linear Regression, Random Forests, Gaussian Process, Sequential Minimal Optimization Regression (SMOreg), Elastic Net, and Random Trees. One limitation of these algorithms is that they are only able to predict one value at a time, meaning we had to run them once for each of the three sperm motility variables.

Deep learning approach. For our primary approach, we use methods based on CNNs to perform regression on the three motility variables. For each deep learning-based experiment, we extracted 250 frame samples (single frames or frame sequences) from each of the 85 videos of our dataset. The reason for only extracting 250 frames
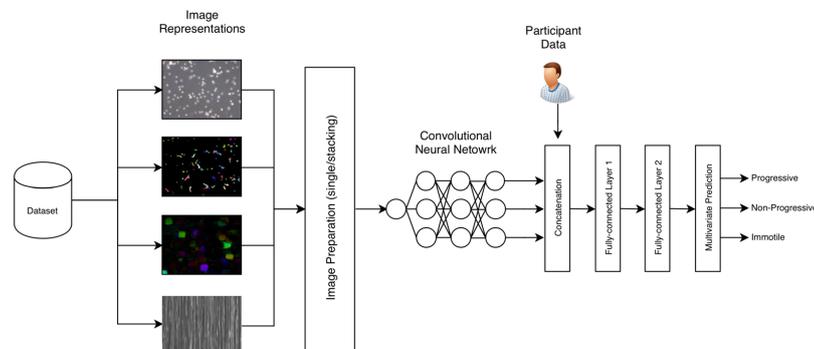
242

Figure 2. The deep learning pipeline used for all multimodal neural network-based experiments. Starting with our dataset, we extract frame data into four different representations. These four different "images" are sent to the image preparation were we either pass a single image or stacked images to a convolutional neural network (CNN). The CNN is trained to learn a model that captures the spatial or spatial and temporal combined features of sperm motility. This is based on the image representation and preparation (stacking or single frame). The output of the CNN model is then combined with the participant data. This combined vector is passed through two fully-connected layers before performing multivariate prediction on the three motility variables.

per video was due to some videos being too short for collecting more than 250 sequences of 30 frames, which is about 7,500 frames equalling about 2 minutes of video at 50 frames-per-second. This results in a total of 21,250 frames used for training and validation. As we are evaluating each method using three-fold cross-validation, the split between the training and validation datasets is 14,166 and 7,083 frame samples, respectively.

Our deep learning approaches can be split into three groups. Firstly, we analyze raw frames as they are extracted from the videos. The analysis is done by looking at the raw pixel values from a single or a sequence of frames and using these to make a prediction. Secondly, we use optical flow to generate temporal representations of frame sequences to condense the information of the temporal dimension into a single image. The advantages of this representation is that it can model the temporal dependencies in the videos, and it is able to alleviate the hardware costs of analyzing raw frame sequences using CNNs. Lastly, we combine the two previous methods to exploit the advantages of both, by using the visual features of raw video frames together with the temporal information of the optical flow representations.

The baseline for the deep learning approaches are the machine learning algorithms as described above and ZeroR. For each experiment, we predict the percentage of progressive spermatozoa, non-progressive spermatozoa, and immotile spermatozoa for a single semen sample. In contrast to the classical machine learning algorithms, neural networks can predict all three values at once. Figure 2 illustrates a high level overview of the complete deep learning analysis pipeline.

All deep learning-based models were trained using mse to calculate loss and Nadam[25] to optimize the weights. The Nadam optimizer had a learning rate of 0.002, β_1 value of 0.900, and β_2 value of 0.999. We trained each model for as long as it improved with a patience value of 20 epochs, meaning if the mse did not improve on the validation set for 20 epochs, we stopped the training to avoid overfitting. The model used for evaluation was the one which performed best on the validation set, not the model from the last epoch. Furthermore, for each method we trained two models. One model uses only frame data, and the other uses a combination of the frame data and the related participant data (BMI, age, and days of sexual abstinence). To include the participant data in the analysis, we first pass a frame sample through the CNN. Then, we take the output of the last convolutional layer and globally average pool it to produce a one-dimensional feature vector which is concatenated with the participant data. This combined vector is then passed through two fully-connected layers consisting of 2,048 neurons each before being making the final prediction (shown in Fig. 2). In the following few sections, we will describe six different methods used to predict sperm motility; a method using single frames for prediction, a method which stacks frames channel-wise, a method using vertical frame matrices, a method based on sparse optical flow, a method based on dense optical flow, and a method based on two-stream networks.

Single frame prediction. For the single frame-based method, we extracted 250 single frames from each video and used this to train various CNNs models based on popular neural network architectures (such as DenseNet[26], ResNet[27], and Inception[28]). We experimented using transfer learning from the ImageNet[29] weights included with the Keras[30] implementations of the different CNN architectures and found that, in general, using these weights as a base for further training worked better than training from scratch. Note that we did not fine-tune the models, meaning we did not freeze any layers during training. We only report the model which performed best, which in our case was a ResNet-50 model implemented in Keras with a TensorFlow[31] back-end. The frames were resized to 224×224 before being passed through the model, which is the recommended size for the ResNet-based architectures[27].
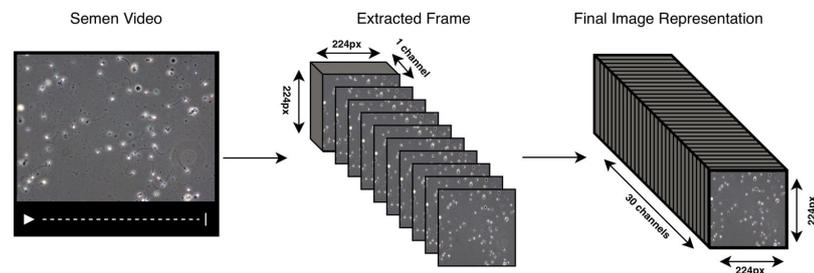
Appendix A. Published Articles

Figure 3. An illustration of how frames are stacked channel-wise after being greyscaled. From a video, a sequence of *n* frames are extracted and greyscaled. These frames are then stacked channel-wise, meaning each frame occupied one channel-dimension of the final image. The final stacked "image" is then of shape $224 \times 224 \times 30$.

The single frame-based approach is simple and comes with some obvious limitations. Most notably, we lose the temporal information present within the video. Losing the temporal information may be acceptable when measuring attributes that rely on visual clues, such as morphology, but for motility the change over time is an important feature.

Greyscale frame stacking. The Greyscale Frame Stacking method is an extension of the single-frame prediction approach. Here, we extract 250 batches of 30 frames and greyscaled them before stacking them channel-wise (shown in Fig. 3). This results in 21,250 frame samples with a shape of $224 \times 224 \times 30$, which contains the information of 30 consecutive frames. The reasons for greyscaling the frames before stacking them is two-fold. Firstly, seeing as the color of the videos are a feature of the microscope and lab preparation, and not the spermatozoon itself, we assume that this feature may confuse the model in unintended ways. Secondly, greyscaling the frames reduces the size of each frame by three, making stacking 30 frames feasible on less powerful hardware. The motivation behind this approach was to keep the temporal information present in a given frame sequence, yet still, keep the size of the input relatively small.

These extracted frame sequences were used to train a ResNet-50 model implemented in Keras[30]. Note that because we changed the size of the channel dimension, we could not perform transfer learning as we did in the previous method. Apart from this, the model was trained in the same manner as described in the beginning of the Deep Learning Approach section.

Vertical frame matrix. To create the vertical frame matrix, 250 batches of 30 frames were extracted and greyscaled. Each frame was resized to 64×64 before being flattened into a one-dimensional vector. The reason for resizing each frame was to keep the length of the flattened images relatively short. With a size of 64×64, the final vector had a length of 4096. Each vector was then stacked on top of each other which resulted in a matrix with a shape of $30 \times 4096 \times 1$. Examples images using this transformation can be seen in row four of Fig. 4. Similar to the Greyscale Frame Stacking approach described in the previous section, we condense the information of multiple frames into a single image, which we can then pass through a standard two-dimensional CNN. Due to size constraints, the model used for this method was ResNet-18. Otherwise, it was trained in the same way as the previous two methods.

Sparse optical flow. For the Sparse Optical Flow approach, we use Lucas-Kanade's[32] algorithm of estimating optical flow. What makes sparse optical flow "sparse," is that we only measure the difference between a few tracked features from one frame to another. In our case, we use Harris and Stephens corner detection algorithm[33] to detect individual sperm heads (implemented in OpenCV[34] as "goodFeaturesToTrack"). Then, we track the progression of each spermatozoon using Lucas-Kanade's algorithm over a sequence of 30 frames. Similar to the previous methods, sequences were sampled at evenly spaced intervals to maximize differences between optical flow representations. We used a CNN model based on the ResNet-50 architecture implemented in Keras and trained using the same configuration described previously. Examples for the sparse optical flow image representation can be seen in row two of Fig. 4.

Dense optical flow. The Dense Optical Flow approach generates optical flow representations using Gunner Farneback's algorithm[35,36] for two-frame motion estimation. Dense optical flow, in contrast to sparse optical flow, processes all pixels of a given image instead of a few tracked features. For this method, we tried two configurations. The first configuration measures the difference between two consecutive frames. The second configuration adds a stride of 10 frames between selected frame samples. This is done to increase the measured difference between frame comparisons. We collected 250 dense optical flow images and trained one model for each of the two configurations to evaluate the result of this method. For both stride configurations, we train each model using the same architecture (ResNet-50) and training configuration as for the other deep learning methods. Examples for the created image representations using the dense optical flow can be seen in row three of Fig. 4.
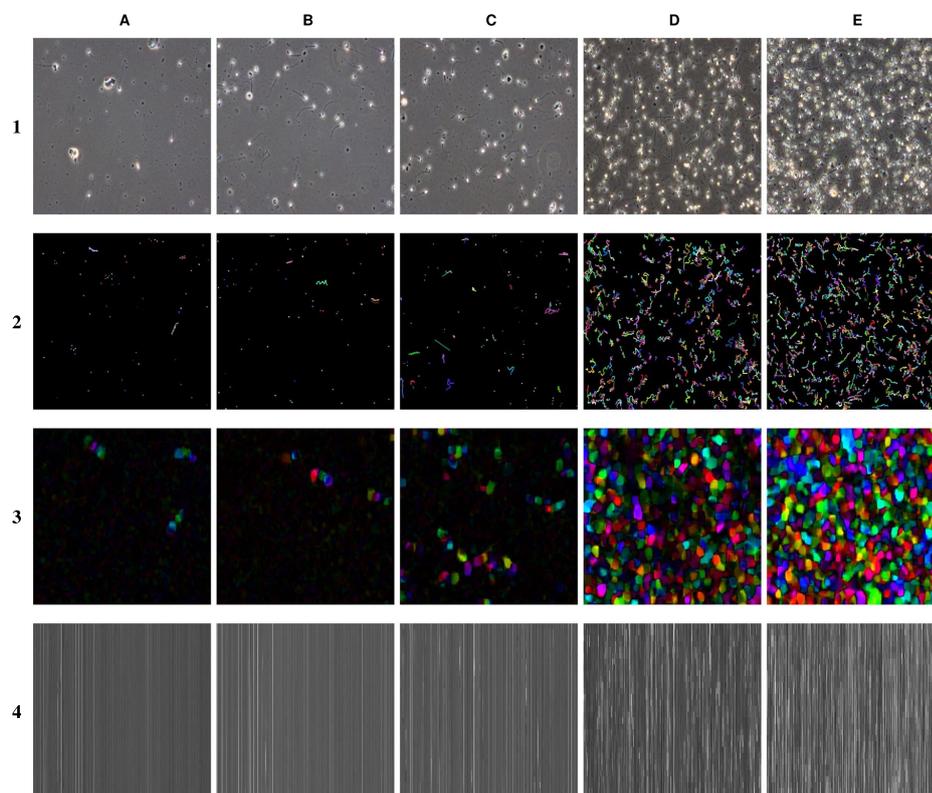
Figure 4. Examples of images from videos of semen samples with different concentrations (columns) and the four image representations used to train the neural network-based algorithms (rows). Sperm concentration; (**A**) 4 per $x106/mL$, (**B**) 33 per $x106/mL$, (**C**) 105 per $x106/mL$, (**D**) 192 per $x106/mL$, and (**E**) 350 per $x106/mL$. Image representation; (1) original video, (2) sparse optical flow, (3) dense optical flow, and (4) vertical frame matrix.

Two-stream network. For the last approach, we combine the two previous methods (visual features of raw frames and the temporal information of optical flow), which is inspired by the work done by Simonyan and Zisserman's[36], where they used a dual-network to perform human action recognition and classification. The model architecture follows a similar structure as described in their article, with the difference being how we input the optical flow representations into the model (we do not stack multiple optical flow representations for different sequences).

Based on this modification, we propose three different methods. Firstly, we use the dual network to analyze one raw video frame in parallel with a Lukas-Kanade sparse optical flow representation of the previous 30 frames. Secondly, we process one raw frame together with a Farneback's dense optical representation. Lastly, we again use one raw frame, but now we combine both the Lukas-Kanade and Farneback's optical flow method by stacking them channel-wise and pass these together through the network. Frames were extracted in the same way as performed for the Single Frame Prediction approach, and the optical flow representations were reused from the Optical Flow-based experiments.

Ethical approval and informed consent. In this study, we used fully anonymized data originally collected based on written informed consent and approval by the Regional Committee for Medical and Health Research Ethics - South East Norway. Furthermore, we confirm that all experiments were performed in accordance with the relevant guidelines and regulations of the Regional Committee for Medical and Health Research Ethics - South East Norway, and the GDPR.

Appendix A. Published Articles

Classical Machine Learning Results				
Method	**Progressive**	**Non-progressive**	**Immotile**	**Average Mean Absolute Error**
Baseline				
ZeroR	17.260	7.860	13.660	12.927
Participant Data Only				
Elastic Net	15.198	9.525	13.441	12.721
Gaussian Process	15.556	9.762	13.474	12.931
Simple Linear Regression	15.416	9.281	13.601	12.766
SMOreg	15.355	9.441	12.959	12.585
Random Forests	**13.312**	**8.886**	**11.905**	**11.368**
Random Tree	17.801	10.952	14.984	14.579
Tamura Image Features Only				
Elastic Net	14.400	7.750	12.190	11.447
Gaussian Process	13.230	7.260	11.920	10.803
Simple Linear Regression	13.520	8.170	12.690	11.460
SMOreg	**13.220**	**7.260**	**11.920**	**10.800**
Random Forests	13.530	7.400	12.060	10.997
Random Tree	18.700	9.960	16.520	15.060
Tamura Image Features and Participant Data				
Elastic Net	14.130	9.890	11.750	11.923
Gaussian Process	13.700	10.120	11.460	11.760
Simple Linear Regression	13.940	10.240	11.410	11.863
SMOreg	13.710	10.140	11.460	11.770
Random Forests	**13.510**	**10.000**	**11.340**	**11.617**
Random Tree	18.660	13.270	16.960	16.297

Table 1. Prediction performance of the machine learning-based methods in terms of mean absolute error for each of the motility values and the overall average. The best performing algorithm in each category is in bold.

Results and Discussion

A complete overview of the results for each method can be seen in Tables 1 and 2. A chart comparing the results is presented in Fig. 5. Table 1 presents the results for the classical machine learning algorithms trained on participant data, Tamura image features, and a combination of the two. For these results, the Gaussian Process, SMOreg, and Random Forests have a MAE below 11, which according to the paired t-test analysis is significant. One interesting finding is that for all cases where participant data is added, the algorithm performs worse. Although a preliminary result, for BMI this is not in line with the finding in our previous work Andersen *et al.*[21], where BMI was found to be negatively correlated with sperm motility using multiple linear regression. However, the methods are very different and therefore not directly comparable. As future work, we plan to perform an extensive analysis of all methodologies on a new dataset. Another interesting insight gained from this experiment is that the Tamura features seem to be well suited for sperm analysis, which will be interesting to investigate more closely.

Since sperm concentration is an important confounding variable when assessing sperm motility by CASA, we performed additional experiments using the two best-performing algorithms to investigate whether or not it had any influence. For the Random Forest, we achieved a MAE of 11.091 when including sperm concentration, compared to 10.996 when we did not. For SMOReg, the MAE was 10.902 with and 10.800 without. This minor difference in error indicates that our method is not gaining or losing any predictive power when including sperm concentration in the analysis, which can be seen as an advantage compared with CASA systems.

To assess the performance of the deep learning-based methods, we used the best performing classical machine learning approach (SMOreg with a MAE of 10.800) and ZeroR as a baseline. In Table 2, the results for single and multimodal deep learning approaches are shown. For most of the experiments, the deep learning models outperform the best machine learning algorithm (SMOreg) by a margin of one or two points. The two methods which are not significant better than ZeroR are the two-stream neural networks, which combined the two optical flow representations in a custom network.

We hypothesize that this is related to the fact that these networks are not able to learn the association between the temporal information of the optical flow and the visual data of the raw frame. Similar to the machine learning algorithms, all methods which combined the participant data with the videos performed worse than those without, leading to the same conclusion as previously discussed. Thus, in our study, adding patient data does not improve the results compared to using only video data, regardless of the algorithms used. If these findings also apply to other patient data needs to be further investigated.

The best performing approaches were a near tie between the method Channel-wise Greyscale and Dense Optical Flow using a stride of 1 or 10 (see Fig. 5). The Channel-wise Greyscale approach achieved a MAE of 8.786, which is two points lower than that of the best performing classical machine learning algorithm (see Table 2). The two Dense Optical Flow methods have the same performance as the Channel-wise Greyscale approach but using one-tenth of the image size, which makes them faster and less computational resource demanding.

Deep Learning Results				
Method	Progressive	Non-progressive	Immotile	Average Mean Absolute Error
Raw Frame Data Approach				
Single Frames (ResNet50)	13.162	8.024	10.967	10.718
Single Frames (ResNet50) + PD	13.659	8.196	12.293	11.383
Channel-wise Greyscale	**10.498**	**7.037**	**8.822**	**8.786**
Channel-wise Greyscale + PD	11.599	7.849	10.132	9.860
Vertical Frame Matrix	11.149	8.218	9.418	9.595
Vertical Frame Matrix + PD	11.182	8.199	9.274	9.552
Optical Flow Approach				
Sparse Optical Flow	11.573	7.263	10.155	9.664
Sparse Optical Flow + PD	12.214	7.760	10.802	10.259
Dense Optical Flow (stride = 1)	**10.191**	**7.114**	**8.914**	**8.740**
Dense Optical Flow (stride = 1) + PD	10.795	7.856	8.745	9.132
Dense Optical Flow (stride = 10)	10.319	7.546	8.782	8.882
Dense Optical Flow (stride = 10) + PD	11.386	7.825	9.734	9.648
Two Stream Network Approach				
Two Stream Sparse	15.888	8.187	13.326	12.467
Two Stream Sparse + PD	16.435	8.197	13.172	12.601
Two Stream Dense (stride = 1)	14.583	7.393	11.996	11.324
Two Stream Dense (stride = 1) + PD	18.166	8.570	15.983	13.940
Two Stream SP + DE (stride = 1)	**11.848**	**7.070**	**10.823**	**9.917**
Two Stream SP + DE (stride = 1) + PD	17.304	8.066	13.783	13.051

Table 2. Prediction performance of the deep learning-based methods in terms of mean absolute error for each of the motility values and overall mean. Note that for each method, we trained two models, one with participant data and one without. Methods which used participant data under training are marked with (+PD). For the methods which use dense optical flow, stride represents the number of frames skipped when comparing the difference of two frames.
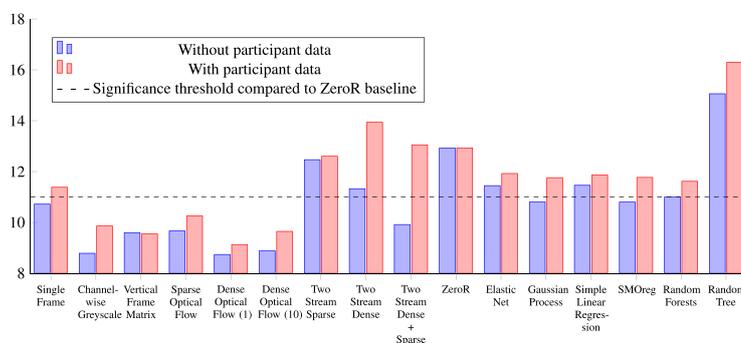
Figure 5. The different machine learning-based algorithms (classical and deep learning) used to predict semen quality in terms of progressive, non-progressive, and immotile spermatozoon. The stippled line represents the threshold for the results to be considered significant compared to the ZeroR baseline. The y-axis does not start at 0 to better highlight the differences. For the methods which used dense optical flow, stride values, how many frames are skipped when comparing two frames, are presented with a *1* or *10* indicating the number of skipped frames. Dense Optical Flow (1) and Channel-wise Greyscale are the best-performing ones but, several of our proposed methods are below the significance threshold.

It is important to point out that the 250 frames used in the analysis were extracted evenly distributed across the entire video length. This means that if there were a noticeable reduction in sperm motility after a certain amount of time, it would be taken into account by the algorithm. The results also support this assumption as the deep learning methods outperformed all classical machine learning methods. This is one of the advantages of the deep learning-based methods presented here.

247

Appendix A. Published Articles

In terms of time needed for the analysis, all presented methods perform the prediction within five minutes, including data preparation which takes most of the time. This is considerably faster than manual sperm motility assessment would be. The classical machine learning methods are faster to train, but in terms of application of the model, the speed is comparable with the deep learning methods.

Conclusion and Future Work

Overall, our results indicate that deep learning algorithms have the potential to predict sperm motility consistently and time efficiently. Multimodal analysis methods combining video data with participant data did not improve the prediction of sperm motility compared to using only the video data. However, it is possible that multimodal analysis using other participant data could improve the prediction. Our results indicate that the deep learning models can incorporate time into their analysis, and therefore are able to predict motility values better than the classical machine learning algorithms. In the future, deep learning-based methods could be used as an efficient support tool for human semen analysis. The presented methods can easily be applied to other relevant assessments such as automatic evaluation of sperm morphology.

Efficient analysis of long videos is a challenge, and future work should focus on how to combine the different modalities of time, imaging, and patient data. The dataset used in this study is also shared openly to ensure comparability and reproducibility of the results. Furthermore, we hope that the methods described in this work will inspire to further development of automatic analysis within the field of male reproduction.

Data availability

References

1. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nat. medicine* **25**, 44 (2019).
2. Boll, S., Meyer, J. & O'Connor, N. E. Health media: From multimedia signals to personal health insights. *IEEE MultiMedia* **25**, 51–60 (2018).
3. Riegler, M. *et al*. Multimedia and medicine: Teammates for better disease detection and survival. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, 968–977, https://doi.org/10.1145/2964284.2976760 (ACM, 2016).
4. Hannun, A. Y. *et al*. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat. Medicine* **25**, 65–69, https://doi.org/10.1038/s41591-018-0268-3 (2019).
5. Esteva, A. *et al*. Dermatologist-level classification of skin cancer with deep neural networks. *Nat.* **542**, 115–118, https://doi.org/10.1038/nature21056 (2017).
6. Pogorelov, K. *et al*. Efficient disease detection in gastrointestinal videos – global features versus neural networks. *Multimed. Tools Appl.* **76**, 22493–22525, https://doi.org/10.1007/s11042-017-4989-y (2017).
7. Carlsen, E., Giwercman, A., Keiding, N. & Skakkebæk, N. E. Evidence for decreasing quality of semen during past 50 years. *Br. Med. J.* **305**, 609–613 (1992).
8. Levine, H. *et al*. Temporal trends in sperm count: a systematic review and meta-regression analysis. *Hum. Reproduction Updat.* **23**, 646–659 (2017).
9. Jørgensen, N. *et al*. East–west gradient in semen quality in the nordic–baltic area: a study of men from the general population in denmark, norway, estonia and finland. *Hum. Reproduction* **17**, 2199–2208 (2002).
10. Tomlinson, M. Uncertainty of measurement and clinical value of semen analysis: has standardisation through professional guidelines helped or hindered progress? *Androl.* **4**, 763–770 (2016).
11. World Health Organization, Department of Reproductive Health and Research. *WHO laboratory manual for the examination and processing of human semen* (Geneva: World Health Organization, 2010).
12. Cooper, T. G. *et al*. World health organization reference values for human semen characteristics. *Hum. Reproduction Updat.* **16**, 231–245, https://doi.org/10.1093/humupd/dmp048 (2010).
13. Mortimer, S. T., van der Horst, G. & Mortimer, D. The future of computer-aided sperm analysis. *Asian journal andrology* **17**, 545 (2015).
14. Dearing, C. G., Kilburn, S. & Lindsay, K. S. Validation of the sperm class analyser casa system for sperm counting in a busy diagnostic semen analysis laboratory. *Hum. Fertility* **17**, 37–44 (2014).
15. Dearing, C., Jayasena, C. & Lindsay, K. Can the sperm class analyser (sca) casa-mot system for human sperm motility analysis reduce imprecision and operator subjectivity and improve semen analysis? *Hum. Fertility* 1–11 (2019).
16. Urbano, L. F., Masson, P., VerMilyea, M. & Kam, M. Automatic tracking and motility analysis of human sperm in time-lapse images. *IEEE Transactions on Med. Imaging* **36**, 792–801, https://doi.org/10.1109/TMI.2016.2630720 (2017).
17. Dewan, K., Rai Dastidar, T. & Ahmad, M. Estimation of sperm concentration and total motility from microscopic videos of human semen samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2018).
18. Ghasemian, F., Mirroshandel, S. A., Monji-Azad, S., Azarnia, M. & Zahiri, Z. An efficient method for automatic morphological abnormality detection from human sperm images. *Comput. methods programs biomedicine* **122**, 409–420 (2015).
19. Shaker, F., Monadjemi, S. A., Alirezaie, J. & Naghsh-Nilchi, A. R. A dictionary learning approach for human sperm heads classification. *Comput. biology medicine* **91**, 181–190 (2017).
20. Haugen, T. *et al*. Visem: A multimodal video dataset of human spermatozoa. In *Proceedings of the ACM Multimedia Systems Conference (MMSYS)*, https://doi.org/10.1145/3304109.3325814 (ACM, 2019).
21. Andersen, J. M. *et al*. Body mass index is associated with impaired semen characteristics and reduced levels of anti müllerian hormone across a wide weight range. *PloS one* **10**, e0130210 (2015).
22. Nadeau, C. & Bengio, Y. Inference for the generalization error. In *Proceeding of the Advances in neural information processing systems (NIPS)*, 307–313 (2000).
23. Lux, M., Riegler, M., Halvorsen, P., Pogorelov, K. & Anagnostopoulos, N. Lire: open source visual information retrieval. In *Proceedings of the ACM Multimedia Systems Conference (MMSYS)*, **30** (2016).
24. Hall, M. *et al*. The WEKA data mining software: an update. *SIGKDD Explor.* **11**, 10–18 (2009).
25. Dozat, T. *Incorporating nesterov momentum into adam*. (2015).
26. Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

9

248

A.10. Paper X - Machine Learning-Based Analysis of Sperm Videos and Participant Data for Male Fertility Prediction

27. He, K., Zhang, X., Ren, S. & Sun, J. *Deep residual learning for image recognition.* 770–778 (2016).
28. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567* (2015).
29. Deng, J. *et al.* ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2009).
30. Chollet, F. *et al. Keras: Deep learning library for theano and tensorflow.* https://keras.io (2015).
31. Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 265–283 (2016).
32. Lucas, B. D. & Kanade, T. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) - Volume 2*, 674–679 (Morgan Kaufmann Publishers Inc., 1981).
33. Harris, C. & Stephens, M. A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference*, 147–151 (1988).
34. Bradski, G. *The OpenCV Library. Dr. Dobb's J. Softw. Tools* (2000).
35. Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In Bigun, J. & Gustavsson, T. (eds) *Image Analysis*, 363–370 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2003).
36. Simonyan, K. & Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 568–576 (2014).

Author contributions

S.A.H., M.A.R. and T.B.H. conceived the experiment(s), S.A.H., V.T. and M.A.R. conducted the experiment(s), all authors analysed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.A.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A.11 Paper XI - Stacked Dense Optical Flows and Dropout Layers to Predict Sperm Motility and Morphology

Authors: Vajira Thambawita, Pål Halvorsen, Hugo Hammer, Michael Riegler, and Trine B. Haugen

Abstract: In this paper, we analyse two deep learning methods to predict sperm motility and sperm morphology from sperm videos. We use two different inputs: stacked pure frames of videos and dense optical flows of video frames. To solve this regression task of predicting motility and morphology, stacked dense optical flows and extracted original frames from sperm videos were used with the modified state of the art convolution neural networks. For modifications of the selected models, we have introduced an additional multi-layer perceptron to overcome the problem of over-fitting. The method which had an additional multi-layer perceptron with dropout layers, shows the best results when the inputs consist of both dense optical flows and an original frame of videos.

Candidate contributions: Vajira contributed to the conception and design of this working-note paper. He conducted all the experiments of this paper using two different deep learning approaches to predict motility and morphology of the given videos of sperm samples by organizers of MediaEval 2019-MedicoTask. He analyzed the results collected from his methods using three-folds cross-validation and presented the results at MedicaEval-2019 and they were the best results from all the participants of Medicotask-2019. Vajira contributed to drafting the paper and revising it.

Thesis objectives: Sub-objective I, Sub-objective III

Stacked Dense Optical Flows and Dropout Layers to Predict Sperm Motility and Morphology

Vajira Thambawita[1,2], Pål Halvorsen[1,2], Hugo Hammer[1,2], Michael Riegler[1,3], Trine B. Haugen[2]

[1]SimulaMet, Norway   [2]Oslo Metropolitan University, Norway   [3]Kristiania University College, Norway

Contact:vajira@simula.no

ABSTRACT

In this paper, we analyse two deep learning methods to predict sperm motility and sperm morphology from sperm videos. We use two different inputs: stacked pure frames of videos and dense optical flows of video frames. To solve this regression task of predicting motility and morphology, stacked dense optical flows and extracted original frames from sperm videos were used with the modified state of the art convolution neural networks. For modifications of the selected models, we have introduced an additional multi-layer perceptron to overcome the problem of over-fitting. The method which had an additional multi-layer perceptron with dropout layers, shows the best results when the inputs consist of both dense optical flows and an original frame of videos.

1 INTRODUCTION

Our main goal of this task is to predict the sperm motility and sperm morphology from videos of sperm samples. In the 2019 Medico task [8], a video dataset was provided with ground truth values of sperm motility such as progressive motility, non-progressive motility, and immotility, and sperm morphology such as head defects, tail defects, and midpiece and neck defects. This task was introduced as completely new this year, and therefore, we could not find any previous work in previous mediaeval Medico task competitions [14, 15]. In this competition, the VISEM dataset [6] which contains sperm videos recorded from 85 participants is used. In the dataset paper, the authors presented baseline mean absolute error values for motility and morphology. Moreover, the importance of computer-aided sperm analysis can be identified from the research works which have been done to develop automatic sperm analysis method in last few decades [3, 13, 19].

Video analysis is a hot research topic in the field of deep learning. Some researchers are experimenting with video classification [2], detection [1], segmentation [5], and generations [12, 18] for various type of video datasets. Yue-Hei Ng et al. [20] experimented with video classification problem using well knows datasets such as sports-1M [10] and UCF101 [16]. In these experiments, they have generated dense optical flow images and row frames of videos to classify 120 seconds long videos. In this paper, we use very short video segments such as nine frames compared to these long segments such as 120s X 30 frames/s.

To solve this new regression problem of predicting morphology and motility from videos of sperm samples, this paper presents two deep learning methods where we used extracted dense optical flows and raw frames from the videos. In Section 2, we are going
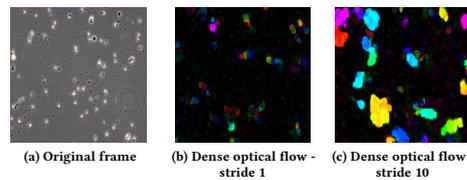
Figure 1: Sample images used to construct input image stacks into the models

to present our two types of input data and two types of methods used in our experiments. Then, the results collected from these experiments will be discussed in Section 3. Finally, the paper ends up with conclusions and future work in Section 4.

2 APPROACH

We have selected the pre-trained ResNet-34 [7] to do some basic experiments of predicting sperm motility and sperm morphology using stacked normal raw video frames and a combination of stacked dense optical flows and raw frames of videos. In this paper, we obtain experimental results using two different types of inputs and from two different types of models.

2.1 Preprocessing data

To find estimates for the sperm motility and sperm morphology, we first preprocessed the input videos to generate two types of input. In the first type **(dataset - D1)**, we stacked nine consecutive frames from a video to make a single input data point. A sample of a raw frame of a video is given in Figure 1a. Before stacking raw video frames, we converted the RGB format frames of the video into grayscale images and resized them into 256x256. These nine frames represent nine different consecutive frames of a video. Moreover, we collected 250 stacked data points (chunks) from 250 locations in time from a video as described above.

For the second type of input **(dataset - D2)**, we generated a tensor with nine channels, which consists of a three-channels (RGB) original video frame (Figure 1a), a three-channels dense optical flow image of stride 1 (Figure 1b), and a three-channels image of dense optical flow of stride 10 (Figure 1c). The dense optical flow image of stride 1 was generated from two consecutive video frames from a selected location of a video. Then, we generated the stride-10 dense optical flow image using two frames; the first frame of the video chuck and the 10th frame of a selected video chunk. To generate dense optical flows [4] of two different frames of a video, the OpenCV library [9] was used with its inbuilt functions.
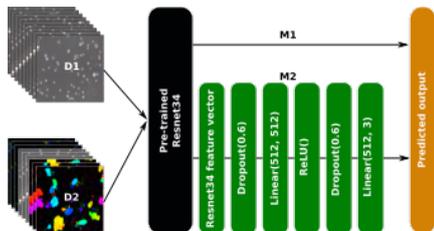
Appendix A. Published Articles

Figure 2: Big picture of our deep learning model: M1 - the base model of Resnet-34 with a three output last layer, M2 - the modified version of Resnet-34 with an additional MLP, D1 and D2 represent the two different types of input used in our experiments.

For both input types, we split the datasets into three folds based on the folds given in the video dataset provided by organizers. Then, a three-fold cross-validation was performed to evaluate our deep learning models which will be introduced in the later sections.

2.2 Deep learning model implementation

For implementation of our deep learning models, we selected Resnet-34 which is larger than the smallest, Resnet-18, and smaller than other large scales Resnet models like Resnet-50, Resnet-101, and Resnet-152. The selections of this intermediate Resnet-34 was done based on expandability of the model by adding additional multi-layer perceptron (MLP) within the available hardware resources (considering memory limitations of the available graphics processing units). In addition to that, the pre-experiments were done to identify over-fitting problems of strong models for simpler predictions and computation time required to finish training. Furthermore, expandability of the number of input channels of the model within the available GPU memory was examined.

For **method 1 (M1)**, we modified the input layer of the selected pre-trained Resnet-34 to take nine channel inputs and modified the last layer of the model to output only three values which are representing either three values of sperm motility or three values of sperm morphology. We used this method as our base model with the two different datasets (D1 and D2) as introduced in Section 2.1 and recorded results collected from this experiment in D1-M1 and D2-M1 rows in Table 1.

In **method 2 (M2)**, to avoid over-fitting problems of this task, we have embedded additional MLP to the end of the network with dropout layers [17]. The full structure of this additional MLP is depicted in Figure 2 using a green colour. The dropout values of this MLP were selected using pre-experiments, and it is a hyper-parameter for this model. The collected results of this method are tabulated in rows D1-M2 and D2-M2 of Table 1.

In the training process of all the above methods, the Adam optimizer [11] with a learning rate 0.001 was used. The mean square error (MSE) was used as the loss function for back-propagating error, and mean absolute error (MAE) was used for calculating the actual loss of predictions based on ground truth values of motility and morphology.

Table 1: MAE values collected from the proposed methods: D1-stacked gray-scale nine consecutive frames, D2-stacked an original frame + a dense optical flow image from two consecutive frames + a dense optical flow from two frames with stride=10; M1 - the basic model of Resnet-34 with modifications of number of input channels and outputs, M2 - the modified model with an additional MLP with dropout layers

Input	Method	Fold	Motility MAE	Motility Average	Morphology MAE	Morphology Average
D1	M1	Fold 1	9.562		5.626	
		Fold 2	8.959	9.200	5.749	5.649
		Fold 3	9.079		5.573	
	M2	Fold 1	9.585		5.424	
		Fold 2	9.28	9.185	5.382	5.394
		Fold 3	8.689		5.375	
D2	M1	Fold 1	9.044		5.933	
		Fold 2	8.062	9.372	5.394	5.525
		Fold 3	11.01		5.248	
	M2	Fold 1	8.612		5.549	
		Fold 2	7.873	**8.825**	5.463	**5.293**
		Fold 3	9.991		4.868	

3 RESULTS AND ANALYSIS

According to the average MAE values shown in Table 1, the M2 method with the input type 2 (D2) shows best results among other methods and other input types. This method shows the best MAE value of 8.825 for the sperm motility and 5.293 for the sperm morphology. This improvement of error values can be seen as results of accumulated benefits of showing pre-processed temporal information such as dense optical flows to the model and the additional MLP to overcome the problem of over-fitting. Moreover, the added MLP in M2 gives better results with both input types (D1 and D2) for both predictions: sperm motility and sperm morphology. We achieved this performance as a result of the pre-processed input data with dense optical flows and the MLP introduced to overcome the over-fitting problem.

4 CONCLUSION AND FUTURE WORK

The input with a raw frame and dense optical flows of two difference stride values show better results compared to the stacked normal frames of videos. Moreover, the modified Resnet-34 model with an MLP which consists of dropout layers with high probabilities did achieve better results than the base model in the both cases because it helped to overcome the problem of over-fitting in the training stage. Finally, the combination of the input with dense optical flows and the modified Resnet-34 with an additional MLP shows the best overall performance.

In future work, it is worth to try CNN models with long short-term memory units to capture temporal features of video frames. Moreover, a 3D CNN can be a promising approach for this kind of task because 3D CNN models have capabilities to capture temporal information of videos.

REFERENCES

[1] Alan C Bovik. 2010. *Handbook of image and video processing*. Academic press.

[2] Darin Brezeale and Diane J Cook. 2008. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 3 (2008), 416–430.

[3] Karan Dewan, Tathagato Rai Dastidar, and Maroof Ahmad. 2018. Estimation of Sperm Concentration and Total Motility From Microscopic Videos of Human Semen Samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[4] Gunnar Farnebäck. 2003. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the Scandinavian conference on Image analysis*. Springer, 363–370.

[5] Arun Hampapur, Terry Weymouth, and Ramesh Jain. 1994. Digital video segmentation. In *Proceedings of the second ACM international conference on Multimedia*. ACM, 357–364.

[6] Trine B. Haugen, Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, Hugo L. Hammer, Rune Borgli, Pål Halvorsen, and Michael A. Riegler. 2019. VISEM: A Multimodal Video Dataset of Human Spermatozoa. In *Proceedings of the 10th ACM on Multimedia Systems Conference (MMSys'19)*. ACM, New York, NY, USA. https://doi.org/10.1145/3304109.3325814

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Steven Hicks, Pål Halvorsen, Trine B Haugen, Jorunn M Andersen, Oliwia Witczak, Konstantin Pogorelov, Hugo L Hammer, Duc-Tien Dang-Nguyen, Mathias Lux, and Michael Riegler. 2019. Medico Multimedia Task at MediaEval 2019. In *CEUR Workshop Proceedings - Multimedia Benchmark Workshop (MediaEval)*.

[9] Itseez 2014. *The OpenCV Reference Manual* (2.4.9.0 ed.). Itseez.

[10] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[11] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[12] Yitong Li, Martin Renqiang Min, Dinghan Shen, David Carlson, and Lawrence Carin. 2018. Video generation from text. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[13] Sharon T Mortimer, Gerhard van der Horst, and David Mortimer. 2015. The future of computer-aided sperm analysis. *Asian journal of andrology* 17, 4 (2015), 545.

[14] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Steven Alexander Hicks, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, Olga Ostroukhova, and Thomas de Lange. 2018. Medico Multimedia Task at MediaEval 2018.. In *Proceedings of the CEUR Workshop on Multimedia Benchmark Workshop (MediaEval)*.

[15] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Carsten Griwodz, Thomas Lange, Kristin Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, Mathias Lux, and others. 2017. Multimedia for medicine: the medico task at MediaEval 2017. (2017).

[16] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15 (2014), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

[18] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. 2018. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1526–1535.

[19] L. F. Urbano, P. Masson, M. VerMilyea, and M. Kam. 2017. Automatic Tracking and Motility Analysis of Human Sperm in Time-Lapse Images. *IEEE Transactions on Medical Imaging* 36, 3 (March 2017), 792–801. https://doi.org/10.1109/TMI.2016.2630720

[20] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. 2015. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4694–4702.

A.12 Paper XII - Extracting Temporal Features into a Spatial Domain Using Autoencoders for Sperm Video Analysis

Authors: Vajira Thambawita, Pål Halvorsen, Hugo Hammer, Michael Riegler, Trine B. Haugen

Abstract: In this paper, we present a two-step deep learning method that is used to predict sperm motility and morphology-based on video recordings of human spermatozoa. First, we use an autoencoder to extract temporal features from a given semen video and plot these into image-space, which we call feature-images. Second, these feature-images are used to perform transfer learning to predict the motility and morphology values of human sperm. The presented method shows it's capability to extract temporal information into spatial domain feature-images which can be used with traditional convolutional neural networks. Furthermore, the accuracy of the predicted motility of a given semen sample shows that a deep learning-based model can capture the temporal information of microscopic recordings of human semen.

Published: In the Proceedings of MediaEval 2019.

Candidate contributions: Vajira contributed to the conception and design of this paper. He introduced a novel architecture to extract temporal and spatial features of sperm video using auto-encoder-based architecture. Using the extracted features, Vajra predicted motility and morphology levels of a given sperm sample video. Additionally, he critically evaluated results using two different baseline experiments and two different input shapes. He contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective I, Sub-objective III

Extracting Temporal Features into a Spatial Domain Using Autoencoders for Sperm Video Analysis

Vajira Thambawita[1,2], Pål Halvorsen[1,2], Hugo Hammer[1,2], Michael Riegler[1,3], Trine B. Haugen[2]

[1]SimulaMet, Norway   [2]Oslo Metropolitan University, Norway   [3]Kristiania University College, Norway

Contact:vajira@simula.no

ABSTRACT

In this paper, we present a two-step deep learning method that is used to predict sperm motility and morphology based on video recordings of human spermatozoa. First, we use an autoencoder to extract temporal features from a given semen video and plot these into image-space, which we call feature-images. Second, these feature-images are used to perform transfer learning to predict the motility and morphology values of human sperm. The presented method shows it's capability to extract temporal information into spatial domain feature-images which can be used with traditional convolutional neural networks. Furthermore, the accuracy of the predicted motility of a given semen sample shows that a deep learning-based model can capture the temporal information of microscopic recordings of human semen.

1 INTRODUCTION

The 2019 Medico task [7] focuses on automatically predicting semen quality based on video recordings of human spermatozoa. This is change from previous years which have mainly focused on image classification of images taken from the gastrointestinal tract [10, 11]. For this year's task, we look at predicting the morphology and motility of a given semen sample. Motility is defined by three variables, namely, the percentage of progressive, non-progressive, and immotile sperm. Morphology is determined by the percentage of sperm with tail defects, midpiece defects, and head defects. The organizers have provided a dataset consisting of 85 videos of different semen samples and a preliminary analysis of each, which is used as the ground truth. For this competition, the organizers have provided a predefined three-fold split of the VISEM dataset [5], which contains 85 videos from different participants and a preliminary analysis of each semen sample. In the dataset paper, the authors presented baseline mean absolute error (MAE) values for motility and morphology. Furthermore, the importance of computer-aided sperm analysis can be identified from the previous works which have been done over the last few decades [3, 9, 12].

To solve this year's task, we propose a deep learning-based method consisting of two steps - (i) unsupervised feature extraction using an autoencoder [1] and (ii) video regression using a standard convolutional neural networks (CNN) and transfer learning. The autoencoder we use is different from the state-of-the-art autoencoders used to extract video features [2, 13] as they use autoencoders to extract feature vectors which are used with long-short memory models or multi-layer perceptron (MLP)s. In contrast, we use autoencoders to extract feature-images for use in CNNs.

2 APPROACH

Our method can primarily be split into two distinct steps. First, we use an autoencoder to extract temporal features from multiple frames of a video into a feature-image. Second, we pass the extracted feature-image into a standard pre-trained CNN to predict the motility and morphology of the spermatozoa in a given video. In this paper, we present the preliminary results for four experiments based on four different input types. The first input type (I1) uses a single raw frame. Input type two (I2) is a stack of identical frames copied across the channel-dimension. The third (I3) and fourth (I4) input type stack 9 and 18 consecutive frames from a video respectively.

The first two experiments (using I1 and I2) were performed as baseline experiments. The two other experiments (using I3 and I4) were performed to see how the temporal information affects the prediction performance of the approach. For all input types, we split the extracted datasets into three folds based on the folds provided by the organizers. Then, three-fold cross-validation was conducted to evaluate our four experiments. An overview of all experiments is shown in Figure 1.

2.1 Step 1 - Unsupervised temporal feature extraction

In step 1, we trained an autoencoder that takes an input frame or frames (I1, I2, I3 or I4) from the sperm videos as depicted in Figure 1. Then, the encoder of the autoencoder extracted feature-images and passed them through the decoder architecture to reconstruct the input frame or frames back (R1, R2, R3, and R4). These extracted feature-images are different from traditional feature extractions of autoencoders because the traditional autoencoders extract feature vectors instead of feature-images. In this autoencoder, the mean square error (MSE) loss function is used to calculate the difference between input data and reconstructed data. Then, this error value is backpropagated to train the autoencoder. After training 2,000 epochs, we use the encoder architecture of the autoencoder model to step 2.

2.2 Step 2 - CNN regression model

We have selected the pre-trained ResNet-34 [6] as our basic CNN to predict the values of motility and morphology of the sperm videos. However, any pre-trained CNN could be chosen for this step and in future work we will test and compare different ones in more detail. Firstly, we take an input frame or frames (I1, I2, I3 or I4) and pass through the pre-trained encoder model (only the encoder section of the autoencoder model) which was trained also from the same data inputs in an unsupervised way. Then, the outputs of the encoder model were passed through the CNN model which has a modified last layer to output three prediction values for motility or morphology.
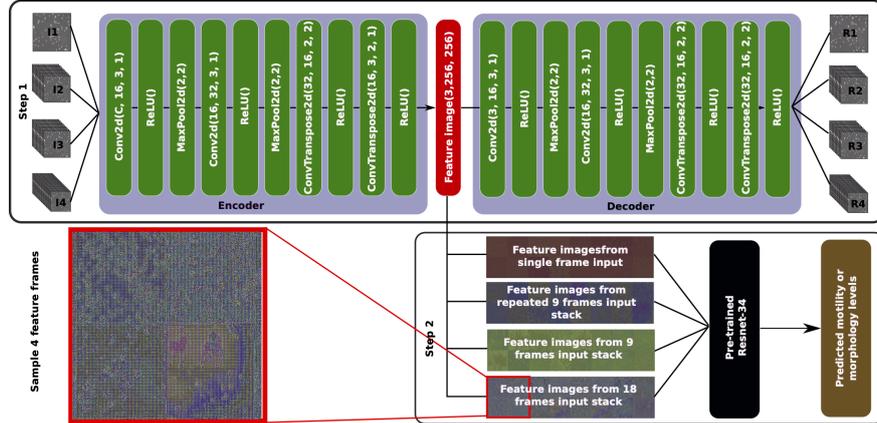
Thambawita et al.

Figure 1: A big picture overview of our two step deep learning model: Step 1 - an autoencoder architecture used to extract image features, Step 2 - the pre-trained Resnet-34 CNN for predicting the regression values of motility and morphology, I1, I2, I3 and I4 - input frames extracted from the video dataset, R1, R2, R3 and R4 - reconstructed data corresponding to the input data I1, I2 I3 and I4, sample 4 feature frames shows extracted 4 feature images from the autoencoder after training 2000 epochs (actual resolution of a feature image is 256X256 which is equal to the original frame size of the input data)

3 RESULTS AND ANALYSIS

According to the average MAE values shown in Table 1, the average motility values of input I3 and I4 shows the best results among other motility values of input I1 and I2. These performance improvements imply that our model is able to learn temporal features into a spatial feature image representation. Furthermore, input I4 which uses 18 stacked frames shows the best motility average values compared to input I3. This performance gain shows that to predict the sperm motility in sperm videos, it is better to analyze more frames at the same time. This might be due to the fact that the behaviour of sperm is something that needs to be observed over time and not in single frames. Moreover, the predictions for our base case inputs I1 and I2 show the same average values. This shows that our model learns temporal information from different sperm video frames. Otherwise, it would be shown different average values for our two base case inputs I1 and I2.

When we consider the predicted morphology average in Table 1, it shows values that are almost equal to each other. This is expected because the morphology of a sperm is something that can be observed using a single frame. In contrast to predicting accurate morphology, the predicted morphology values support the prove that our model has the capability to learn temporal data from multiple frames because motility predictions show an improvement when we increase the number of frames analyzed simultaneously.

4 CONCLUSION AND FUTURE WORKS

In this paper, we proposed a novel method to extract temporal features from videos to create feature-images, which can be used to train traditional CNN models. Furthermore, we show that the

Table 1: Mean absolute error values collected from the proposed method from different inputs: I1, I2, I3 and I4

Input	Fold	Motility		Morphology	
		MAE	Average	MAE	Average
I1	Fold 1	13.330		5.698	
	Fold 2	12.880	13.017	5.748	5.715
	Fold 3	12.840		5.698	
I2	Fold 1	12.890		5.573	
	Fold 2	13.010	13.017	5.593	**5.606**
	Fold 3	13.150		5.653	
I3	Fold 1	10.850		5.567	
	Fold 2	11.310	10.970	5.748	5.632
	Fold 3	10.750		5.580	
I4	Fold 1	9.462		5.900	
	Fold 2	9.426	**9.427**	5.738	5.777
	Fold 3	9.393		5.692	

feature-images capture temporal present in a sequence of frames, which can be used to predict the motility of the sperm videos.

This method can be improved by using different error functions to force the model to learn more temporal data. For example, researchers can experiment with variational autoencoders [8] and generative adversarial learning methods [4] to improve this technique. Additionally, it may be beneficial to embed long short-term memory units to investigate how our feature-images compare to actual extracted temporal features.

A.12. Paper XII - Extracting Temporal Features into a Spatial Domain Using Autoencoders for Sperm Video Analysis

REFERENCES

[1] Pierre Baldi. 2012. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*. 37–49.

[2] Yong Shean Chong and Yong Haur Tay. 2017. Abnormal event detection in videos using spatiotemporal autoencoder. In *Proceedings of the International Symposium on Neural Networks*. Springer, 189–196.

[3] Karan Dewan, Tathagato Rai Dastidar, and Maroof Ahmad. 2018. Estimation of Sperm Concentration and Total Motility From Microscopic Videos of Human Semen Samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of the Advances in neural information processing systems (NIPS)*. 2672–2680.

[5] Trine B. Haugen, Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, Hugo L. Hammer, Rune Borgli, Pål Halvorsen, and Michael A. Riegler. 2019. VISEM: A Multimodal Video Dataset of Human Spermatozoa. In *Proceedings of the 10th ACM on Multimedia Systems Conference (MMSys) (MMSys'19)*. ACM, New York, NY, USA. https://doi.org/10.1145/3304109.3325814

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 770–778.

[7] Steven Hicks, Pål Halvorsen, Trine B Haugen, Jorunn M Andersen, Oliwia Witczak, Konstantin Pogorelov, Hugo L Hammer, Duc-Tien Dang-Nguyen, Mathias Lux, and Michael Riegler. 2019. Medico Multimedia Task at MediaEval 2019. In *Proceedings of the CEUR Workshop on Multimedia Benchmark Workshop (MediaEval)*.

[8] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[9] Sharon T Mortimer, Gerhard van der Horst, and David Mortimer. 2015. The future of computer-aided sperm analysis. *Asian journal of andrology* 17, 4 (2015), 545.

[10] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Steven Alexander Hicks, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, Olga Ostroukhova, and Thomas de Lange. 2018. Medico Multimedia Task at MediaEval 2018.. In *Proceedings of the CEUR Workshop on Multimedia Benchmark Workshop (MediaEval)*.

[11] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Carsten Griwodz, Thomas Lange, Kristin Randel, Sigrun Eskeland, Dang Nguyen, Duc Tien, Mathias Lux, and others. 2017. Multimedia for medicine: the medico task at MediaEval 2017. (2017).

[12] L. F. Urbano, P. Masson, M. VerMilyea, and M. Kam. 2017. Automatic Tracking and Motility Analysis of Human Sperm in Time-Lapse Images. *IEEE Transactions on Medical Imaging (T-MI)* 36, 3 (March 2017), 792–801. https://doi.org/10.1109/TMI.2016.2630720

[13] Huan Yang, Baoyuan Wang, Stephen Lin, David Wipf, Minyi Guo, and Baining Guo. 2015. Unsupervised extraction of video highlights via robust recurrent auto-encoders. In *Proceedings of the IEEE international conference on computer vision (ICCV)*. 4633–4641.

A.13 Paper XIII - ACM Multimedia BioMedia 2020 Grand Challenge Overview

Authors: Steven A. Hicks, **Vajira Thambawita**, Hugo L. Hammer, Trine B. Haugen, Jorunn M. Andersen, Oliwia Witczak, Pål Halvorsen, and Michael A. Riegler.

Abstract: The BioMedia 2020 ACM Multimedia Grand Challenge is the second in a series of competitions focusing on the use of multimedia for different medical use-cases. In this year's challenge, participants are asked to develop algorithms that automatically predict the quality of a given human semen sample using a combination of visual, patient-related, and laboratory-analysis-related data. Compared to last year's challenge, participants are provided with a fully multimodal dataset (videos, analysis data, study participant data) from the field of assisted human reproduction. The tasks encourage the use of the different modalities contained within the dataset and finding smart ways of how they may be combined to further improve prediction accuracy. For example, using only video data or combining video data and patient-related data. The ground truth was developed through a preliminary analysis done by medical experts following the World Health Organization's standard for semen quality assessment. The task lays the basis for automatic, real-time support systems for artificial reproduction. We hope that this challenge motivates multimedia researchers to explore more medical-related applications and use their vast knowledge to make a real impact on people's lives.

Published: Proceedings of the 28th ACM International Conference on Multimedia

Candidate contributions: Vajira contributed to revising and drafting the manuscript.

Thesis objectives: Sub-objective I, Sub-objective III

ACM Multimedia BioMedia 2020 Grand Challenge Overview

Steven A. Hicks*
SimulaMet, Norway

Vajira Thambawita*
SimulaMet, Norway

Hugo L. Hammer†
Oslo Metropolitan University, Norway

Trine B. Haugen
Oslo Metropolitan University, Norway

Jorunn M. Andersen
Oslo Metropolitan University, Norway

Oliwia Witczak
Oslo Metropolitan University, Norway

Pål Halvorsen*
SimulaMet, Norway

Michael A. Riegler
SimulaMet, Norway

ABSTRACT

The BioMedia 2020 ACM Multimedia Grand Challenge is the second in a series of competitions focusing on the use of multimedia for different medical use-cases. In this year's challenge, participants are asked to develop algorithms that automatically predict the quality of a given human semen sample using a combination of visual, patient-related, and laboratory-analysis-related data. Compared to last year's challenge, participants are provided with a fully multimodal dataset (videos, analysis data, study participant data) from the field of assisted human reproduction. The tasks encourage the use of the different modalities contained within the dataset and finding smart ways of how they may be combined to further improve prediction accuracy. For example, using only video data or combining video data and patient-related data. The ground truth was developed through a preliminary analysis done by medical experts following the World Health Organization's standard for semen quality assessment. The task lays the basis for automatic, real-time support systems for artificial reproduction. We hope that this challenge motivates multimedia researchers to explore more medical-related applications and use their vast knowledge to make a real impact on people's lives.

CCS CONCEPTS

• **Applied computing** → **Consumer health**; **Health informatics**; • **Computing methodologies** → *Supervised learning*; • **Information systems** → Summarization.

KEYWORDS

Male fertility, Semen Analysis, Spermatozoa, Machine Learning, Artificial Intelligence

---

*Also affiliated with Oslo Metropolitan University, Norway
†Also affiliated with SimulaMet, Norway

---

1 INTRODUCTION

The BioMedia 2020 ACM Multimedia Grand Challenge[1] is a competition that aims to introduce multimedia researchers to different medical-related tasks that solve real-world challenges. Last year [10], the goal was to automatically analyze images and videos taken from routine investigations of the human digestive tract in order to identify disease, anatomical landmarks, or other relevant findings. This year, we move the focus to assisted reproduction and how multimedia researchers can aid in the development of tools that help determine the quality of a given semen sample. This challenge is the next version of the 2019 *Medico: Multimedia for Medicine* task, which has previously been held at MediaEval Benchmark [10]. Furthermore, the challenge was discussed in the tutorial *Medical Multimedia Systems and Applications* [8] at ACM Multimedia 2019, where attendees showed much interest in this multimodal problem. As the task requires an analysis of a combination of different data modalities, we find that it is a perfect fit for ACM Multimedia and that there is a high chance of making a real impact on the field of human reproduction.

Assessment of semen is usually performed during the early stages of male infertility testing. This is mostly a manual process, where trained clinicians inspect semen samples through a microscope and count the number of different sperms with certain qualities, like the number of moving or not moving sperm, to evaluate quality [4]. As one might expect, this is a tedious and time-consuming process that could greatly benefit from some automation. Furthermore, due to the subjective nature of manually inspecting large numbers of moving sperm, there is a large inter- and intra-observer variability between and within clinics. Having an algorithm that can give consistent results for any given semen sample would be of great benefit. This year, we present four different tasks, each targeting a different aspect of sperm quality assessment. The first two tasks relate to predicting common measurements used for general fertility testing, specifically the motility (movement) and morphology (shape and structure) of the spermatozoa (living sperm). These two tasks encourage participants to combine all available data modalities and data sources to make predictions. Besides
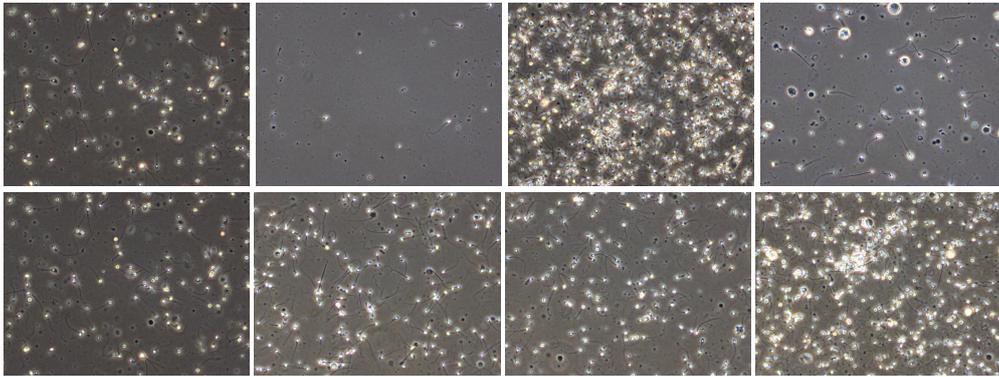
---

[1]https://www.biomediachallenge.com

Appendix A. Published Articles

Figure 1: Sample video frames taken from the dataset. Note the variety of number of cells in the different video samples.

the quantitative and measurable tasks, we also have a more subjective part (optional). Participants are asked to come up with a tool that can help medical doctors choose an ideal sperm for assisted reproduction. This could be as simple as providing an interface to aid in manual tracking to more advanced methods such as unsupervised spermatozoa tracking. The winner of this sub challenge will be chosen by the majority vote of three andrology experts through testing the tool in a clinical setting. The assessment will primarily be based on the usefulness and novelty of the idea.

We provide a dataset consisting of 85 videos, a set of sperm characteristics (such as hormones or fatty acids data), anonymized study participants-related data, and preliminary analysis data collected according to the World Health Organization's standards [20] (ground truth for sperm quality assessment). It is a typical assumption that visual analysis, as provided by the computer vision and medical image processing communities today, is capable of already providing viable and practical approaches to healthcare multimedia challenges. Automatic analysis of human semen is an active field of research supported by several studies on the topic [1, 3, 6, 7, 11, 15, 17]. However, a common theme is that approaches usually focus on one modality and do not incorporate other data sources into their analysis. Although we concede that these methods are indeed essential contributors to promising approaches, we realize the limitations of analyzing images and videos independently in medical fields, such as endoscopy or ultrasound, because of the complexity and needs of medical experts and patients. Neither does it make enough use of the multitude of additional information sources, including sensors and temporal information. The challenge can be seen as very challenging and hard to solve. Due to its novel use-case, we hope to motivate many researchers to look into the field of medical multimedia [16]. Performing research that can have a societal impact will be an essential part of multimedia research. We hope that the challenge can help raise awareness of the topic and provide exciting and meaningful use-cases to researchers.

2 DATASET DETAILS

The challenge uses an open dataset called VISEM [9], which contains data from 85 free-willing male participants aged 18 years or older. For each participant, the dataset presents a set of data collected from a standard semen analysis, which includes a video of the spermatozoa under a microscope, a sperm fatty acid profile, the fatty acid composition of serum phospholipids, participants-related data, and set of preliminary analysis data collected in accordance to the WHO guidelines. The dataset contains a wide variety of samples with varying quality. As we can see in Figure 1, samples vary from containing just a few cells to containing a lot.

The dataset contains over 35 gigabytes of videos, with each video lasting between two to seven minutes. Each video has a resolution of 640×480 and runs at 50 frames-per-second. The dataset contains six CSV-files, a description file, and a folder containing the videos. Each video file's name contains the videos ID, the date it was recorded, and a small description. Then, the end of the filename contains the code of the person who assessed the video. Furthermore, VISEM contains five CSV-files for each category of sensor/patient-related data, a CSV-file with the IDs linked to each their video, and a text file containing descriptions of some of the columns of the CSV-files. One row in each CSV-file represents one participant. In addition to the videos, we provide pre-extracted features for all videos in the dataset.

The features contain a collection of visual features (Tamura, JCD, Edge histogram, and color histogram [14, 18]). All visual features were extracted from the first two frames of every second for sixty seconds of each video. To extract the features, we use the open-source library LIRE [14]. For the final evaluation, participants were asked to use three-fold cross-validation on a pre-defined split of the dataset to calculate the final results. The VISEM dataset [9] is publicly available[2] for participants and other multimedia researchers without any restriction. All study participants agreed to donate their data for science and provided the necessary consent for us to

---

[2]https://datasets.simula.no/visem

be able to distribute the data (checked and approved by Norwegian data authority and ethical committee).

3   TASKS AND EVALUATION METRICS

This year, we present four different tasks, of which two are required to participate in the challenge. For evaluating the first two tasks, we primarily use the mean absolute error (MAE) to compare submissions and report a series of other metrics that the participants may decide to use in their report. For both tasks, we asked participants to perform video analysis over single frame analysis. This is important since single frame-based analysis will not be able to catch the movement of the spermatozoa, which contains essential information to perform the predictions. For the optional challenges, we use manual evaluation with the help of three experts within human reproduction. The experts will deliver a subjective assessment of the submissions based on the delivered result's clinical viability. The script used to evaluate all quantitative submissions is published online for transparency and reproducibility[3].

3.1   Prediction of Motility Task

One of the most common measurements used to evaluate a person's semen quality is looking at the motility of the spermatozoa contained within. Healthy spermatozoon should move at a rapid pace in a forward trajectory. If not, it may be a sign that the sperm will not perform as well. This task aims to predict the percentage of progressive, non-progressive, and immotile sperm in a given semen sample. Motility is the ability of an organism to move independently. Where a progressive spermatozoon can "move forward", a non-progressive would move in circles without progression. An immotile sperm does not move at all. The number of progressive, non-progressive, and immotile spermatozoon should sum up to 100%, meaning every sperm in the sample should be counted.

Submissions to this task should be a CSV file containing the percentage of progressive, non-progressive, and immotile spermatozoon for a given sample. The prediction needs to be performed sample wise resulting in one set of predictions per video sample. No sperm tracking or bounding boxes are required to solve the task. The goal is to maximize the algorithm's performance in terms of prediction accuracy. In this context, accuracy will be evaluated based on the achieved MAE, which is calculated over three-fold cross-validation on the provided test dataset split.

3.2   Prediction of Morphology Task

Morphology is a branch of biology dealing with the study of an organism's form and structural features. In the context of semen, doctors often examine the three parts that make up a spermatozoon, which includes the head, midpiece, and tail, to see if there are any defects. Defects are common and could hamper the spermatozoa ability to move, making it less likely to fertilize. This task should predict the percentage of sperm with head defects, midpiece defects, and tail defects.

Similar to the motility task, the submission to this task should be a CSV file containing the percentage of cells with head defects, percentage of cells with midpiece defects, and the percentage of cells with tail defects. Morphology analysis only requires sample

---

[3]https://github.com/simula/biomedia-2020-submission-evaluation

wise prediction resulting in one value per sample per predicted attribute, meaning one set of predictions per video sample. No sperm tracking or bounding boxes are required to solve this task. For evaluation, we apply the same principals as described under the prediction of motility task.

3.3   Unsupervised Sperm Tracking Task

Finding an optimal sperm for use in assisted reproduction can be crucial for fertilization success. However, detecting the "correct" spermatozoon can be difficult as they can move fast and be surrounded by spermatozoa and other debris. In the first optional task of the challenge, we ask participants to find the spermatozoon that moves faster than all others. This task requires that task participants track the spermatozoa. Within the tracked ones, the fastest is defined as either:

(1) *Fastest average speed* - The spermatozoon that moves the longest distance during the video (total distance/length of the video). This can then be calculated summarizing the different positions (in pixels) between each frame and divide on the number of frames.
(2) *Highest top speed* - The spermatozoon that has the highest average speed across the entire duration of the video. This can be calculated using the maximum of the differences between frames.

A challenge with this task is that the videos change view throughout the sample. This happens because the sample is moved below the microscope to observe the complete sample area. Therefore, the tracking has to be performed per viewpoint on the sample. We decided to keep the submission format to this task quite open, and leave it to the participants to deliver the information in the way that they see fit. Three separate andrology experts will do the evaluation, where they emphasize the correctness of the selected spermatozoa and how well the information is presented.

3.4   Sperm Tracking Support Tool

As previously mentioned, the current procedure for analyzing semen is through manual counting of spermatozoa, which is both tedious and time-consuming. To alleviate this process, we ask participants to come up with a tool that can assist medical experts better in observing and choosing sperms. This could be as simple as providing an interface with different filters that allow for better control of the tracking of spermatozoa. The tool's quality is decided via a majority vote from three medical experts in the field testing the tool. The assessment will focus on the usefulness and novelty of the idea (not user interface or usability in terms of human-computer interaction).

Submission to this task should be a GitHub repository containing the code and a short *README* containing setup and usage instructions for the submitted software. The submission will be evaluated by three separate andrology experts, who will assess how useful the software would be in a real-world clinic. Participating teams will get a report containing the final evaluation and a few pointers on where the software could be improved to be more clinically viable.

4   DISCUSSION AND OUTLOOK

In this paper, we described the BioMedia 2020 challenge, which is the second part of a series of medical related challenges held at ACM Multimedia. This year, the challenge focuses on automatic semen analysis and the development of tools to aid clinicians in finding the optimal spermatozoa for fertilization. We presented four different tasks, two of which were optional, where participants were given an open dataset consisting of video recordings and related sensor data from 85 free-willing participants. Last year, five teams participated in BioMedia.

Only one team participated in the this years challenge. We believe multiple factors influenced the lack of participating teams this year. First of all, COVID-19 has and still is making a large impact on people's lives and their ability to work, which may have affected the number of people who were able to participate. Another possible reason for the lack of participation could be that assisted reproduction has certain cultural barriers that make it a difficult topic to work with [2, 5, 13, 19, 21]. This is also supported by the fact that a lot of people were interested in the task, but in the end, only one team submitted. Another limiting factor could be that this year the task was truly multimodal, and it requires a lot of effort to analyze data in a multimodal manner. BioMedia's primary purpose is to encourage multimedia researchers to explore the field of medical multimedia. In the future, we hope to continue the challenge over the next few years with different medical use-cases each year. The 2021 version will focus on mental health [12] using multimodal data from a variety of different data sources. In conclusion, we believe that the multimedia community can have a great impact on the field of medicine. We can already see several works in that direction to support our effort in encouraging even more researchers to follow that path.

REFERENCES

[1] Ying Bi, Bing Xue, and Mengjie Zhang. 2018. An Automatic Feature Extraction Approach to Image Classification Using Genetic Programming. In *Applications of Evolutionary Computation*, Kevin Sim and Paul Kaufmann (Eds.). 421–438.

[2] Henny M. W. Bos and Floor B. Van Rooij. 2007. The influence of social and cultural factors on infertility and new reproductive technologies. *Journal of Psychosomatic Obstetrics & Gynecology* 28, 2 (2007), 65–68. https://doi.org/10.1080/01674820701447439

[3] Violeta Chang, Laurent Heutte, Caroline Petitjean, Steffen Härtel, and Nancy Hitschfeld. 2017. Automatic classification of human sperm head morphology. *Computers in Biology and Medicine* 84 (2017), 205 – 216. https://doi.org/10.1016/j.compbiomed.2017.03.029

[4] Trevor G. Cooper, Elizabeth Noonan, Kirsten M. Vogelsong, Michael T. Mbizvo, Sigrid von Eckardstein, Jacques Auger, H.W. Gordon Baker, Hermann M. Behre, Trine B. Haugen, Thinus Kruger, and Christina Wang. 2009. World Health Organization reference values for human semen characteristics*‡. *Human Reproduction Update* 16, 3 (11 2009), 231–245. https://doi.org/10.1093/humupd/dmp048 arXiv:http://oup.prod.sis.lan/humupd/article-pdf/16/3/231/1791304/dmp048.pdf

[5] Daisy Deomampo. 2019. Racialized Commodities: Race and Value in Human Egg Donation. *Medical Anthropology* 38, 7 (2019), 620–633. https://doi.org/10.1080/01459740.2019.1570188

[6] Karan Dewan, Tathagato Rai Dastidar, and Maroof Ahmad. 2018. Estimation of Sperm Concentration and Total Motility From Microscopic Videos of Human Semen Samples. In *Proc. of CVPR Workshops*.

[7] Muhammad Farooq and Edward Sazonov. 2017. Feature Extraction Using Deep Learning for Food Type Recognition. In *Bioinformatics and Biomedical Engineering*, Ignacio Rojas and Francisco Ortuño (Eds.). 464–472.

[8] Pål Halvorsen, Michael Alexander Riegler, and Klaus Schoeffmann. 2019. Medical Multimedia Systems and Applications. In *Proc. of ACM MM*. 2711–2713. https://doi.org/10.1145/3343031.3351319

[9] Trine B. Haugen, Steven A. Hicks, Jorunn M. Andersen, Oliwia Witczak, Hugo L. Hammer, Rune Borgli, Pål Halvorsen, and Michael Riegler. 2019. VISEM: A Multimodal Video Dataset of Human Spermatozoa. In *Proc. of MMSYS*. 261–266. https://doi.org/10.1145/3304109.3325814

[10] Steven Hicks, Michael Riegler, Pia Smedsrud, Trine B. Haugen, Kristin Ranheim Randel, Konstantin Pogorelov, Håkon Kvale Stensland, Duc-Tien Dang-Nguyen, Mathias Lux, Andreas Petlund, Thomas de Lange, Peter Thelin Schmidt, and Pål Halvorsen. 2019. ACM Multimedia BioMedia 2019 Grand Challenge Overview. In *Proc. of ACM MM*. 2563–2567. https://doi.org/10.1145/3343031.3356058

[11] Steven A Hicks, Jorunn M Andersen, Oliwia Witczak, Vajira Thambawita, Pål Halvorsen, Hugo L Hammer, Trine B Haugen, and Michael A Riegler. 2019. Machine Learning-Based Analysis of Sperm Videos and Participant Data for Male Fertility Prediction. *Scientific Reports* 9, 1 (2019), 16770. https://doi.org/10.1038/s41598-019-53217-y

[12] Petter Jakobsen, Enrique Garcia-Ceja, Lena Antonsen Stabell, Ketil Joachim Oedegaard, Jan Oystein Berle, Steven Hicks, Vajira Thambawita, Pål Halvorsen, Ole Bernt Fasmer, and Michael Alexander Riegler. 2020. Psykose: A Motor Activity Database of Patients with Schizophrenia. https://doi.org/10.31219/osf.io/e2tzf

[13] S. Kol. 2018. Ultra-Orthodox Jews and infertility diagnosis and treatment. *Andrology* 6, 5 (2018), 662–664. https://doi.org/10.1111/andr.12533

[14] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. 2016. LIRE: Open Source Visual Information Retrieval. In *Proc. of MMSYS*. Article 30, 4 pages. https://doi.org/10.1145/2910017.2910630

[15] F Pérez-sánchez, JJ de Monserrat, and C Soler. 1994. Morphometric analysis of human sperm morphology. *International journal of andrology* 17, 5 (1994), 248–255.

[16] Michael Riegler, Mathias Lux, Carsten Griwodz, Concetto Spampinato, Thomas de Lange, Sigrun L. Eskeland, Konstantin Pogorelov, Wallapak Tavanapong, Peter T. Schmidt, Cathal Gurrin, Dag Johansen, Håvard Johansen, and Pål Halvorsen. 2016. Multimedia and Medicine: Teammates for Better Disease Detection and Survival. In *Proc. of ACM MM*. 968–977. https://doi.org/10.1145/2964284.2976760

[17] Fariba Shaker, S Amirhassan Monadjemi, and Javad Alirezaie. 2017. Classification of human sperm heads using elliptic features and LDA. In *3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*. 151–155.

[18] H. Tamura, S. Mori, and T. Yamawaki. 1978. Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics* 8, 6 (1978), 460–473. https://doi.org/10.1109/TSMC.1978.4309999

[19] Andrea Whittaker, Marcia C. Inhorn, and Francoise Shenfield. 2019. Globalised quests for assisted conception: Reproductive travel for infertility and involuntary childlessness. *Global Public Health* 14, 12 (2019), 1669–1688. https://doi.org/10.1080/17441692.2019.1627479

[20] World Health Organization, Department of Reproductive Health and Research. 2010. *WHO laboratory manual for the examination and processing of human semen.*

[21] Sophie Zadeh. 2016. Disclosure of donor conception in the era of non-anonymity: safeguarding and promoting the interests of donor-conceived individuals? *Human Reproduction* 31, 11 (10 2016), 2416–2420. https://doi.org/10.1093/humrep/dew240

A.14 Paper XIV - Explaining Deep Neural Networks for Knowledge Discovery in Electrocardiogram Analysis

Authors: Steven A. Hicks, Jonas L. Isaksen, **Vajira Thambawita**, Jonas Ghouse, Gustav Ahlberg, Allan Linneberg, Niels Grarup, Inga Strümke, Christina Ellervik, Morten Salling Olesen, Torben Hansen, Claus Graff, Niels-Henrik Holstein-Rathlou, Pål Halvorsen, Mary M. Maleckar, Michael A. Riegler, Jørgen K. Kanters

Abstract: Deep learning-based tools may annotate and interpret medical data more quickly, consistently, and accurately than medical doctors. However, as medical doctors are ultimately responsible for clinical decision-making, any deep learning-based prediction should be accompanied by an explanation that a human can understand. We present an approach called electrocardiogram gradient class activation map (ECGradCAM), which is used to generate attention maps and explain the reasoning behind deep learning-based decision-making in ECG analysis. Attention maps may be used in the clinic to aid diagnosis, discover new medical knowledge, and identify novel features and characteristics of medical tests. In this paper, we showcase how ECGradCAM attention maps can unmask how a novel deep learning model measures both amplitudes and intervals in 12-lead electrocardiograms, and we show an example of how attention maps may be used to develop novel ECG features.

Candidate contributions: Vajira contributed to the conception and design of this paper. He contributed to analyzing the results collected from the deep learning experiments discussed in this manuscript. He contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective I, Sub-objective III

scientific reports

OPEN

Explaining deep neural networks for knowledge discovery in electrocardiogram analysis

Steven A. Hicks[1,2,7✉], Jonas L. Isaksen[3,7], Vajira Thambawita[1,2], Jonas Ghouse[3], Gustav Ahlberg[3], Allan Linneberg[3], Niels Grarup[3,4], Inga Strümke[1], Christina Ellervik[3], Morten Salling Olesen[3], Torben Hansen[3,4], Claus Graff[5], Niels-Henrik Holstein-Rathlou[3], Pål Halvorsen[1,2], Mary M. Maleckar[6], Michael A. Riegler[1,7] & Jørgen K. Kanters[3,7]

Deep learning-based tools may annotate and interpret medical data more quickly, consistently, and accurately than medical doctors. However, as medical doctors are ultimately responsible for clinical decision-making, any deep learning-based prediction should be accompanied by an explanation that a human can understand. We present an approach called electrocardiogram gradient class activation map (ECGradCAM), which is used to generate attention maps and explain the reasoning behind deep learning-based decision-making in ECG analysis. Attention maps may be used in the clinic to aid diagnosis, discover new medical knowledge, and identify novel features and characteristics of medical tests. In this paper, we showcase how ECGradCAM attention maps can unmask how a novel deep learning model measures both amplitudes and intervals in 12-lead electrocardiograms, and we show an example of how attention maps may be used to develop novel ECG features.

Deep learning methods have the potential to become essential tools for diagnosis and analysis in medicine. Automatic analysis of electrocardiograms (ECGs) is a field with a long history and many different approaches[1–5], but recent years have shown that deep learning works better than traditional methods[6]. However, this family of machine learning algorithms may also bring much uncertainty and confusion among the medical practitioners they aim to help because of lacking understanding of how these algorithms work. Despite the impressive results in areas like radiology[7], dermatology[8], and cardiology[9–11], deep neural networks are often criticized for being difficult to explain and for providing little to no insight into why they produce a given result (the so-called "black-box phenomenon")[12]. Since doctors are accountable for their diagnoses, a black-box approach is unacceptable[13,14].

History has shown that doctors in practice prefer simpler, although inferior algorithms to their neural network-based counterparts, primarily because the simple algorithms are more interpretable[15]. Lack of insight has in some cases of machine learning led to obvious mistakes, which has been overlooked because the black-box decision did not allow for interpretation of the neural network predictions[16,17]. A classic example comes from deep learning in radiology (X-ray of the thorax), where the neural networks effectively distinguished between lung cancer and pneumonia simply by predicting the referring department from various labels in the image and not the relevant parts of the X-ray images. When the network is presented with X-rays without similar department labels, the network fails to distinguish between lung cancer and pneumonia[7]. This study is a good example of a mistake rooted in the differences between training and test data distribution. The neural network learned data-specific features that did not generalize to data from outside its domain. This simple but grave mistake could have easily been discovered with an explanation of the predictions where one could easily have observed what the network recognized as the most important feature for its predictions. Hence, it is clear that we need to understand the decisions of the neural network. In this respect, recent developments in explainable artificial intelligence (AI) have shown progress in shedding light on these black-boxes, which seems imperative if deep learning is to be implemented in clinics[18]. Generally, explanations are produced for image data and classification. In this work, we present a method that can obtain explanations for classification and prediction/regression tasks on non-image data. Specifically, we look at ECG where AI has become an emerging topic, where interpretable and explainable results of both classification and prediction will be crucial for clinical implementation and research.

[1]SimulaMet, 0167 Oslo, Norway. [2]Oslo Metropolitan University, 0167 Oslo, Norway. [3]University of Copenhagen, 2200 Copenhagen N, Denmark. [4]Novo Nordisk Foundation Center for Basic Metabolic Research, 2200 Copenhagen N, Denmark. [5]Aalborg University, 9220 Aalborg Ø, Denmark. [6]Simula Research Laboratory, 1364 Fornebu, Norway. [7]These authors contributed equally: Steven A. Hicks, Jonas L. Isaksen, Michael A. Riegler, and Jørgen K. Kanters. ✉email: steven@simula.no
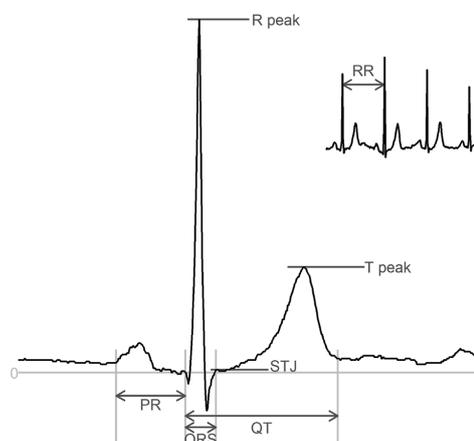
nature portfolio

1

Figure 1. An annotated ECG representative beat and rhythm strip (top-right inset) with intervals (PR, QRS, QT) and amplitudes (R peak, STJ, T peak). Amplitudes are measured with respect to the baseline. STJ denotes J-point elevation. Heart rate is calculated as HR = 60,000/RR where RR is measured in ms.

In the field of electrocardiography, a 12-lead ECG is a recording of the electrical activity of the heart using ten electrodes placed on the patient's thorax and limbs. The ECG consists of a set of voltage time-series, with several characteristic waves (see Fig. 1), which each carry clinical information about the state of the heart. The timing and the amplitude of these waves contain essential information associated with morbidity and mortality[19–22]. The ECG is one of the cheapest and most commonly used medical procedures, and the availability of large training data sets makes the ECG well-suited for neural network analysis. While automated analysis of ECGs has been a topic of research since the early 1960s[23], recently, we have seen an introduction of machine learning in ECG analysis. Deep learning has shown to be successful in using features that may indicate cardiac arrhythmias or other diseases[10].

Incorporating explainability into medical decision-making has three potential advantages. First, for implementing deep learning in the clinic, where medical decisions may be a matter of life and death, a deep learning algorithm that explains how it arrived at a particular decision allows the prevention of rare but potentially fatal mistakes. Such mistakes may be the result of shortcomings in the training of the algorithm (such as biased or narrow training data) or noisy or faulty input data leading to unexpected and extreme decisions. Explainability thus provides a higher level of trust and transparency in the clinical setting because a doctor can understand what the algorithm bases its predictions on[24,25]. This may pave the way for the implementation of neural networks in clinical practice and reduce human error, resulting in fewer fatalities. Second, more explainable models may allow for the identification of novel features that may lead to a new understanding of the disease pathophysiology and increased diagnostic capability, which in the end may save lives. Suppose a deep learning algorithm successfully predicts sudden cardiac death using ECGs from a given population. If the algorithm can explain where the information is located in the ECG, we may combine medical knowledge of the ECG with that location making it possible to identify novel mechanisms of sudden cardiac death. This would potentially make it possible to identify an intervention or possible drug target to prevent untimely death. Third, making the algorithms more interpretable may be important from a legal perspective because one would be able to explain why a model made an incorrect decision and place responsibility accordingly.

The work presented in this paper has three primary contributions. First, we present the architecture of a residual convoluted neural network (CNN) and show that it is able to quantify intervals and amplitudes in the ECG more accurately than trained cardiologists are. Second, we present a modified version of the GradCAM[26] algorithm called electrocardiogram gradient class activation map (ECGradCAM) and show how the resulting attention maps can be utilized for ECG analysis to understand, interpret, and learn from neural network predictions. Third, we show how network and attention maps may be combined to identify novel features in the ECG by identifying a novel feature to determine the sex of a person based on an ECG.

Results

Automatic ECG analysis and data description. We define two case studies for model evaluation: a regression study, measuring standard, clinically relevant intervals and amplitudes from the ECG, and a classification study to predict the sex from the ECG. Numerous cardiovascular diseases are diagnosed by measuring key intervals and amplitudes present in the ECG[27,28], and we leverage this to predict these intervals directly instead of categorizing ECGs into normal and abnormal groups. The predicted intervals and amplitudes include the PR interval, QRS duration, heart rate, J-point elevation, QT interval, R-wave amplitude, and T-wave amplitude (see

2

265

Appendix A. Published Articles

Variable	GESUS (training)	Inter99 (replication)
Number of samples (n)	8939	6667
Age (years)	56.6 [35.1; 78.4]	45.3 [34.5; 60.0]
Female sex	54.3% (4852)	51.2% (3412)
BMI (kg/m^2)	26.1 [20.4; 35.3]	25.7 [20.2; 34.8]
Heart rate (bpm)	64 [48; 85]	66 [51; 86]
QT interval (ms)	408 [364; 460]	402 [362; 450]
PR interval (ms)	158 [126; 204]	156 [124; 196]
QRS duration (ms)	92 [76; 118]	90 [76; 110]
J-point elevation V5 (μV)	− 5 [− 54; 48]	4 [− 35; 58]
R-peak amplitude V5 (μV)	1376 [698; 2,426]	1171 [600; 2,044]
T-wave amplitude V5 (μV)	346 [122; 698]	327 [122; 649]
Bundle branch block (QRS ≥ 120 ms)	4.4% (390)	1.6% (107)
1° AV block (PR > 220 ms)	2.0% (174)	0.9% (61)

Table 1. Characteristics of the participants in both population studies. Values are presented as median [fifth to ninety-fifth percentiles] for continuous measures and % (n) for categorical variables.

Type	Variables	Validation on GESUS (fivefold)		Replication on Inter99		Zero R on Inter99	
		MAE	RMSE	MAE	RMSE	MAE	RMSE
Median	QT Interval (ms)	3.26 ± 0.80	5.08 ± 0.40	3.13 ± 0.19	4.89 ± 0.19	21.7	27.4
	PR Interval (ms)	2.82 ± 0.15	4.52 ± 0.49	2.73 ± 0.06	4.70 ± 0.23	17.6	22.8
	QRS duration (ms)	2.98 ± 0.15	4.10 ± 0.22	2.58 ± 0.07	3.43 ± 0.07	9.0	11.6
	Heart Rate (beats per min)	1.54 ± 0.07	2.44 ± 0.09	1.57 ± 0.06	2.33 ± 0.07	8.6	11.1
	J-point elevation (μV)	8.16 ± 0.40	11.20 ± 0.69	5.77 ± 0.10	8.09 ± 0.12	22.2	29.0
	T-wave amplitude (μV)	5.63 ± 1.31	15.2 ± 6.83	5.80 ± 1.13	16.10 ± 0.53	129.0	167.0
	R-peak amplitude (μV)	8.60 ± 1.05	16.00 ± 4.13	8.30 ± 0.98	21.70 ± 0.71	413.0	501.0
Rhythm	QT Interval (ms)	3.97 ± 0.03	6.05 ± 0.39	3.62 ± 0.03	5.82 ± 0.20	21.7	27.4
	PR Interval (ms)	3.67 ± 0.21	5.60 ± 0.60	3.58 ± 0.60	5.80 ± 0.31	17.6	22.8
	QRS duration (ms)	3.08 ± 0.12	4.33 ± 0.17	3.39 ± 0.06	4.49 ± 0.07	9.0	11.6
	Heart Rate (beats per min)	0.31 ± 0.01	0.40 ± 0.02	0.18 ± 0.01	0.6 ± 0.1	8.6	11.1
	J-point elevation (μV)	10.50 ± 0.31	14.10 ± 0.50	7.90 ± 0.19	10.70 ± 0.24	22.2	29.0
	T-wave amplitude (μV)	11.50 ± 0.43	25.00 ± 5.55	9.40 ± 0.41	19.40 ± 1.79	129.0	167.0
	R-peak amplitude (μV)	20.10 ± 0.70	33.00 ± 4.54	17.4 ± 0.55	51.00 ± 13.21	413.0	501.0

Table 2. Training and validation error in GESUS[25] and replication error in Inter99[2].

Fig. 1). By predicting these measurements directly, we allow for better interpretation of the results rather than limiting it to a predefined set of categories. The second case study looks at differentiating between male and female ECGs. This use case is motivated by the difficulty for humans to determine sex based on ECGs alone, making it a good candidate for visualization as one may find certain features that correlate to sex that have previously gone unfound.

All models are trained and evaluated on either raw 10-s 12-lead ECGs or on the 12SL-generated median beat from the GESUS dataset[29]. The performance of all GESUS generated models is replicated in the Inter99 dataset[30]. The demographics of the study populations are summarized in Table 1. To evaluate the effect of ECG abnormalities on the prediction performance of our network, we tested the prediction errors on subgroups with bundle branch blocks (QRS > 120 ms) or first-degree AV block (PR > 220 ms). Furthermore, to study the performance on ECG with strange, abnormal T waves, we used the existing T-wave Morphology Combination Score[31] to divide the ECGs into four different groups ranging from peaked to flattened T-waves. Supplementary Table S1 shows that the network performed only slightly worse in subjects with bundle branch block, AV block, and flattened T waves, respectively.

CNN results. The performance of our method for predicting ECG intervals and amplitudes is evaluated using quantitative regression metrics, as seen in Table 2. The primary metrics used for evaluation are the mean absolute error (MAE) as it is easily interpretable, and the root-mean-squared error (RMSE) as it is more sensitive to outliers. In Table 2, we see that every model beats the ZeroR-estimate (predicting the mean) by a large margin. This shows that the proposed architecture successfully analyzes the ECG, both in the voltage and time domains. For interval measurements, the MAEs are close to two samples (4 ms) for both the median beat and rhythm strip
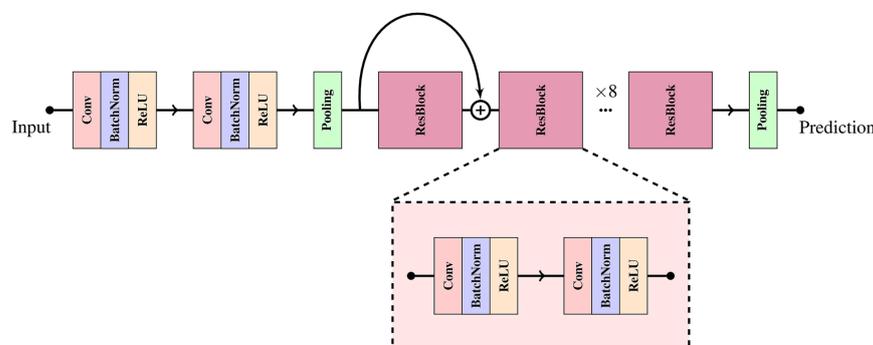
Figure 2. The convolutional neural network-based architecture used for all experiments. Each red block corresponds to a convolutional layer, the blue blocks represent a batch normalization layer, the yellow blocks are ReLU activations, and the green block represents average pooling. Besides these base building blocks, the model consists of eight residual modules (purple blocks) composed of two sequential blocks of convolution, batch normalization, and ReLU activation.

(10 s) measurements. Amplitude measurements varied similarly (the least significant bit is 4.88 μV), indicating that the network performed equally well with voltage and time-domain measurements.

Attention maps. To create meaningful and detailed visualizations, we modified the GradCAM approach so that visualizations are generated for each lead of the ECG, where the final attention maps are produced by averaging the importance values across all leads. We call this method ECGGradCAM since it can give a more accurate representation of what regions of the ECG are most important for the model. We focus our interpretation on the last layer of the last residual module of the neural network (as shown in Fig. 2). This corresponds to the final layer before prediction, meaning the visualizations show what areas of the ECG are deemed most important at the moment of prediction. It can also be useful to interpret the intermediate layers of a network[32] as these layers may offer insight into how the network's perception changes and how it narrows down the analysis to the final result (see Supplementary Figure S1). Here, we note that the initial residual block recognizes several features in the ECG, which becomes more and more focused on the relevant wave as the ECG progresses through the residual layers.

The attention maps often highlight the areas we expect when predicting a specific interval or amplitude. Figure 3 presents a median ECG visualized for six of the predicted variables. For instance, the QRS complex is highlighted when we predict QRS duration, and the end of the T-wave is delineated along with the beginning of the QRS complex for QT interval measurement. For amplitude measurements, the corresponding wave top is correctly pinpointed by the attention maps. One should note that for amplitude measurements, other parts of the ECG are given minor importance, likely for the network to learn about the ECG voltage baseline. For intervals, secondary activations are also observed, such as the T-wave being highlighted when measuring the PR-interval. We hypothesize that these secondary activations may be happening because the network needs to appreciate the whole ECG in order to narrow its search down and perform the actual measurements. This is further supported by the PR interval attention maps generated for the intermediate layers (see Supplementary Figure S1), where the network highlights the QT in the former layers but less so at the moment of prediction.

Sex prediction. For a cardiologist, the task of determining a subject's sex from the ECG is nearly impossible. However, our network is able to correctly identify the sex with an accuracy of 89% (Table 3). Here, we can see the potential of attention maps, as the accuracy output from the network does not give any clue or insight into how the network made its decision on sex classification. The attention maps (see Fig. 4) clearly show that the ECG sex classification is mainly based on the QRS complex and more specifically on the downslope of the R-wave, offering new insight into electrophysiology. Using findings from the attention maps, we did a post-hoc analysis with logistic regression predicting sex using QRS duration, R- and S-amplitudes, and the timing of the R- and S-waves. It revealed an accuracy of 73% (our CNN: 89% QRS duration alone: 69%) and an AUC of 0.80 (our CNN 0.96 QRS duration alone: 0.72). The wave blocking experiments (Table 3) verified this observation, since removing the P-wave has almost no influence on the accuracy of the sex prediction, removing the T-wave had only minor influence, whereas removing the QRS complex resulted in a drastic reduction in performance. This shows that one can obtain new knowledge by using our ECGGradCam method combined with the deep neural network.

Human cardiologist vs neural network evaluation. To assess how the neural network compares to standard clinical decision making, we further evaluate the performance of our model by comparing its predictions to predictions made by cardiologists who have manually annotated a set of twenty randomly selected ECGs
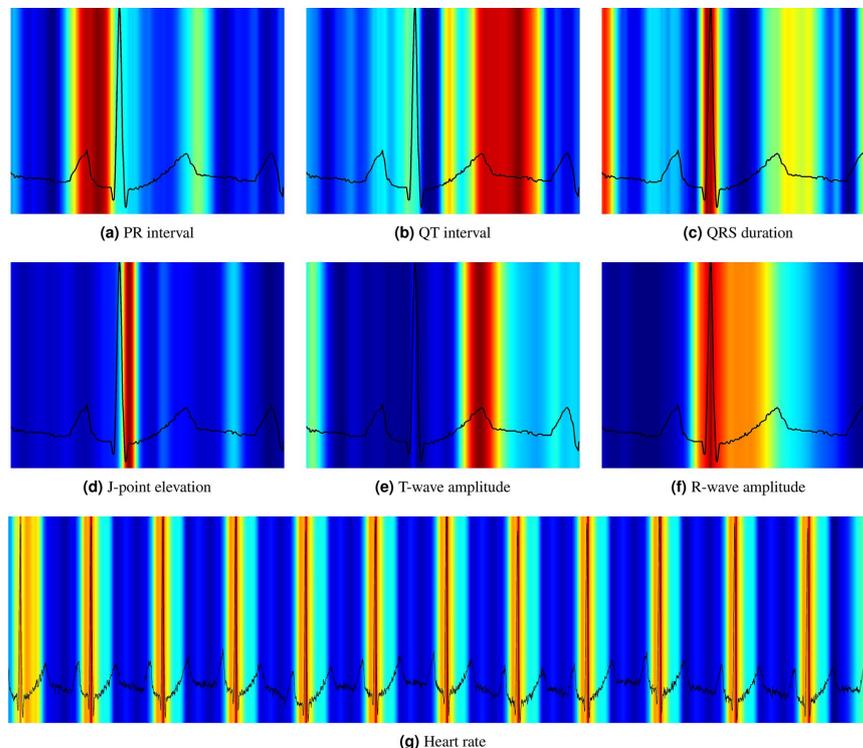
Appendix A. Published Articles

(a) PR interval   **(b)** QT interval   **(c)** QRS duration

(d) J-point elevation   **(e)** T-wave amplitude   **(f)** R-wave amplitude

(g) Heart rate

Figure 3. Visualization of the attention maps generated for the interval and amplitude prediction models. As we can see from the plots, the model learns to inspect the waves and intervals that are related to the predicted variable. Red color indicates high importance and blue color low importance of the ECG for the decision of the neural network.

Variable	Metric	Median ECG	Blanking P-wave	Blanking QRS complex	Blanking T-wave
QT interval (ms)	MAE ↓	3.13 ± 0.21	3.20 ± 0.22	31.76 ± 6.34	47.35 ± 12.38
PR interval (ms)	MAE ↓	2.73 ± 0.06	32.95 ± 10.31	40.55 ± 5.88	3.84 ± 0.71
QRS duration (ms)	MAE ↓	2.58 ± 0.08	3.99 ± 0.48	40.55 ± 5.88	3.50 ± 0.10
Heart Rate (bpm)	MAE ↓	1.57 ± 0.07	2.92 ± 0.19	3.62 ± 1.89	4.79 ± 1.10
J-point elevation (μV)	MAE ↓	5.77 ± 0.18	6.43 ± 0.56	23.07 ± 3.29	8.62 ± 0.42
T-wave amplitude (μV)	MAE ↓	5.80 ± 1.29	6.13 ± 1.45	8.70 ± 1.49	339.04 ± 8.0
R-wave amplitude (μV)	MAE ↓	8.35 ± 1.12	8.64 ± 1.11	927.38 ± 16.0	10.48 ± 3.56
Sex classification (%)	ACC ↑	88.80 ± 0.7	87.50 ± 1.0	62.40 ± 6.6	79.80 ± 2.5

Table 3. Mean absolute error (MAE) and accuracy (ACC) ± standard deviation measured on the replication dataset when blanking specific waves of a median heartbeat. Prediction errors increased dramatically when the feature in question is blanked out. Prediction errors also often increased slightly when other parts of the ECG are blanked. In the metric column, the arrow signifies whether higher or lower values are better, i.e., an arrow pointing downwards means that lower values are preferable.

from the Inter99 replication dataset. As seen in Table 4, the trained networks prove substantially more precise and consistent than human expert assessments. Human bias-corrected MAE and RMSE are around 15–20 ms, i.e., a factor 4 to 5× higher than the neural network. Errors in heart rate measurements are below one beat per minute (BPM) for the network but about 3 BPM for the human operators with multiple errors above 10 BPM. Amplitude measurements are much more difficult for humans, given the resolution of the digital ECG and the
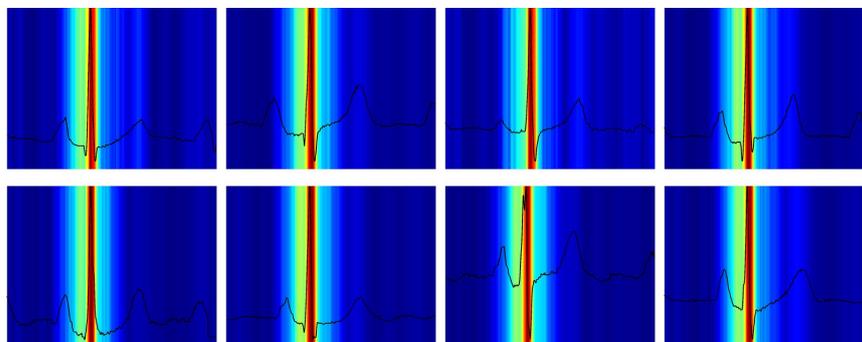
Figure 4. Visualizations of the attention maps from the sex classification model for eight different ECGs. The plots suggest that the QRS-complex and especially the downslope of the R-wave is most important when distinguishing between a male and female ECG. Red color indicates high importance and blue color low importance of the ECG for the decision of the neural network.

Variables	Test set, mean ± SD	Neural network			Doctor A			Doctor B		
		Bias	MAE*	R	Bias	MAE*	R	Bias	MAE*	R
Heart rate (bpm)	71 ± 8	0.04	0.20	0.99	0.98	3.20	0.68	− 0.52	1,38	0.99
QT interval (ms)	392 ± 21	0.50	3.70	0.98	− 8.50	15.8	0.49	− 17.8	9.2	0.89
QRS duration (ms)	91 ± 10	− 3.30	3.00	0.98	− 7.80	11.9	0.39	− 7.6	8.22	0.55
PR interval (ms)	161 ± 16	− 2.50	4.70	0.99	5.90	8.00	0.87	6.45	9.01	0.82
R-peak amplitude (mm)	12 ± 4	0.02	0.16	0.99	0.57	0.42	0.98	0.06	0.50	0.95
J-point elevation (mm)	− 0.04 ± 0.28	0.02	0.09	0.98	0.12	0.19	− 0.01	− 0.01	0.17	0.86
T-wave amplitude (mm)	3.00 ± 1.50	0.03	0.10	0.99	− 0.13	0.32	0.89	0.20	0.56	0.80

Table 4. Evaluation of twenty randomly selected ECGs by two experienced cardiologists. Bias is the average difference between the ground truth and the doctor/network. Mean absolute error (MAE*) is the bias-subtracted mean absolute error, to account for the fact that there is no universal correct measurement for an ECG, ensuring that any personal bias does not contribute to the error (e.g., some doctors measure QT consistently shorter or longer than others). R: Pearson's correlation coefficient between the doctor's two measurements/the neural networks five folds. *SD* Standard deviation.

accuracy of the human eye. For the R-peak amplitude, the network operates at an MAE of twice the least significant bit at 4.8 µV. In contrast, the human reviewer operates with an MAE in the magnitude of half a millimeter (corresponding to 50 µV or 10 the times least significant bit).

Discussion

This paper identified three novel findings. First, we presented a residual CNN that reliably analyzes both ECG intervals (time dimension) and amplitudes (voltage dimension) independent of whether the ECG presented is a 10-s 12-lead ECG or a 1.2-s median representative beat. The architecture proved accurate for a variety of different ECG tasks. In all cases, CNN prediction outperformed the cardiologists by a large margin. Furthermore, with repeated blinded measurements, the cardiologists had a large intra-observer variation, whereas the neural network was very consistent in its predictions between folds. The MAE was between 3 and 4 ms corresponding to two samples which is close to the physical obtainable lower limit (since the interval uncertainty consists of two cumulative uncertainties of 2 ms at each end of the interval). In general, measurements were more accurate when using 1.2-s median beats compared to 10-s rhythm strips, except for heart rate. The generation of the median beat reduces noise by averaging all beats during the 10-s ECG, stretching each complex to minimize the influence of variations, making measurements more accurate. The exception of heart rate is not surprising since several ECG complexes are needed to properly estimate heart rate, and these are only found in the rhythm strip. More surprising is the finding that the neural network uses other features than the RR-interval to calculate heart rate and obtains a relatively good estimate from the single ECG complex of the median beat. In fact, the heart rate estimate from the median was only slightly worse than that based on the rhythm strip (see Table 2). Similar to human overreaders, the network performed slightly worse in subjects with conduction blocks. However, the increase in prediction error was small compared to the human overreader error, showing a satisfactory accuracy for clinical use. The QT interval is an important feature in the ECG because the interval is related to

269

Appendix A. Published Articles

sudden death[33]. It is well known that abnormal T waves are difficult to measure correctly, which is important because these abnormal T waves are associated with increased mortality[34]. As seen in the supplementary (Supplementary Table S1), the network accurately measured even the most flattened T waves only marginally worse than the "easy-to-measure" peaked T waves with a sharp, well-defined end.

Second, we presented ECGradCAM attention maps for 12-lead ECG-analysis to explain how the network made its decisions. In medical practice, explainability is crucial because medical doctors are concerned that algorithms may produce erroneous results, either due to bias or trying to predict outcomes not appropriately represented in the training data. For example, measurement of the PR interval in a case of ventricular tachycardia would be unreliable if the training set only consisted of ECGs taken in sinus rhythm. Interpretation and transparency should be at the forefront when developing new algorithms intended for medical use. Although the results suggest that deep learning could be an essential tool for cardiologists for doing analysis and interpretation of the ECG, it is doubtful that the neural network models without explanations of an estimate of uncertainty would be accepted by doctors.

The attention maps showed nicely that all amplitude measurements focused on the proper ECG wave, and in cases of interval measurements, both the beginning and the end of the specific interval are most often highlighted by the algorithm. The obscurity tests confirmed the attention map results. When we removed ECG waves used for the specific interval or amplitude measurement, the MAE increased dramatically, confirming the message from the attention maps that our CNN focused on the same features as human cardiologists, just more accurately and reliably. One may also notice that the network tries to extract information about the baseline from the ECG. Since we use batch normalization (a standard feature in neural networks to avoid exploding gradients), the network had to get an idea of the magnitude of normalization to restore the absolute values needed for amplitude prediction. This may be why the network also focuses on more steady, constant parts of the ECG. By providing these explanations with a predicted variable, we allow the users to interpret the results with confidence that model had some notion of the traits that make up the variable in question.

Third, sex prediction is an excellent example of how neural networks combined with an explanation method can be used to discover novel medical knowledge. It is well known that there are sex differences in the ECG. Female ECGs, on average, have a longer QT interval, faster heart rate, and shorter QRS duration[35]. However, if one were to ask a cardiologist to determine the sex based on an ECG alone, they would not be able to make a confident prediction. Recent studies using deep learning have shown that neural networks can differentiate between the sexes from the ECG alone[36], but the underlying reasoning is not provided. The attention maps indicate that the physiological background seems to be differences in the R-wave downslope, which may provide important mechanistic insight into the observed sex differences. We confirmed the findings by our neural network that simple logistic regression with QRS duration, wave amplitudes, and timing (slopes can be inferred by wave amplitudes and timings) significantly improved sex prediction compared to QRS duration alone. Although adding the R-wave downslope to the QRS duration significantly increased sex prediction, the neural network still performed better than the logistic regression, most likely because the R-wave downslope is not the sole source of information. However, it is also possible that the R-wave downslope contains nonlinear information or that only part of the downslope is relevant. This study constitutes a scholarly example that the use of attention maps can assist scientists in discovering novel insights and identify hitherto unknown features for classification, which may lead to important physiological understanding. Classifying an appropriate outcome in a suitable population, one may identify novel prognostic markers in the ECG for that outcome, which may lead to a suggestion for possible treatments.

Conclusions

This paper presented a study on interpreting deep learning models used for ECG analysis. We propose a neural network architecture that predicts multiple attributes of a standard median or 10-s rhythm ECG with low error. The model was compared against real-world cardiologists, and our model outperformed the cardiologists by a large margin. The predictions were interpreted using attention maps (ECGradCAM), which show how the network operates and confirmed that the neural network analyzes ECGs similarly to trained cardiologists. Furthermore, we show that the neural network can differentiate between male and female ECGs with over 90% accuracy. Using the ECGradCAM attention maps, we find that the down-slope of the R wave is a crucial feature of an ECG when determining sex. This emphasizes the need for more interpretable machine learning methods as they can be used to find new insights in rather mature medical fields such as ECG analysis. We believe that open and transparent systems are paramount for their adoption and use in medicine. Making high-risk decisions based on the output of a black-box algorithm is irresponsible and could potentially have fatal consequences that could easily be avoided. We hope this paper motivates a more thorough evaluation and interpretation of deep learning-based models applied to all of medicine and not only to ECGs.

Methods

Data populations. We use digital ECGs from two population studies. (1) The Danish General Suburban Population Study (GESUS)[29] consisting of 8939 free-living subjects (age 56.5 ± 13.5, 54% females) at least 18 years old from the Naestved municipality, 90 km south of Copenhagen, the Capital of Denmark, randomly chosen. The study was approved by the local ethics committee (SJ-113, SJ-114, SJ-147, SJ-278). (2) The Inter99 study (CT00289237, ClinicalTrials.gov) consists of 6,667 free-living subjects (age 46.1 ± 7.9, 51% females) randomly drawn from the Glostrup municipality with an age of 30–65 years[30]. This yields a collection of ECGs from people with and without cardiac disease and an equal representation of men and women. Both studies are conducted in accordance with the Declaration of Helsinki.

Electrocardiography. All ECGs are digitally recorded as 10-s ECGs with 12 leads. All ECGs are transferred to a MUSE Cardiology Information system (GE Healthcare, Wauwatosa, WI, USA) and ground truths are calculated with version 21 of the Marquette 12 SL algorithm (GE Healthcare, Wauwatosa, WI, USA). The ECGs are recorded with a sample rate of 500 Hz and a resolution of 4.88 μV per least significant bit.

Prediction model. *Architecture.* A digital electronic ECG can be represented as a two-dimensional matrix of integers representing the voltage at a specific point in time. To analyze these measurements, we use a standard convolutional neural network (CNN) consisting of eight residual modules (as introduced by He et al.[37]) to capture the complex features and relationships present in a standard ECG. The neural network architecture consists of 1,652,993 parameters and is built to handle two different types of input, a single representative median heartbeat of 1.2-s duration and a 10-s rhythm ECG. Both input types contain data from 12-lead ECGs. A detailed view of the neural network architecture can be seen in Fig. 2. From the input layer, the ECG is passed through two convolutional layers before being average pooled. The two convolutional layers generate 64 and 32 feature maps with a kernel size of 8 and 3. After this initial convolution block, the output is sent through eight residual blocks, each consisting of two convolutions. Each convolutional layer in the residual blocks generates 64 and 32 feature maps, respectively, and both layers use a kernel size of 50. We use a large kernel size to extend the receptive field to include multiple parts of a typical ECG. This could, for example, capture both the P wave and the QRS complex in a single convolution. We add batch normalization after each convolution and dropout after the final convolution with a drop rate of 50%. After the eight residual blocks, the output is globally average pooled before making the final prediction. The prediction layer consists of a single neuron with a linear activation that predicts a single variable of the ECG.

Training. All models are trained for a maximum of 1000 epochs on a computer consisting of two Intel Xeon Silver 4116 CPUs running at 2.1 GHz, four Nvidia RTX 2080Ti graphics cards, and 96 gigabytes of RAM. The models are implemented using Keras version 2.1.0 with a TensorFlow backend on Ubuntu 18.04.2. Processing one ECG takes about 0.06 s using the aforementioned hardware and Python libraries. To optimize the weights, we used the gradient descent-based optimizer Nadam[38] with a learning rate of 0.0005. The learning rate is selected based on manual testing and prior experience from our previous works[32]. Otherwise, we used the Keras defaults for all optimizer parameters. In total, we performed 14 different experiments, seven using the median for prediction and seven using the rhythm. The variables predicted with regression parameters (include the QT interval, PR interval, QRS duration, heart rate, ST-segment deviation from baseline at the junction (STJ), T-wave amplitude, and R-peak amplitude). The three amplitudes are lead specific and lead V5 is used. Some of these variables cater more to rhythm analyses (such as heart rate), while others are more appropriate for median complexes (such as the R-peak). One problem with training on the median complexes is that they are all centered in a manufacturer-specific way, whereby each wave appears in nearly the same place in each of the ECG. Thereby, the network can learn to predict a particular vicinity and guarantee a relatively low error. To circumvent this problem, we time-shifted all median complexes by a random amount (− 40 to +40 ms) so that the network learns to find the individual waves. This increases the likelihood that the model can be used on ECGs from other manufactures with different temporal alignment. No alignment is performed for the rhythm ECGs; the start of the recording is random with respect to the ECG. Furthermore, to test the network's ability to classify in binary outcomes, we classified the ECG for sex (male/female).

Attention maps. To obtain physiological insights from the neural network's decisions, it is necessary to understand how and why a decision is achieved. In this study, we used attention maps to visualize which parts of the ECG have importance for each interval/amplitude prediction. To explain the predictions of our model, we use gradient-based activation maps (attention maps) to visualize which parts of the ECG are the most important when predicting a given variable or class. The technique used is a modified version of GradCAM[26], commonly used to interpret image classification models. As we show in our study, this approach works just as well for regression tasks of quantitative measurements in the data. Visualizations are generated based on a given network layer and output neuron, which produces a heat map that marks the most important areas as hot (red color) and less important regions as cold (blue color). In this context, importance signifies how much weight a specific area contributes to the overall prediction. We are not the first to use attention maps to interpret deep neural networks applied to ECGs[39,40]. Most other works use these visualizations to confirm that their model does not deviate from the expectations of the medical doctors[41]. Our work goes one step further and expands the method of explanations to find new insights into the unique properties of ECGs through a case study on sex classification. Furthermore, even though the attention maps are generated on a per-lead basis, we average the explanations for each lead to produce a visualization that contains more fine-grained details and thus is able to more accurately represent what regions of the ECG are most important for the model when making a specific prediction.

Evaluation. To ensure a fair and robust evaluation, we trained each model with five-fold cross-validation for 1000 epochs on the GESUS dataset[29], resulting in 7152 samples being used for training and 1787 for validation. After training and internal cross-validation, the results of GESUS models are replicated in the Inter99 dataset[30] to examine whether the models are generalizable or not. As seen in Table 1, GESUS and Inter99 datasets are comparable regarding ECG measurements, although participants in the GESUS study are on average older than participants in the Inter99 study. The neural network performance is evaluated by the MAEs (|predicted – actual value|) and the RMSEs ($\sqrt{\frac{1}{n} \sum \left(|predicted - actual\ value| \right)^2}$) to evaluate the mistakes of the neural networks

Appendix A. Published Articles

relative to the ground truth. To give an idea of the magnitude of uncertainty, we calculated the ZeroR-estimate, defined as constantly guessing the population mean of the desired variable. If a model's performance is not better than ZeroR, the model has not learned anything except the population mean. Conversely, if the model performance is better than the ZeroR, it follows that the model has succeeded in extracting and processing features from the ECGs.

Furthermore, ECGs were evaluated manually by two skilled cardiologists. Whereas the neural network by definition has a bias (i.e., average error) of zero (ignoring an eventual bias in the ground truth from the 12SL algorithm), the human overreaders may exhibit substantial bias relative to the ground truth (i.e., the measure consistently shorter or longer intervals), which originates from their own training and personal preference. Since this bias is not an error, the human bias is subtracted from the errors before calculating human MAE and RMSE.

Wave blocking. To verify that the neural network model is focusing on relevant waves of the ECG, and as an alternative to the attention maps, we remove specific parts of the ECG (either the P- QRS- or T-wave) from the median ECGs of the replication set. Using the MUSE 12SL fiducial points, we blank out a wave by replacing it from the start to the end with a lead-specific linear interpolation. This analysis represents an alternative measure of explainability for representative beats of an ECG by analyzing the decrease in performance when different waves of the ECG are blanked. Thus, we can test how dependent our model is on different parts of the ECG and verify which waves the model is focusing on when making a prediction. The results in Table 3 show that the model performance drop when the wave involved with a particular feature is removed. However, we also find that removing non-involved waves typically decreases performance slightly, suggesting that the neural network also includes other parts of the ECG to stabilize the model to ensure that it is analyzing the correct part of the ECG.

Ethics. We confirm that all experiments were performed in accordance with Helsinki guidelines and regulations of the Danish Regional Committees for Medical and Health Research Ethics. The data studies were approved by the ethical committee of Region Zealand (SJ-113, SJ-114, SJ-191), ethical committee of Copenhagen Amt (KA 98 155). Written informed consent was obtained from all study participants.

Data availability
The data is not available to the public.

Code availability
The code used to conduct the experiments and generated the related attention maps is available on GitHub at https://github.com/stevenah/ecg-attention-maps.

References
1. Gupta, V. & Mittal, M. Arrhythmia detection in ECG signal using fractional wavelet transform with principal component analysis. *J. Inst. Eng. India Ser. B* **101**, 451–461 (2020).
2. Gupta, V., Mittal, M. & Mittal, V. An efficient low computational cost method of R-peak detection. *Wirel. Pers. Commun.* **118**, 359–381 (2021).
3. Josko, A. Discrete Wavelet Transform In Automatic ECG Signal Analysis. in *2007 IEEE Instrumentation Measurement Technology Conference IMTC 2007* 1–3 (2007). doi:https://doi.org/10.1109/IMTC.2007.379244.
4. Silipo, R. & Marchesi, C. Artificial neural networks for automatic ECG analysis. *IEEE Trans. Signal Process.* **46**, 1417–1425 (1998).
5. Schreier, G., Kastner, P. & Marko, W. An automatic ECG processing algorithm to identify patients prone to paroxysmal atrial fibrillation. in *Computers in Cardiology 2001. Vol.28 (Cat. No.01CH37287)* 133–135 (2001). doi:https://doi.org/10.1109/CIC.2001.977609.
6. N, S., P, W., T, S. & W, S. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. *IEEE J. Biomed. Health Inform.* **PP**, (2020).
7. Zech, J. R. *et al.* Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Med* **15**, e1002683 (2018).
8. Esteva, A. *et al.* Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115–118 (2017).
9. Acharya, U. R. *et al.* Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network. *Inf. Sci.* **405**, 81–90 (2017).
10. Zihlmann, M., Perekrestenko, D. & Tschannen, M. Convolutional recurrent neural networks for electrocardiogram classification. **1710.06122v2**, (2019).
11. Hannun, A. Y. *et al.* Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nat. Med.* **25**, 65–69 (2019).
12. Carvalho, D. V., Pereira, E. M. & Cardoso, J. S. Machine learning interpretability: A survey on methods and metrics. *Electronics* **8**, 832 (2019).
13. Caruana, R. *et al.* Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1721–1730 (Association for Computing Machinery, 2015). doi:https://doi.org/10.1145/2783258.2788613.
14. Cooper, G. F. *et al.* An evaluation of machine-learning methods for predicting pneumonia mortality. *Artif. Intell. Med.* **9**, 107–138 (1997).
15. Riegler, M. *et al.* Multimedia and Medicine: Teammates for Better Disease Detection and Survival. in *Proceedings of the 24th ACM international conference on Multimedia* 968–977 (Association for Computing Machinery, 2016). doi:https://doi.org/10.1145/2964284.2976760.
16. Badgeley, M. A. *et al.* Deep learning predicts hip fracture using confounding patient and healthcare variables. *NPJ Digit. Med.* **2**, 31 (2019).

9

272

17. Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G. & King, D. Key challenges for delivering clinical impact with artificial intelligence. *BMC Med.* **17**, 195 (2019).
18. Chen, D. *et al.* Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ Digit. Med.* **2**, 43 (2019).
19. Cullen, K., Stenhouse, N. S., Wearne, K. L. & Cumpston, G. N. Electrocardiograms and 13 year cardiovascular mortality in Busselton study. *Br. Heart J.* **47**, 209–212 (1982).
20. Goldberg, R. J. *et al.* Duration of the QT interval and total and cardiovascular mortality in healthy persons (The Framingham Heart Study experience). *Am. J. Cardiol.* **67**, 55–58 (1991).
21. Nielsen, J. B. *et al.* Risk prediction of cardiovascular death based on the QTc interval: evaluating age and gender differences in a large primary care population. *Eur. Heart J.* **35**, 1335–1344 (2014).
22. Nielsen, J. B. *et al.* J-shaped association between QTc interval duration and the risk of atrial fibrillation: results from the Copenhagen ECG study. *J. Am. Coll. Cardiol.* **61**, 2557–2564 (2013).
23. Stallmann, F. W. & Pipberger, H. V. Automatic recognition of electrocardiographic waves by digital computer. *Circ. Res.* **9**, 1138–1143 (1961).
24. Bussone, A., Stumpf, S. & O'Sullivan, D. The Role of Explanations on Trust and Reliance in Clinical Decision Support Systems. in *2015 International Conference on Healthcare Informatics* 160–169 (IEEE, 2015). doi:https://doi.org/10.1109/ICHI.2015.26.
25. Cabitza, F., Rasoini, R. & Gensini, G. F. Unintended consequences of machine learning in medicine. *JAMA* **318**, 517–518 (2017).
26. Selvaraju, R. R. *et al.* Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **128**, 336–359 (2020).
27. Macfarlane, P. W. *et al. Comprehensive Electrocardiology*. (Springer-Verlag, 2010).
28. GE Healthcare. MarquetteTM 12SLTM ECG analysis program physician's guide 2056246-002 Revision C. (2015).
29. Juhl, C. R., Miller, I. M., Jemec, G. B., Kanters, J. K. & Ellervik, C. Hidradenitis suppurativa and electrocardiographic changes: A cross-sectional population study. *Br. J. Dermatol.* **178**, 222–228 (2018).
30. Ghouse, J. *et al.* Rare genetic variants previously associated with congenital forms of long QT syndrome have little or no effect on the QT interval. *Eur. Heart J.* **36**, 2523–2529 (2015).
31. Graff, C. *et al.* Quantitative analysis of T-wave morphology increases confidence in drug-induced cardiac repolarization abnormalities: Evidence from the investigational IKr inhibitor Lu 35–138. *J. Clin. Pharmacol.* **49**, 1331–1342 (2009).
32. Hicks, S. *et al.* Dissecting Deep Neural Networks for Better Medical Image Classification and Classification Understanding. in *2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS)* 363–368 (2018). doi:https://doi.org/10.1109/CBMS.2018.00070.
33. Haarmark, C. *et al.* Reference values of electrocardiogram repolarization variables in a healthy population. *J. Electrocardiol.* **43**, 31–39 (2010).
34. Isaksen, J. L. *et al.* Electrocardiographic T-wave morphology and risk of mortality. *Int. J. Cardiol.* **328**, 199–205 (2021).
35. Sachin Khane, R. & Surdi, A. D. Gender differences in the prevalence of electrocardiogram abnormalities in the elderly: a population survey in India. *Iran. J. Med. Sci.* **37**, 92–99 (2012).
36. Zachi, I. *et al.* Age and sex estimation using artificial intelligence from standard 12-lead ECGs. *Circ. Arrhythm. Electrophysiol.* **12**(9). https://doi.org/10.1161/CIRCEP.119.007284 (2019).
37. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 770–778 (IEEE, 2016). doi:https://doi.org/10.1109/CVPR.2016.90.
38. Dozat, T. Incorporating Nesterov Momentum into ADAM. in *ICLR* 4 (2016).
39. van de Leur Rutger R. *et al.* Automatic Triage of 12-Lead ECGs Using Deep Convolutional Neural Networks. *J. Am. Heart Assoc.* **9**, e015138 (2020).
40. Strodthoff, N. & Strodthoff, C. Detecting and interpreting myocardial infarction using fully convolutional neural networks. *Physiol. Meas.* **40**, 015001 (2019).
41. Raghunath, S. *et al.* Prediction of mortality from 12-lead electrocardiogram voltage data using a deep neural network. *Nat. Med.* **26**, 886–891 (2020).

Acknowledgements

Author contributions

S.A.H., J.L.I., V.T., M.A.R., and J.K.K. conceived the experiment(s). S.A.H., J.L.I., and J.K.K. conducted the experiment(s). S.A.H., J.L.I., V.T., M.A.R., and J.K.K. analyzed the results. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-90285-5.

Correspondence and requests for materials should be addressed to S.A.H.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Appendix A. Published Articles

274

A.15 Paper XV - Pyramid-Focus-Augmentation: Medical Image Segmentation with Step-Wise Focus

Authors: Vajira Thambawita, Steven Hicks, Pål Halvorsen, Michael A. Riegler

Abstract: Segmentation of findings in the gastrointestinal tract is a challenging but also an important task which is an important building stone for sufficient automatic decision support systems. In this work, we present our solution for the Medico 2020 task, which focused on the problem of colon polyp segmentation. We present our simple but efficient idea of using an augmentation method that uses grids in a pyramid-like manner (large to small) for segmentation. Our results show that the proposed methods work as indented and can also lead to comparable results when competing with other methods.

Published: In the Proceedings of MediaEval 2020.

Candidate contributions: Vajira contributed to the conception and design of the pyramid-focus-augmentation study. He conducted all the experiments for this manuscript and analyzed the results with baseline experiments. Vajira published the finding of this study as a python package index (`https://pypi.org/project/pyra-pytorch/`) and GitHub repository (`https://vlbthambawita.github.io/PYRA/`) which can be used by other researchers. He contributed to drafting the manuscript and revising it.

Thesis objectives: Sub-objective I, Sub-objective III

Pyramid-Focus-Augmentation: Medical Image Segmentation with Step-Wise Focus

Vajira Thambawita[1,2], Steven Hicks[1,2], Pål Halvorsen[1,2], Michael A. Riegler[1]

[1]SimulaMet, Norway   [2]Oslo Metropolitan University, Norway

Contact:vajira@simula.no

ABSTRACT

Segmentation of findings in the gastrointestinal tract is a challenging but also important task which is an important building stone for sufficient automatic decision support systems. In this work, we present our solution for the Medico 2020 task, which focused on the problem of colon polyp segmentation. We present our simple but efficient idea of using an augmentation method that uses grids in a pyramid-like manner (large to small) for segmentation. Our results show that the proposed methods work as indented and can also lead to comparable results when competing with other methods.

1 INTRODUCTION

Segmented polyp regions in Gastrointestinal Tract (GI) images [1] can provide detailed analysis to doctors to identify correct areas to proceed with treatments compared to other computer-aided analysis such as classification [2, 9, 10] and detection [7] which provide less detailed information about the exact region and size of the affected area. However, training Deep Learning (DL) models to perform segmentation for medical data is challenging because of the lack of medical domain images as a result of tight privacy restrictions, the high cost for annotating medical data using experts, and a lower number of true positive findings compared to true negatives. In this paper, we present our approach for the participation in the 2020 Medico Segmentation Challenge [4], for which we introduce a novel augmentation technique called **pyramid-focus-augmentation (PYRA)**. PYRA can be used to improve the performance of segmentation tasks when we have a small dataset to train our DL models or if the number of positive findings is small. Further, our method can focus doctors' attention to regions of polyps gradually. In addition to that the output of the method is also adjustable meaning, we could present a lower resolution of the grid if this is sufficient for the task at hand which can help to save processing time. Finally, our technique can also be applied to any segmentation task using any deep learning segmentation model.

2 METHOD

Our method has two main steps: data augmentation with PYRA using pre-defined grid sizes followed by training of a DL model with the resulting augmented data. The source code for our method can be found in our GitHub[1] repository. The development dataset [5]

---

[1]GitHub: https://vlbthambawita.github.io/PYRA/
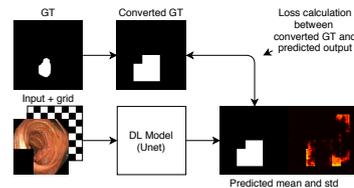
Figure 1: Training steps for a segmentation model with the new augmentation technique.

provided by the organizers has 1000 polyp images with corresponding ground truth masks. We divided it into two parts such that 800 images are used for model training and 200 for testing.

2.1 PYRA Data Augmentation

As the first step in PYRA, we generate checker board grids as illustrated in the first row of Figure 2 with sizes of $N \times N$ with N values of 4, 8, 16, 32, 64, 128 and 256. N should be selected such that $image\_size \% N = 0$. Applying these eight grid augmentations to the training dataset with 800 images increases the training data to $800 \times 8 = 6400$ images.

For the second step, we convert the Ground Truth (GT) segmentation masks into a grid-based representation of the GT corresponding to the grid sizes. For example, if the grid size is 8×8, then the corresponding GT is a 8×8 converted GT.

The transformation of the ground truth masks to gridded masks is performed as following: (i) we divide the gt into the input grid size, (ii) we counted true pixels of each grid cell, (iii) if the number of true pixels is larger than 0, we converted the whole cell into a true cell. An example of a converted GT is depicted on the top of Figure 1.

2.2 Experimental Setup and Model training

We have set up four experiments: Exp-1, Exp-2, Exp-3, and Exp-4 to show the performance of PYRA. Exp-1 and Exp-2 represent two baseline experiments. Exp-1 uses only the 800 training images without any augmentations. In Exp-2, we used general augmentations such as Affine, Coarse Dropout, and Additive Gaussian Noise from the library called *imgaug* [6]. Exp-3 and Exp-4 are using our PYRA with the data from Exp-1 and Exp-2, respectively. The training dataset size was changed from 800 to 6400 after applying PYRA. However, we validated our experiments only using 200 images reserved for testing. We have used one data loader for all experiments to maintain a fair evaluation. The baseline experiments Exp-1 and Exp-2 used the data loader with a grid size of 256×256 which represents the original GT masks without any conversion.
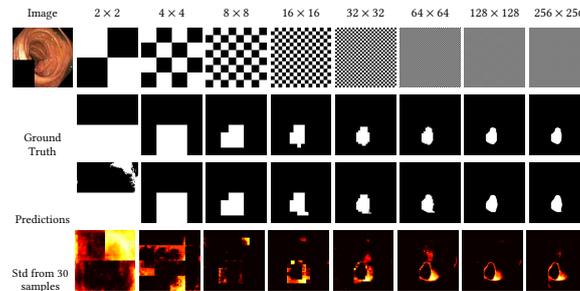
Figure 2: A representation of input and corresponding outputs of grid-augmentation-based segmentation. The first row shows an input image and all grid sizes used as stacked grid image with the input image. The second row represent ground truth. The third and fourth rows show predicted mean and std output images calculated from 30 samples.

Table 1: Result collected from validation data and test data. All test data results were provided by organizers of Medico task in MediaEval 2020.

Method	Validation results		Official test results	
	mIOU	Dice	mIOU	Dice
Exp-1	0.7640	0.8422	0.6934	0.7817
Exp-2	0.7077	0.7957	0.6759	0.7700
Exp-3	**0.7693**	**0.8447**	**0.6981**	**0.7887**
Exp-4	0.6898	0.7822	0.6696	0.7665

We have used the Unet architecture [8] as our DL model to perform the polyp segmentation task. We trained the Unet model with a stacked input using a polyp image and a random grid image selected from the eight sizes. Then, the model was trained to predict converted GT which were formed by converting the real GT into a grid-based GT as in the previous section.

The Unet model used dropout layers with the probability of 0.5. Then, we used our Unet model as a stochastic model to perform Monte Carlo sampling for the validation data. We kept our Unet model in the training state to perform this sampling while predicting the output for the validation data. In the Pytorch library, which is used for all our implementations, we can do this simply by keeping the model state in the `model.train()` state. We iterated 50 times for a single input to predict the output. We calculated the mean from these 50 predictions, which is used as the final prediction for the competition and Standard Deviation (std) images to know the model's confidence for the predictions. The whole training process is illustrated in Figure 1 with an example image and a grid size of 8×8 as an input. However, we submitted the predicted mean images for the gird size of 256×256 which generate predictions with the size of true GT (without any transformations). All the experiments used a fixed learning rate of 0.001 with the RMSprop optimizer [3], which were selected from preliminary experiments.

3 RESULT AND DISCUSSION

Table 1 summarizes the Mean Intersection over Union (mIoU) and the Dice Coefficient (DC) for the validation dataset and the test dataset. The final results to the competition were collected from mean images calculated by sampling 50 outputs for the same input with the grid size of 256. Additionally, we have calculated std images for the validation dataset to show the benefits of using PYRA. Example outputs for a given input image are illustrated in Figure 2.

According to the results in Table 1, Exp-3 which use only Pyramid-focus-augmentation shows the best validation results with mIoU of 0.7693 and DC of 0.8447, and the best test results with mIoU of 0.6981 and DC of 0.7887. The advantage of our Pyramid-focus-augmentation can be identified using the third row of Figure 2 along the fourth row of the same figure. We can see that our model can focus on polyp regions step by step. The third row of Figure 2 shows how our model predicts correct polyp cells in 2×2, 4×4, 8×8, 16×16, 32×32, 64×64, 128×128 and 256×256 grid sizes, respectively. When we compare this row with the last row of the images of std, we can see that the model has high confidence for the identified polyp regions. For example, it shows high confidence (black color region) for the middle part of the polyps. In contrast, our model shows less confidence (yellow color region) for a polyps' outer borders.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a novel augmentation method called Pyramid-focus-augmentation (PYRA), which can be used to train segmentation DL methods. Our method shows a large benefit in the medical diagnosis use-case, by focusing a doctors' attention on regions with findings step by step.

Our experiments did not use post-processing to clean up output corresponding to the input grid. In future work, we will evaluate our approach with additional post-processing steps for smaller grid sizes. For example, we can do convolution operations to the output using a convolutional window equal to the input grid size to clean the results. However, post-processing techniques will not improve the final results when the grid size equals the input images' resolution.

5 ACKNOWLEDGMENT

Appendix A. Published Articles

278

REFERENCES

[1] M. Akbari, M. Mohrekesh, E. Nasr-Esfahani, S. M. R. Soroushmehr, N. Karimi, S. Samavi, and K. Najarian. 2018. Polyp Segmentation in Colonoscopy Images Using Fully Convolutional Network. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 69–72. https://doi.org/10.1109/EMBC.2018.8512197

[2] Steven Alexander Hicks, Pia H Smedsrud, Pål Halvorsen, and Michael Riegler. 2018. Deep Learning Based Disease Detection Using Domain Specific Transfer Learning. *Proc. of MediaEval*.

[3] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. 2012. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. (2012).

[4] Debesh Jha, Steven A. Hicks, Krister Emanuelsen, Håvard Johansen, Dag Johansen, Thomas de Lange, Michael A. Riegler, and Pål Halvorsen. 2020. Medico Multimedia Task at MediaEval 2020: Automatic Polyp Segmentation. In *Proc. of the MediaEval 2020 Workshop*.

[5] Debesh Jha, Pia H Smedsrud, Michael A Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D Johansen. 2020. Kvasir-seg: A segmented polyp dataset. In *International Conference on Multimedia Modeling*. Springer, 451–462.

[6] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, and others. 2020. imgaug. https://github.com/aleju/imgaug. (2020). Online; accessed 01-Nov-2020.

[7] Ji Young Lee, Jinhoon Jeong, Eun Mi Song, Chunae Ha, Hyo Jeong Lee, Ja Eun Koo, Dong-Hoon Yang, Namkug Kim, and Jeong-Sik Byeon. 2020. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. *Scientific Reports* 10, 1 (2020), 8379.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 234–241.

[9] Vajira Thambawita, Debesh Jha, Hugo Lewi Hammer, Håvard D. Johansen, Dag Johansen, Pål Halvorsen, and Michael A. Riegler. 2020. An Extensive Study on Cross-Dataset Bias and Evaluation Metrics Interpretation for Machine Learning Applied to Gastrointestinal Tract Abnormality Classification. *ACM Trans. Comput. Healthcare* 1, 3, Article 17 (June 2020), 29 pages. https://doi.org/10.1145/3386295

[10] Vajira Thambawita, Debesh Jha, Michael Riegler, Pål Halvorsen, Hugo Lewi Hammer, Håvard D Johansen, and Dag Johansen. 2018. The medico-task 2018: Disease detection in the gastrointestinal tract using global features and deep learning. *Proc. of MediaEval* (2018).

A.16 Paper XVI - Impact of Image Resolution on Convolutional Neural Networks Performance in Gastrointestinal Endoscopy

Authors: Vajira Thambawita, Steven Hicks, Inga Strümke, Michael Riegler, Pål Halvorsen, Sravanthi Parasa

Abstract: Convolutional neural networks (CNNs) are increasingly used to improve and automate processes in gastroenterology, like the detection of polyps during a colonoscopy. An important input to these methods is images and videos. Up until now, no well-defined, common understanding or standard regarding the resolution of the images and video frames has been defined, and to reduce processing time and resource requirements, images are today almost always down-sampled. However, how such down-sampling and the image resolution influence the performance in context with medical data is unknown. In this work, we investigate how the resolution relates to the performance of convolutional neural networks. This can help set standards for image or video characteristics for future CNN based models in gastrointestinal endoscopy.

Published: AGA, DDW Abstract Issue, 2021

Candidate contributions: Vajira contributed to the conception and design of this abstract. He conducted all the experiments presenting in this study and he tested the effect of image resolution for deep neural networks using two different well-known neural networks, namely ResNet-151 and DenseNet-161. Vajira contributed to analyzing the results collected from these experiments. He contributed to drafting and revising the abstract.

Thesis objectives: Sub-objective III

IMPACT OF IMAGE RESOLUTION ON CONVOLUTIONAL NEURAL NETWORKS PERFORMANCE IN GASTROINTESTINAL ENDOSCOPY

Thambawita, Vajira Lasantha[1, 3]; Hicks, Steven[1, 3]; Strümke, Inga[1]; Riegler, Michael Alexander[1]; Halvorsen, Pål[1, 3]; Parasa, Sravanthi[2]

[1.] Host, Simula Metropolitan Center for Digital Engineering AS, Oslo, Norway.
[2.] Swedish Medical Group, Seattle, WA, United States.
[3.] OsloMet - storbyuniversitetet, Oslo, Akershus, Norway.

Introduction: Convolutional neural networks (CNNs) are increasingly used to improve and automate processes in gastroenterology, like the detection of polyps during a colonoscopy. An important input to these methods is images and videos. Up until now, no well-defined, common understanding or standard regarding the resolution of the images and video frames has been defined, and to reduce processing time and resource requirements, images are today almost always down-sampled. However, how such down-sampling and the image resolution influence the performance in context with medical data is unknown. In this work, we investigate how the resolution relates to the performance of convolutional neural networks. This can help set standards for image or video characteristics for future CNN based models in gastrointestinal endoscopy.

Methods: This study examines the changes in the performance of CNNs when trained with different resolutions. For all experiments, we rely on the Kvasir data set, consisting of 10,662 GI images from 23 different findings. We evaluate two state-of-the-art CNN models, ResNet-152 and DenseNet-161, for classification under quality distortions with image resolutions for training and testing ranging from 32×32 to 512×512 pixels as shown in Figure 1. For training the models transfer learning is performed with ImageNet weights. The model performance is evaluated using two-fold cross-validation and F1-score, MCC, precision, and sensitivity as metrics.

Results: Increased performance was observed with higher image resolution for all findings in the data set. Lower resolution has a significantly lower performance with an MCC of 0.34 for the lowest and 0.9 for the highest. Table 1 shows the evaluation results in terms of precision, sensitivity, F1-score and MCC for the evaluated ResNet-152 and DenseNet-161 models. The presented numbers are the average over both folds in the cross-validation. Increasing the resolution leads to increased performance measured in almost all metrics. There is a slight decrease in sensitivity for the highest resolution, but taking MCC into account, there is still an overall improvement. For both CNNs, we observe the same behavior.

Conclusion: Different image resolutions and their effect on CNNs are explored. We show that image resolution has a clear influence on the performance which calls for standards in the field in the future. Currently, CNNs usually operate on low to mid-level resolutions. Higher resolution data sets might require new methods, architectures and hardware. As hardware improvements and algorithmic advances continue to occur, developing deep learning applications for endoscopy at higher image resolutions becomes increasingly feasible. Nevertheless, although the full potential of high-resolution data sets might not be exploitable yet, it is evidently important to collect data with the highest resolution possible.

(a) 32 × 32   (b) 64 × 64   (c) 128 × 128

(d) 256 × 256   (e) 512 × 512

Figure 1: Examples of an image with the different resolutions used for the experiments for this abstract. Clear differences in the level of details that detectable can be observed. Note that for this figure all resolutions are re-scaled to the same size to show quality differences.

	Resolution	Precision	Sensitivity	F1-score	MCC (R_K)
ResNet-152	32 × 32	0.525	0.514	0.511	0.808
	64 × 64	0.573	0.576	0.573	0.856
	128 × 128	0.614	0.614	0.611	0.887
	256 × 256	0.618	0.619	0.618	0.897
	512 × 512	0.633	0.617	0.611	**0.900**
DenseNet-161	32 × 32	0.541	0.540	0.537	0.824
	64 × 64	0.572	0.575	0.570	0.855
	128 × 128	0.603	0.601	0.600	0.886
	256 × 256	0.623	0.614	0.615	0.900
	512 × 512	0.645	0.634	0.635	**0.900**

Table 1: Average ResNet-152 and DenseNet-161 results for both cross-validation splits. Best MCC score in bold.

IMPACT OF IMAGE RESOLUTION ON CONVOLUTIONAL NEURAL NETWORKS PERFORMANCE IN GASTROINTESTINAL ENDOSCOPY

V. Thambawita[1,3], S. Hicks[1,3], I. Strümke[1], M. Riegler[1], P. Halvorsen[1,3], and S. Parasa[2].

[1]Host, Simula Metropolitan Center for Digital Engineering AS, Oslo, Norway.

[2]Swedish Medical Group, Seattle, WA, United States.

[3]Oslo Metropolitan University, Oslo, Norway.

INTRODUCTION

Convolutional neural networks (CNNs) are increasingly used to improve and automate processes in gastroenterology, like the detection of polyps during a colonoscopy.

An important input to these methods is images and videos. Up until now, no well-defined, common understanding or standard regarding the resolution of the images and video frames has been defined, and to reduce processing time and resource requirements, images are today almost always down-sampled.

However, how such down-sampling and the image resolution influence the performance in context with medical data is unknown.

AIMS:

In this work, we investigate how image resolution relates to the performance of convolutional neural networks. This can help set standards for image or video characteristics for future CNN based models in gastrointestinal endoscopy.

METHODS

This study examines the changes in the performance of CNNs when trained with different resolutions.

Using the Kvasir data set, consisting of 10,662 GI images from 23 different findings we evaluate two state-of-the-art CNN models, ResNet-152 and DenseNet-161, for classification under quality distortions with image resolutions for training and testing ranging from 32×32 to 512×512 pixels as shown in Figure 1.

For training the models, transfer learning is performed with ImageNet weights. The model performance is evaluated using two-fold cross-validation and F1-score, MCC, precision, and sensitivity as metrics.

RESULTS

- Increased performance was observed with higher image resolution for all findings in the data set.
- **Lower resolution has a significantly lower performance with an MCC of 0.34 for the lowest and 0.9 for the highest.** Table 1 shows the evaluation results in terms of precision, sensitivity, F1-score and MCC for the evaluated ResNet-152 and DenseNet-161 models.
- **Increasing the resolution leads to increased performance measured in almost all metrics.** There is a slight decrease in sensitivity for the highest resolution, but taking MCC into account, there is still an overall improvement. For both CNNs, we observe the same behavior. The presented numbers are the average over both folds in the cross-validation.

	Resolution	Precision	Sensitivity	F1-score	MCC (R_K)
ResNet-152	32 × 32	0.525	0.514	0.511	0.808
	64 × 64	0.573	0.576	0.573	0.856
	128 × 128	0.614	0.614	0.611	0.887
	256 × 256	0.618	0.619	0.618	0.897
	512 × 512	0.633	0.617	0.611	**0.900**
DenseNet-161	32 × 32	0.541	0.540	0.537	0.824
	64 × 64	0.572	0.575	0.570	0.855
	128 × 128	0.603	0.601	0.600	0.886
	256 × 256	0.623	0.614	0.615	0.900
	512 × 512	0.645	0.634	0.635	**0.900**

Table 1: Average ResNet-152 and DenseNet-161 results for both cross-validation splits. Best MCC score in bold.
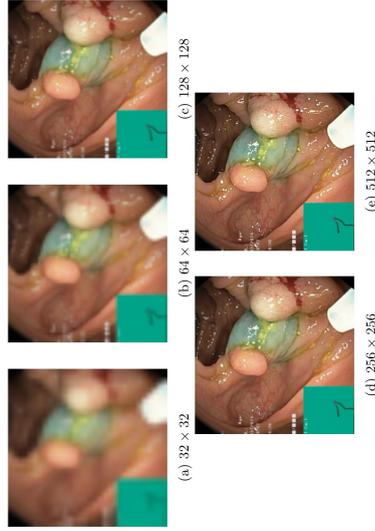

Figure 1: Examples of an image with the different resolutions used for the experiments for this abstract. Clear differences in the level of details that detectable can be observed. Note that for this figure all resolutions are re-scaled to the same size to show quality differences.

CONCLUSIONS

- Different image resolutions and their effect on CNNs are explored.
- **We show that image resolution has a clear influence on the performance which calls for standards in the field in the future.** Currently, CNNs usually operate on low to mid-level resolutions. Higher resolution data sets might require new methods, architectures and hardware.
- As hardware improvements and algorithmic advances continue to occur, developing deep learning applications for endoscopy at higher image resolutions becomes increasingly feasible. Nevertheless, although the full potential of high-resolution data sets might not be exploitable yet, it is evidently important to collect data with the highest resolution possible.

simulamet
Simula Metropolitan Center for Digital Engineering AS

SWEDISH MEDICAL GROUP

OsloMet — OSLO METROPOLITAN UNIVERSITY

DDW 2021
May 21-23 | VIRTUAL

A.17 Paper XVII - On Evaluation Metrics for Medical Applications of Artificial Intelligence

Authors: Steven A. Hicks, Inga Strümke, **Vajira Thambawita**, Malek Hammou, Michael A. Riegler, Pål Halvorsen, Sravanthi Parasa

Abstract: Clinicians and model developers need to understand how proposed machine learning (ML) models could improve patient care. In fact, no single metric captures all the desirable properties of a model and several metrics are typically reported to summarize a model's performance. Unfortunately, these measures are not easily understandable by many clinicians. Moreover, comparison of models across studies in an objective manner is challenging, and no tool exists to compare models using the same performance metrics. This paper looks at previous ML studies done in gastroenterology, provides an explanation of what different metrics mean in the context of the presented studies, and gives a thorough explanation of how different metrics should be interpreted. We also release an open source web-based tool that may be used to aid in calculating the most relevant metrics presented in this paper so that other researchers and clinicians may easily incorporate them into their research.

Published: Submitted for publication, Preprint is available at medRxiv.

Candidate contributions: Vajira contributed to designing and developing the concept of this paper. He also contributed to the main analysis of the results collected from the literature reviews. Also, Vajira contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective III

On evaluation metrics for medical applications of artificial intelligence

Steven A. Hicks[1,2*], **Inga Strümke**[1], **Vajira Thambawita**[1,2], **Malek Hammou**[1], **Michael A. Riegler**[1], **Pål Halvorsen**[1,2], **and Sravanthi Parasa**[3]

[1]SimulaMet, Oslo, Norway
[2]Oslo Metropolitan University, Oslo, Norway
[3]Swedish Medical Center, Seattle, USA
[*]contact/corresponding author steven@simula.no

ABSTRACT

Clinicians and model developers need to understand how proposed machine learning (ML) models could improve patient care. In fact, no single metric captures all the desirable properties of a model and several metrics are typically reported to summarize a model's performance. Unfortunately, these measures are not easily understandable by many clinicians. Moreover, comparison of models across studies in an objective manner is challenging, and no tool exists to compare models using the same performance metrics. This paper looks at previous ML studies done in gastroenterology, provides an explanation of what different metrics mean in the context of the presented studies, and gives a thorough explanation of how different metrics should be interpreted. We also release an open source web-based tool that may be used to aid in calculating the most relevant metrics presented in this paper so that other researchers and clinicians may easily incorporate them into their research.

Keywords: Gastroenterology, standardization, machine learning, evaluation metrics

Improving healthcare applications and supporting decision making for medical professionals using methods from Artificial Intelligence (AI), specifically Machine Learning (ML), is a rapidly developing field with numerous retrospective studies being published every week. We also observe an increasing number of prospective studies involving large multi-center clinical trials testing ML systems' suitability for clinical use. The technical and methodological maturity of the different areas varies, radiology and dermatology being examples of the more advanced ones[1]. In addition to these two examples, we observe a recent surge of studies in the field of gastroenterology[2]. Therefore, the present study focuses on the current development in gastroenterology due to its timeliness. However, our discussions, recommendations, and proposed tool are valid and useful in every clinical field adopting and employing ML-based systems.

The use of ML in gastroenterology is expected to significantly improve detection and characterization of colon polyps and other precancerous lesions of the Gastrointestinal (GI) tract[3]. These potential advances are mainly expected from artificial neural networks, specifically deep learning-based methods[4]. Safe and efficient adoption of ML tools in clinical gastroenterology requires a thorough understanding of the performance metrics of the resulting models and confirmation of their clinical utility[5].

Creating strong evidence for the usefulness of ML models in clinical settings is an involved process. In addition to the relevant epidemiological principles, it requires a thorough understanding of the properties of the model itself and its performance. However, despite increased interest in ML as a medical tool, understanding of how such models work and how to aptly evaluate them using different metrics is widely lacking. In this article, we use examples of metrics and evaluations drawn from a variety of peer-reviewed and published studies in gastroenterology to provide a guide explaining different evaluation metrics, including how to interpret them. Note that we do not discuss the quality of these studies, but merely use them to discuss how different metrics give different interpretations of the quality of an ML model.

The main contributions are: We present a detailed discussion on metrics commonly used for evaluating ML classifiers, examine existing research using ML in gastroenterology along with reported metrics, and we discuss the different metrics' interpretations, usefulness, and shortcomings. To this end, we recalculate the reported metrics and calculate additional ones to further analyze the performance of the presented methods. Additionally, we present a web-based open source tool intended to let researchers perform metrics calculations easily, both for their own and other reported results to allow for comparison. The tool is accessible via `www.medimetrics.no`, and the source code via `github.com/simula/medimetrics`.

Metrics

The relevant quantities for calculating the metrics for a binary classifier are the four entries in the confusion matrix

$$\mathbf{M} = \begin{pmatrix} \mathrm{TP} & \mathrm{FN} \\ \mathrm{FP} & \mathrm{TN} \end{pmatrix}, \tag{1}$$

which are introduced below. Note that we limit ourselves to binary classification in this study, which is at the time of writing most common for medical applications, e.g., cancer/no-cancer or polyp/no-polyp. Some metrics have different interpretations in the context of evaluating multi-class classification methods. Although these discussions fall outside the scope of the present one, the underlying principles still apply in multi-class settings. Furthermore, some methods have their metrics calculated on a per-finding basis, meaning there can be multiple instances for one image, and hence more positive samples than total samples (e.g., images or videos) in a data set.

True Positive (TP)   The true positive denotes the number of correctly classified positive samples. For example, the number of frames containing a polyp correctly predicted as having a polyp.

True Negative (TN)   The true negative denotes the number of correctly classified negative samples. For example, the number of frames not containing a polyp correctly predicted as not having a polyp.

False Positive (FP)   The false positive denotes the number of samples incorrectly classified as positive. For example, the number of frames not containing a polyp incorrectly predicted as having a polyp.

False Negative (FN)   The false negatives denotes the number of samples incorrectly classified as negative. For example, the number of frames containing a polyp incorrectly predicted as not having a polyp.

Accuracy (ACC)   The accuracy is the ratio between the correctly classified samples and the total number of samples in the evaluation data set. This metric is among the most commonly used in applications of ML in medicine, but is also known for being misleading in case of different class proportions, since simply assigning all samples to the prevalent class is an easy way of achieving high accuracy. The accuracy is bounded to $[0,1]$, where 1 represents predicting all positive and negative samples correctly, and 0 represents predicting none of the positive or negative samples correctly.

$$\mathrm{ACC} = \frac{\text{\# correctly classified samples}}{\text{\# all samples}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}} \tag{2}$$

Recall (REC)   The recall, also known as the sensitivity or True Positive Rate (TPR), denotes the rate of positive samples correctly classified, and is calculated as the ratio between correctly classified positive samples and all samples assigned to the positive class. The recall is bounded to $[0,1]$, where 1 represents perfectly predicting the positive class, and 0 represents incorrect prediction of all positive class samples. This metric is also regarded as being among the most important for medical studies, since it is desired to miss as few positive instances as possible, which translates to a high recall.

$$\mathrm{REC} = \frac{\text{\# true positive samples}}{\text{\# samples classified positive}} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{3}$$

Specificity (SPEC)   The specificity is the negative class version of the recall (sensitivity) and denotes the rate of negative samples correctly classified. It is calculated as the ratio between correctly classified negative samples and all samples classified as negative. The specificity is bounded to $[0,1]$, where 1 represents perfectly predicting the negative class, and 0 represents incorrect prediction of all negative class samples.

$$\mathrm{SPEC} = \frac{\text{\# true negative samples}}{\text{\# samples classified negative}} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{4}$$

Precision (PREC)   The precision denotes the proportion of the retrieved samples which are relevant and is calculated as the ratio between correctly classified samples and all samples assigned to that class. The precision is bounded to $[0,1]$, where 1 represents all samples in the class correctly predicted, and 0 represents no correct predictions in the class.

$$\mathrm{PREC} = \frac{\text{\# samples correctly classified}}{\text{\# samples assigned to class}} = \frac{\mathrm{TC}}{\mathrm{TC} + \mathrm{FC}}, \tag{5}$$

where C denotes "class", and can in binary classification be either positive (P) or negative (N). The terms precision and Positive Predictive Value (PPV) are often used interchangeably.

2/10

Appendix A. Published Articles

F1 score (F1)   The F1 score is the harmonic mean of precision and recall, meaning that it penalizes extreme values of either. This metric is not symmetric between the classes, i.e., it depends on which class is defined as positive and negative. For example, in the case of a large positive class and a classifier biased towards this majority, the F1 score, being proportional to TP, would be high. Redefining the class labels so that the negative class is the majority and the classifier is biased towards the negative class would result in a low F1 score, although neither the data nor the relative class distribution have changed. The F1-score is bounded to $[0, 1]$, where 1 represents maximum precision and recall values and 0 represents zero precision and/or recall.

$$\text{F1} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{6}$$

Matthews Correlation Coefficient (MCC)   Pearson's correlation coefficient[6], takes on a particularly simple form in the binary case. This special case has been coined the MCC[7], and become popular in ML settings for its favorable properties in cases of imbalanced classes[8]. It is essentially a correlation coefficient between the true and predicted classes, and achieves a high value only if the classifier obtains good results in all the entries of the confusion matrix Equation 1. The MCC is bounded to $[-1, 1]$, where a value of 1 represents perfect prediction, 0 random guessing and -1 total disagreement between prediction and observation.

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}} \tag{7}$$

PPV   The PPV is the ratio between correctly classified positive samples and all samples classified as positive, and equals the precision for the positive class. The PPV is bounded to $[0, 1]$, where 1 represents all positive samples predicted correctly, and 0 represents no correct positive class predictions.

$$\text{PPV} = \frac{\text{\# correct positive predictions}}{\text{\# samples classified as positive}} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

Negative Predictive Value (NPV)   The NPV is the ratio between correctly classified negative samples and all samples classified as negative, and equals the precision for the negative class. The NPV is bounded to $[0, 1]$, where 1 represents all negative samples predicted correctly, and 0 represents no correct negative class predictions.

$$\text{NPV} = \frac{\text{\# correct negative predictions}}{\text{\# samples classified as negative}} = \frac{\text{TN}}{\text{TN} + \text{FN}} \tag{9}$$

Threat Score (TS)   The TS, also called the Critical Success Index (CSI), is the ratio between the number of correctly predicted positive samples against the sum of correctly predicted positive samples and all incorrect predictions. It takes into account both false alarms and missed events in a balanced way, and excludes only the correctly predicted negative samples. As such, this metric is well suited for detecting rare events, where the model evaluation should be sensitive to correct classification of rare positive events, and not overwhelmed by many correct identifications of negative class instances. The TS is bounded to $[0, 1]$, where 1 represents no false predictions in either class, and 0 represents no correctly classified positive samples.

$$\text{TS} = \frac{\text{\# correct positive predictions}}{\text{\# correct positive and all false predictions}} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{10}$$

We do not consider the AUROC (Area under the Receiver Operating Characteristic Curve) or AUPRC (Area under the Precision Recall Curve) since these cannot be calculated without access to the model, or from the entries of the confusion matrix. Extensive research has been done on their usefulness, and we refer the interested reader to[9].

Class mixture

Binary classification problems can be expressed in terms of a mixture model, the total data distribution modelled as

$$p(X, \alpha) = \alpha p_P(X) + (1 - \alpha) p_N(X), \tag{11}$$

where X represents data samples, $p_{P/N}$ denotes the positive/negative class distributions, and α the mixture parameter of the positive class, calculated as $\alpha = \frac{N_P}{N_P + N_N}$, with $N_{P/N}$ the total number of positive/negative class data samples. Studies in which the classification threshold of model outputs are tuned using a class imbalanced data set, should investigate how these perform on other class admixtures. This is an important step to assess whether bias towards either class has been introduced and to what extent.

3/10

286

Blinded data

Model development is typically split into three phases. First, the model is trained on a training dataset appropriate for the given task. Second, during training, the model is continuously validated on data not part of the training data, to evaluate the model's performance on unseen data. Last, after the model has finished training, it is tested on a test dataset for which the final metrics should be calculated. Regardless of which metric is used, this can only be as informative as the classifier's performance on the test data. Blinding data, i.e., withholding data from those performing the experiment, is an important tool in many research fields, such as medicine. In some experiment types, it is difficult to achieve blinding, but the analysis in a setting where the data has already been collected can almost always be blinded. Misconceptions regarding the objectivity of statistical analysis should not keep researchers from blinding the data[10]. For ML analyses, such as the ones described in the present work, this means that one should set aside representative data that can be used for testing after the training and tuning processes are finished.

Methods

In the following, we identify a subset of relevant studies for our analysis. Medical studies presenting ML applications often refer to them simply as "AI systems". While AI has certainly received an unprecedented amount of attention over the past years, and presenting systems using this term emphasizes their novelty, the term is imprecise. Hence, we refrain from using this generic term in the following and instead refer to the exact model architecture used.

Study selection

The studies used for this work are chosen based on the following rational considerations. Our starting point is a recent review of AI in gastroenterology[11]. The review contains 138 articles, from which we select five studies that represent existing work using ML in gastroenterology. The selection criteria are as follows

(i) Report sufficient information (many studies report so few metrics that it is not possible to calculate other metrics) for reproducing the reported metrics and calculating metrics not reported.

(ii) Represent different cases of interest for performance metrics discussions.

In addition, we select a recent study reporting results from a large clinical trial, which was not included in the aforementioned review. The following contains a brief description of the selected studies and reported metrics.

Study 1

Hassan et al.[12] introduce an unspecified "AI system" called GI-Genius to detect polyps, trained and validated using $2,684$ videos from 840 patients collected from white-light endoscopy. All 840 patients are randomly split into two separate data sets, one for training and one for validation. The validation data set contains 338 polyps from 105 patients, where 168 of the identified polyps are either adenomas or sessile serrated adenomas. The authors report a sensitivity of 99.7% as the main performance metric, which is calculated from 337 TPs out of the total 338 positive samples. From this, readers are likely to conclude that only one FN instance is identified in the validation set. No other metrics are reported, and while it is reported that each colonoscopy contains $50,000$ frames, no further details are given on the exact number of frames per video.

Study 2

Mossotto et al.[13] use several ML models to classify diseases commonly found in the GI tract, using endoscopic and histological data. The data consist of 287 patients, from which 178 cases are Crohn's Disease (CD), 80 cases are Ulcerative Colitis (UC), and 29 cases are Unclassified Inflammatory Bowel Disease (IBDU). Results are shown from unsupervised (clustering) and supervised learning. The latter is used to classify CD and UC patients. For this, the data is divided into a model construction set consisting of 210 patients (CD = 143, UC = 67), a model validation set of 48 patients (CD = 35, UC = 13), and an IBDU reclassification set containing 29 IBDU patients. The model is thus not trained on IBDU data, and the latter data set is excluded from the present discussion. The model construction set is stratified into a discovery set used to tune the parameters for CD versus UC discovery, and one for training and testing. For the best performing supervised model, tested on the test set, an accuracy of 82.7%, a precision of 0.91, a recall of 0.83 and an F1 score of 0.87 are reported, see Table 2 in[13]. On the validation set, the reported numbers are an accuracy of 83.3%, a precision of 0.86, a recall of 0.83, and an F1 score of 0.84, see Table 3 in[13]. These reported results are also listed in Table 1 under study 2.

Study 3

Byrne et al.[14] introduce a Convolutional Neural Network (CNN) to differentiate diminutive adenomas from hyperplastic polyps. They define the four classes NICE type 1, NICE type 2, no polyp, and unsuitable. The training data set contains 223 polyp videos, consisting of $60,089$ frames in total, with 29% containing hyperplastic polyps, 53% containing adenomas polyps, and 18% containing no polyps. The model is tested on 158 videos, and 32 of these are removed due to the reported instances in the

4/10

Appendix A. Published Articles

Table 1. The reported metrics of the selected studies. The *STUDY* column represents each of the five studies selected for metric recalculation. The *SET* column is the different metrics calculated for the same set of data. The *REPORTED* column is how the metrics were reported in the respective study. To refer to the tables in each respective paper, we use T to refer to the table number and R for the row number. The *EVALUATION* column is the method used to generate the metrics. The *TOTAL* column is the total number of samples used in the metrics calculations. The *POS* and *NEG* columns represent the total number of positive and negative samples, respectively. The remaining columns correspond the aforementioned metric acronyms described in the main text.

STUDY	SET	REPORTED	EVALUATION	TOTAL	POS	NEG	TP	TN	FP	FN	ACC	PREC	REC	F1	SPEC	MCC	NPV	TS
1	1	In-text	Per-finding	-	338	-	337	-	-	1	-	-	1.00	-	-	-	-	-
2	1	T2.R1	Per-frame	210	143	67	-	-	-	-	0.71	0.89	0.68	0.75	-	-	-	-
	1	T2.R2	Per-frame	210	143	67	-	-	-	-	0.77	0.81	0.88	0.83	-	-	-	-
	1	T2.R3	Per-frame	210	143	67	-	-	-	-	0.87	0.91	0.84	0.87	-	-	-	-
	2	T3.R1	Per-frame	48	13	35	-	-	-	-	-	0.65	0.85	0.73	-	-	-	-
	2	T3.R2	Per-frame	48	35	13	-	-	-	-	-	0.94	0.83	0.88	-	-	-	-
	2	T3.R3	Per-frame	-	-	-	-	-	-	-	0.83	0.86	0.83	0.84	-	-	-	-
3	1	T1	Per-frame	106	-	-	65	33	7	1	0.94	0.90	0.98	-	0.83	-	0.97	-
4	1	T2.R1	Per-frame	6,000	6,000	0	5,663	0	251	337	-	-	0.94	-	-	-	-	-
	1	T2.R2	Per-frame	1,414	1,414	0	1,296	0	41	118	-	-	0.92	-	-	-	-	-
	1	T2.R3	Per-frame	21,572	0	21,572	0	20,691	1,004	0	-	NA	-	-	0.96	-	-	-
	2	T2.R4	Per-frame	-	-	-	570	0	42	76	-	-	0.88	-	-	-	-	-
	3	In-text	Per-frame	60,914	-	-	-	-	-	-	-	-	0.92	-	-	-	-	-
	4	In-text	Per-frame	1,072,483	0	1,072,483	0	-	-	0	-	-	-	-	0.95	-	-	-
5	1	T1	Per-frame	-	-	-	3723	4735	262	930	0.88	-	0.80	-	0.95	-	-	-

videos. Three are sessile serrated polyps, 25 are identified as normal tissue or lymphoid aggregate, two are fecal material, one video is corrupted, and two contain multiple polyp frames. The resulting 125 videos are used to evaluate the CNN model again, which is unable to confidently[1] identify 19 of the 125 polyps. The 19 videos on which the model does not reach this confidence threshold are therefore removed from the test data set, and the model is evaluated using the remaining 106 videos. Finally, after this data filtering, the model achieves an accuracy of 94%, a sensitivity of 98%, a specificity of 83%, a PPV of 90% and an NPV of 97%, see Table 1 under study 3.

Study 4
Wang et al.[15] present a near real-time deep learning-based system for detecting colon polyps using videos from colonoscopies. The model is trained on data collected from 1,290 patients and validated on 27,113 colonoscopy images from 1,138 patients showing at least one detected polyp. It is then tested on a public database containing 612 images with polyps[16]. As the presented method is able to differentiate between different polyps within the same image, there may be more true positives than images in the data set. This is also the reason why the metrics are reported on a per-image basis. The reported results show that the method is highly effective, with a per-image-sensitivity of 94.38% and a per-image-specificity of 95.92%. As the metrics are reported separately for images containing polyps and those that do not, recalculating the metrics as presented provides an inaccurate representation of the model's actual performance. This is because there are either no true positives or no true negatives, depending on the metrics used.

Study 5
Sakai et al.[17] propose a CNN-based system to automatically detect gastric cancer in images from colonoscopies. The model is trained on a data set of 172,555 images containing gastric cancer and 176,388 images of normal colon. For evaluation, the model is tested on 4,653 cancer images and 4,997 normal images, on which it achieves an accuracy of 87.6%, a sensitivity of 80.0%, a specificity of 94.8% (see Table 1 in[17]), and a PPV of 93.4%. A method capable of distinguishing which regions of an image contain signs of gastric cancer is also presented. This method uses a sliding-window approach, where the model predicts the presence of gastric cancer in specific regions of the image to generate a block-like heat map covering the afflicted areas. This detection model is tested on 926 images, where it achieves an accuracy of 89.9% on cancer images and an accuracy of 70.3% on normal images.

Results

In this section, we perform a recalculation of all reported and missing metrics in the selected studies. Based on this, we discuss the usefulness of different metrics and how to obtain a realistic and complete picture of the performance of a classifier. This is

---

[1]For each image, the model gives a confidence value ranging from 0 to 1. If the confidence level is below 0.5, the model is not considered confident enough to keep the prediction.

A.17. Paper XVII - On Evaluation Metrics for Medical Applications of Artificial Intelligence

Table 2. The recalculated metrics of the selected papers. Columns represent the same as described in Table 1.

STUDY	SET	REPORTED	EVALUATION	TOTAL	POS	NEG	TP	TN	FP	FN	ACC	PREC	REC	F1	SPEC	MCC	NPV	TS
1	1	In-text	Per-polyp	16,900,000	338	16,899,662	337	16,730,662	169,000	1	0.99	0.00	1.00	0.00	0.99	0.04	1.00	0.00
	1	Calculated	Per-frame	16,900,000	84,500	16,815,500	84,500	16,646,500	169,000	250	0.99	0.33	1.00	0.50	0.99	0.57	1.00	0.33
2	1	T2.R1	Per-frame	210	143	67	97	55	12	46	0.71	0.89	0.68	0.77	0.82	0.47	0.55	0.63
	1	T2.R2	Per-frame	210	143	67	116	40	27	27	0.77	0.81	0.88	0.81	0.59	0.40	0.59	0.68
	1	T2.R3	Per-frame	210	143	67	130	54	13	13	0.87	0.91	0.84	0.91	0.81	0.72	0.81	0.83
	2	T3.R1	Per-frame	48	13	35	11	29	6	2	0.84	0.65	0.85	0.74	0.83	0.63	0.94	0.58
	2	T3.R2	Per-frame	48	35	13	29	11	2	6	0.84	0.94	0.83	0.88	0.86	0.64	0.65	0.79
	2	T3.R3	AVG	-	-	-	-	-	-	-	0.84	0.80	0.84	0.81	0.84	0.63		0.69
	2	Calculated	WAVG	-	-	-	-	-	-	-	0.84	0.86	0.84	0.84	0.85	0.64	0.73	0.73
3	1	T1	Per-frame	106	66	40	65	33	7	1	0.92	0.90	0.98	0.94	0.83	0.84	0.97	0.89
4	1	T2.R1	Per-frame	6,000	6,000	0	5,663	0	251	337	0.94	0.96	0.94	0.95	NA	−0.05	0	0.91
	1	T2.R2	Per-frame	1,414	1,414	0	1,296	0	41	118	0.92	0.97	0.92	0.94	NA	−0.05	0	0.89
	1	T2.R3	Per-frame	21,572	0	21,695	0	20,691	1,004	0	0.95	0	NA	0	0.95	NA	1	0
	1	Calculated	Combined	27,572	6,000	21,695	5,663	20,691	1,255	337	0.95	0.82	0.94	0.88	0.95	0.84	0.98	0.78
	1	Calculated	Biased POS	27,572	6,000	21,695	6,000	0	21,695	0	0.22	0.22	1	0.36	0	NA	NA	0.22
	1	Calculated	Biased NEG	27,572	6,000	21,695	0	21,695	0	6,000	0.78	NA	0	0	1	NA	0.78	0
	2	T2.R4	Per-frame	646	646	42	570	0	42	76	0.83	0.93	0.88	0.91	0	−0.09	0	0.83
	3	In-text	Per-frame	60,914	-	-	-	-	-	-	-	-	0.92	-	-	-	-	
	4	In-text	Per-frame	1,072,483	0	1,072,483	0	1,023,149	49,334	0	0.95	0	NA	0	0.95	NA	1	-
5	1	T1	Per-frame	9,650	4,653	4,997	3,723	4,735	262	930	0.88	0.93	0.80	0.86	0.95	0.76	0.84	0.76

done by extracting reported numbers and metrics from each study and using these to calculate additional metrics, which gives additional perspectives on the possible evaluations and could lead to different conclusions. In some cases, assumptions must be made in order to calculate metrics or assess model performances under different conditions. All assumptions made in this study are detailed in the relevant discussions.

We also present our freely available online tool, which allows medical experts to calculate all presented metrics from classifier predictions, or those which can be calculated from a subset of metrics. This can be used for a variety of different usage scenarios, like gaining a better understanding of studies using ML classifiers, calculate missing metrics for studies which do not report them, to double-check calculations, and to calculate metrics for new studies.

Precision and recall

To reproduce the results of *Study 1*[12], it is necessary to make some assumptions. Primarily, no information is given regarding the total number of frames for all videos, but as an average of $50,000$ frames per video is reported, we use this to calculate the total number. Further, we calculate two sets of metrics, see Table 2 under study 1. For the first row, we calculate TP=337, TN=16,730,662, FP=169,000, and FN=1 using the same per polyp detection evaluation as *Study 1*. In the second row, we assume that ten seconds around the polyp are either detected correctly or missed with a frame rate of 25 fps. This yields TP=84,500, TN=16,646,500, FP=169,000, and FN=250, which are used in our calculations. FP for both calculations are obtained based on the reported 1% FPs per video, i.e., $\frac{50,000}{100} = 500$. These assumptions yield two sets of results for the evaluation, which, if considered jointly, give a more thorough understanding of the performance. In any case, the reported values are not sufficient for reproducibility without making assumptions.

Assuming the most optimistic case amounts to mixture components of 0.995 and 0.005 for the positive and negative classes, respectively, meaning extremely imbalanced classes. The authors report a recall of 99.7%, in which case we calculate a precision of 0.33. Clearly, the recall must be interpreted with care in cases of strongly imbalanced classes. The reason is that precision and recall are both proportional to TP, but have an inverse mutual relationship: High precision requires low FP, so a classifier maximizing precision will return only very strong positive predictions, which can result in positive events missed. On the other hand, high recall is achieved by assigning more instances to the positive class, to achieve a low FN. Whether to maximize recall or precision depends on the application: Is it most important to identify *only* relevant instances, or to make sure that all relevant instances are identified? Regardless of which is the case, this should be clearly stated, and both metrics should be reported. The balance between the two has to be based on the medical use case and associated requirements. For example, some false alarms are acceptable in cancer detection, since it is crucial to identify all positive cases. On the other hand, for the identification of less severe disease with high prevalence, it can be important to achieve the highest possible precision. A low precision combined with a high recall implies that the classifier is prone to set of false alarms (FPs), which can result in an overwhelming manual workload and time wasted.

Mixture parameter dependent tuning

In *Study 2*, Mossotto et al.[13] split the model construction data set into two subsets of equal size and class distribution, with mixture components 0.68 and 0.32 for the two classes CD and UC. One of these subsets is used to tune parameters to maximize CD versus UC classification, meaning that the classification task is done with the underlying assumption that the class admixture will remain constant. This is trivially true for the training and test data, being the other of the two subsets, but not for the

6/10

Appendix A. Published Articles

validation set, where the corresponding mixture components are 0.73 and 0.27. The authors do not mention the deviation from the tuned admixture, nor do they investigate systematically how much a given deviation affects the reported performance metrics. Without access to the resulting model, this cannot be investigated further in this article or by other interested readers. Consequently, there is no way of knowing the sensitivity the presented method has to the class admixture.

The study does not report any of the confusion matrix entries, thus not facilitating the process of reproducible results. However, we can derive the TN by multiplying the total number of positive samples with the recall, $TP = REC \times (TP + FN)$. The FP are then obtained via $FP = \frac{TP}{PREC} - TP$. From this, we can calculate the reported and missing metrics. The authors report the positive class precision, which represents the PPV. In addition, the NPV should also have been reported for completeness. As shown in Table 2, the NPV is lower for the validation data sets where the precision is high, and vice versa. Calculating the MCC, this is stable around 0.63 over all validation sets listed in Table 3 of[13], although not as high as any of the reported metrics.

Blinded data

In *Study 3*, Byrne et al.[14] remove data samples for which the classifier does not achieve high confidence, as well as videos with more than one polyp. They calculate the model's performance for the different videos in the study, and remove the ones on which the model performs poorly. As such, the results reported from the study concerns a very specific selection of their data, made after the model has been adjusted. Excluding data on which a model performs poorly leads to a misrepresentation of its abilities and should not be done. If the classification task is too difficult or the removed data was faulty, this should instead be reported, and a new classifier should be trained for a more limited task. The metrics reported from a study should be calculated *after* the final model calibration and subsequent testing on blinded data.

Negative and positive class performance

In *Study 4*, Wang et al.[15] report high per-image-sensitivity (recall) values of 0.94, 0.92 and 0.92, see Table 2 in[15], or metric set 1 under study 4 in Table 1. For the first two of these, sufficient numbers are reported to reproduce the reported metrics, as well as to calculate the corresponding positive class precisions, which are reported as 0.96 and 0.97, respectively. In the third case, the positive class precision cannot be calculated since the positive and negative samples are separated for the test. No explanation or reason is given regarding why the tests are performed only on the separated classes and not together, which would give a better overall impression of the performance. For the first data set, it is not clear how the numbers are calculated, as no test set is mentioned. This could mean that the reported sensitivity is calculated on the training data.

Rows five to seven in Table 2 show the results achieved when combining the negative and positive class samples. Since we do not know if the obtained model is biased towards the negative or positive class, we present three evaluations: In row five, we assume that the positive and negative results can simply be combined, which gives an overall MCC of 0.84 and NPV of 0.98, indicating good performance. In rows six and seven, we assume that the model is biased towards the positive or negative class, respectively. The resulting MCCs are both -0.05 and the NPVs both 0. This means that the classifier, which seemed to perform exceptionally well based on the reported numbers, is actually severely under-performing on the negative class. Besides these ambiguities, the results for the first three data sets indicate strong performance, but using the same numbers to calculate metrics more sensitive to bias, reveals severe under-performance (see Tables 1 and 2 for all reported and calculated numbers).

While detection and classification are in principle the same task for a fixed number of instances per class, the study in[15] faces a challenge: The negative class is unbounded, i.e., the number of negative instances is undefined. The more sensitive the classifier is, the larger the negative class effectively becomes, as the classifier generates FP instances, and the negative class instances can be calculated as $Neg = TN + FP$. In general, evaluating without clearly defining boundaries for the classes is risky, as it can lead to an unclear impression of the model performance, in either the positive or negative direction. It is also nearly impossible for follow up studies to reproduce and compare results.

Without a well-defined number of true negatives in a video (or set of images), each of the frames not containing a polyp and each of the pixels not being part of a polyp are in principle true negatives. Optimally, the classes should instead be balanced, at best with a mixture parameter of 0.5. If this is not possible, the study should at least be based on well-motivated assumptions informed by real-world properties. For example, a standard colonoscopy contains on average *n* number of frames and *m* polyps found per examination. Most colonoscopies take less than an hour, so assuming a 24 hour time frame would be an unreasonable assumption within such boundaries.

Keeping the positive and negative classes separated in an evaluation can lead to misleading results and can make a model appear very different in terms of performance, depending on the presentation. The most important question that one should ask before performing the evaluation is: Which evaluation and metrics provide the most accurate representation regarding how the model will perform in the real world? This needs to be an overarching picture including both classes, and a set of diverse and well-suited metrics.

7/10

290

Class dependent performance

Study 5[17] contains a confusion matrix, enabling us to calculate most metrics for the reported results. As shown in Table 1, reported accuracy is 0.88, the specificity 0.95, the sensitivity 0.80, and the PPV 0.93. The PPV indicates the precision for the positive class and should be accompanied with the corresponding metric for the negative class, i.e., the NPV, which we calculate to be 0.84. This could indicate that the model is better at classifying positive than negative samples correctly, which would be surprising, given that the model is trained on slightly more negative than positive class images. However, not knowing the loss function or which measure the model was optimized for, we cannot investigate this further. What we do know is that a large number of FNs directly cause the low NPV: On the test data set, the model has a significantly higher number of FNs, meaning missed detection, than FPs, meaning a false alarm and in this case over-detection of cancer. The study specifically states reducing misdetection to be the primary motivation for using ML assisted diagnosis, and should thus have reported metrics providing a more comprehensive representation of the model's performance in this regard. For instance, the MCC, which measures the correlation between the true and predicted classes, and is high only if the prediction is good on the positive *and* the negative class. By using the reported results, we calculate the MCC to be 0.76, which is still an acceptable performance, although not as high as the metrics reported by[17]. Which metric values are acceptable depends on non-technical aspects, e.g., the human performance baseline or requirements from hospitals or health authorities.

A potential weakness associated with the NPV is its dependence on TNs, which can overwhelm a classifier whose purpose is detecting rare events. In such cases, the TS, which does not take TNs into account, can be advantageous. From the values reported in Table 1 of[17], the TS value is 0.76, again indicating that the model performs sub-optimally with respect to the objective, despite achieving high accuracy and precision values. In conclusion, the reported metrics show that the model, for the most part, performs well on the evaluation data set. When taking the recalculated metrics into account, we see that the model is more prone to misdetection than causing false alarms.

MediMetrics

Together with this study, we release a web-based tool called *MediMetrics* for calculating the metrics introduced in Table , to make them easily accessible for medical doctors and ML researchers alike. From the provided input, the tool automatically calculates all possible metrics and generates useful visualizations and comparisons the user may freely use in their research. The tool is open-source, and the code available on GitHub[2].

Discussion

There are many available metrics that can be used to evaluate binary classification models. Using only a subset could give a false impression of a model's actual performance, and in turn, yield unexpected results when deployed to a clinical setting. It is therefore important to use a combination of multiple metrics and interpret the performance holistically. Calculating multiple metrics does not require extra work in terms of study design or time, thus there is no apparent reason not to include a set of metrics, besides lack of space, obfuscating actual performance, or lack of knowledge regarding classifier evaluation. Besides interpreting the different metrics together, metrics for the separate classes should be calculated individually. Special care should be taken in cases of imbalanced classes, and the robustness of the classifier's performance tested over a range of class admixtures. In general, a high score in any metric should be regarded with suspicion.

Training and evaluation sets should be strictly separated: Optimally, the data should be split into training, validation, and test data sets. The test data set should be separate from the other partitions to avoid introducing bias on the parameters set during the tuning phase. Furthermore, data regarding the same instance should not be shared across data splits. For example, frames of the same polyp from different angels should not shared across the training and test data sets. Once the model's performance has been optimized on the training data, including tests on a validation set, it can be finally evaluated using the test set. This last step should thus not involve additional tuning, and the test data should not be made available to the analysis before results are fixated for publication. We argue strongly that this should be the standard for studies on the performance of ML classifiers used in medicine in the future. If possible, cross-dataset testing should be performed, meaning in this context that the training and test data are obtained from different hospitals or at least different patients.

In general, all studies involving classification should report the obtained TP, FP, TN, and FN values for validation and test data. In addition, the data along with either the source code, the final models or both should be made available. If this is not possible, other alternatives, like performing additional evaluation on public data sets, such as Kvasir[18] or the Sun database[19]), should be considered. If such an alternative is chosen, it is important to check if the test data is outside the distribution of the training data[20], and in that case, re-fit the model's parameters. Although public data sets do not match the purpose of the study, evaluating the model on such data, by either re-training it or applying it directly on the data if similar to the initial data used for training, would allow others to compare methods and results.

---

[2]github.com/simula/medimetrics

Appendix A. Published Articles

Author contributions

Steven A. Hicks, Inga Strümke, Vajira Thambawita, Michael A. Riegler, Pål Halvorsen, Malek Hammou and Sravanthi Parasa
conceived the experiment(s).
Steven A. Hicks, Inga Strümke, Vajira Thambawita and Michael A. Riegler conducted the experiment(s).
Steven A. Hicks, Inga Strümke, Vajira Thambawita, Michael A. Riegler and Pål Halvorsen analyzed the results.
All authors reviewed the manuscript.

Competing Interests Statement

Steven A. Hicks: Nothing to disclose
Inga Strumke: Nothing to disclose
Vajira Thambawita: Nothing to disclose
Malek Hammou: Nothing to disclose
Michael A. Riegler: Nothing to disclose
Pal Halvorsen: Board member of Augere Medical
Sravanthi Parasa: Consultant Covidien LP; Medical advisory board of Fujifilms

References

1. Nagendran, M. *et al.* Artificial intelligence versus clinicians: systematic review of design, reporting standards, and claims of deep learning studies. *bmj* **368**, 10.1136/bmj.m689 (2020).

2. Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nat. medicine* **25**, 44–56, 10.1038/s41591-018-0300-7 (2019).

3. Schmitz, R. *et al.* Artificial intelligence in gi endoscopy: stumbling blocks, gold standards and the role of endoscopy societies. *Gut* 10.1136/gutjnl-2020-323115 (2021).

4. Hoogenboom, S. A., Bagci, U. & Wallace, M. B. Ai in gastroenterology. the current state of play and the potential. how will it affect our practice and when? *Tech. Gastrointest. Endosc.* 150634, 10.1016/j.tgie.2019.150634 (2019).

5. Ahmad, O. F. *et al.* Establishing key research questions for the implementation of artificial intelligence in colonoscopy-a modified delphi method. *Endoscopy* 10.1055/a-1306-7590 (2020).

6. Cramer, H. *Mathematical methods of statistics* (Princeton University Press Princeton, 1946).

7. Matthews, B. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophys. Acta (BBA) - Protein Struct.* **405**, 442 – 451, 10.1016/0005-2795(75)90109-9 (1975).

8. Boughorbel, S., Jarray, F. & El-Anbari, M. Optimal classifier for imbalanced data using matthews correlation coefficient metric. *PLOS ONE* **12**, 1–17, 10.1371/journal.pone.0177678 (2017).

9. Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* **10**, e0118432, 10.1371/journal.pone.0118432 (2015).

10. Polit, D. F. Blinding during the analysis of research data. *Int. J. Nurs. Stud.* **48**, 636 – 641, 10.1016/j.ijnurstu.2011.02.010 (2011).

11. Le Berre, C. *et al.* Application of artificial intelligence to gastroenterology and hepatology. *Gastroenterology* **158**, 76–94, 10.1053/j.gastro.2019.08.058 (2020).

12. Hassan, C. *et al.* New artificial intelligence system: first validation study versus experienced endoscopists for colorectal polyp detection. *Gut* **69**, 799–800, 10.1136/gutjnl-2019-319914 (2020).

13. Mossotto, E. *et al.* Classification of paediatric inflammatory bowel disease using machine learning. *Sci. reports* **7**, 1–10, 10.1038/s41598-017-02606-2 (2017).

14. Byrne, M. F. *et al.* Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model. *Gut* **68**, 94–100, 10.1136/gutjnl-2017-314547 (2019).

15. Wang, P. *et al.* Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. *Nat. biomedical engineering* **2**, 741–748, 10.1038/s41551-018-0301-3 (2018).

16. Bernal, J. *et al.* WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Comput. Med. Imaging Graph.* **43**, 99–111, 10.1016/j.compmedimag.2015.02.007 (2015).

17. Sakai, Y. *et al.* Automatic detection of early gastric cancer in endoscopic images using a transferring convolutional neural network. *Annu. Int. Conf. IEEE Eng. Medicine Biol. Soc. IEEE Eng. Medicine Biol. Soc. Annu. Int. Conf.* **2018**, 4138–4141, 10.1109/EMBC.2018.8513274 (2018).

18. Borgli, H. *et al.* HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. *Sci. Data* **7**, 283, 10.1038/s41597-020-00622-y (2020).

19. Misawa, M. *et al.* Development of a computer-aided detection system for colonoscopy and a publicly accessible large colonoscopy video database (with video). *Gastrointest. Endosc.* 10.1016/j.gie.2020.07.060 (2020).

20. Robey, A., Hassani, H. & Pappas, G. J. Model-based robust deep learning: Generalizing to natural, out-of-distribution data (2020).

10/10

293

A.18 Paper XVIII - DivergentNets: Medical Image Segmentation by Network Ensemble

Authors: Vajira Thambawita, Steven A. Hicks, Pål Halvorsen, Michael A. Riegler

Abstract: Detection of colon polyps has become a trending topic in the intersecting fields of machine learningand gastrointestinal endoscopy. The focus has mainly been on per-frame classification. More recently,polyp segmentation has gained attention in the medical community. Segmentation has the advantageof being more accurate than per-frame classification or object detection as it can show the affected areain greater detail. For our contribution to the EndoCV 2021 segmentation challenge, we propose twoseparate approaches. First, a segmentation model namedTriUNet-composed of three separate UNetmodels. Second, we combine TriUNet with an ensemble of well-known segmentation models, namelyUNet++, FPN, DeepLabv3, and DeepLabv3+, into a model calledDivergentNetsto produce more generalizable medical image segmentation masks. In addition, we propose a modified Dice loss that calculatesloss only for a single class when performing multi-class segmentation, forcing the model to focus onwhat is most important. Overall, the proposed methods achieved the best average scores for each re-spective round in the challenge, with TriUNet being the winning model in Round I and DivergentNetsbeing the winning model in Round II of the segmentation generalization challenge at EndoCV 2021. Theimplementation of our approach is made publicly available on GitHub.

Published: In proceedings of EndoCV 2021.

Candidate contributions: Vajira contributed to the conception and design of this study. He introduced two new deep neural network architectures named as TriUNet and DivergentNets to perform the segmentation task of EndoCV grand challenge 2021. Vajira contributed to developing these two architectures and performed the experiments. He collected results from the two networks and submitted them to the cloud platform of the challenge. According to his submissions, his team won first place in the polyp segmentation task. Vajira contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective I, Sub-objective III

DivergentNets: Medical Image Segmentation by Network Ensemble

Vajira Thambawita[a,b], Steven A. Hicks[a,b], Pål Halvorsen[a,b] and Michael A. Riegler[a]

[a]*SimulaMet, Pilestredet 52, 0167 Oslo, Norway*
[b]*OsloMet, Pilestredet 46, 0167 Oslo, Norway*

Abstract

Detection of colon polyps has become a trending topic in the intersecting fields of machine learning and gastrointestinal endoscopy. The focus has mainly been on per-frame classification. More recently, polyp segmentation has gained attention in the medical community. Segmentation has the advantage of being more accurate than per-frame classification or object detection as it can show the affected area in greater detail. For our contribution to the EndoCV 2021 segmentation challenge, we propose two separate approaches. First, a segmentation model named *TriUNet* composed of three separate UNet models. Second, we combine TriUNet with an ensemble of well-known segmentation models, namely UNet++, FPN, DeepLabv3, and DeepLabv3+, into a model called *DivergentNets* to produce more generalizable medical image segmentation masks. In addition, we propose a modified Dice loss that calculates loss only for a single class when performing multi-class segmentation, forcing the model to focus on what is most important. Overall, the proposed methods achieved the best average scores for each respective round in the challenge, with TriUNet being the winning model in Round I and DivergentNets being the winning model in Round II of the segmentation generalization challenge at EndoCV 2021. The implementation of our approach is made publicly available on GitHub.

Keywords

Deep learning, medical image segmentation, colonoscopy, generalisation, computer-assisted diagnosis

1. Introduction

Automatic segmentation of medical images is a common use case in machine learning that has gained a lot of attention over the last few years. Popular applications include segmenting tumors in computed tomography (CT) scans [1, 2], finding abnormalities in magnetic resonance images (MRIs) [3, 4], or segmenting organs and tissue in medical applications [5, 6]. Segmentation goes a step beyond standard classification and object detection as it extracts the area in an image that corresponds to the target class or classes at pixel-level precision. This comes with two advantages that are important in the medical field. The first one is that the algorithm learns pixel-wise and has more examples to learn from compared to if it would learn image-wise [7, 8]. This can help for use cases where one does not have many images from a disease. Secondly, the segmented area makes it easier for the physician to determine what the algorithm
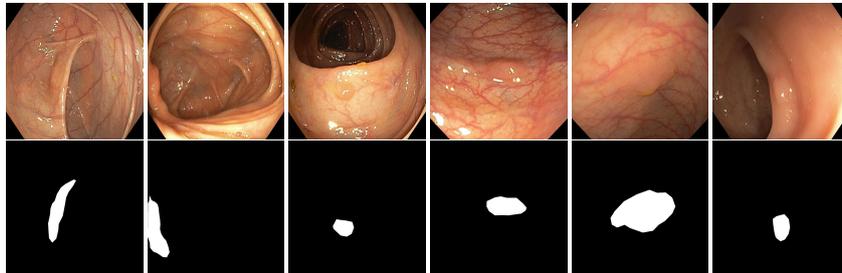
Figure 1: Some example images and their corresponding ground truth masks taken from the development dataset of EndCV2021 [10]

detected and classified as a disease which serves in a broader sense as an explanation. Thus, detailed image segmentation can also be seen as a type of explanation method. This makes it highly desired by medical professionals as explainable machine learning is seen as one of the requirements for the successful implementation of automatic decision support systems in hospitals [9]. As part of the EndoCV2021 challenge (https://endocv2021.grand-challenge.org/), we were tasked with creating machine learning models that automatically segment polyps [11, 12, 13] in video frames collected from real-world endoscopies. This is a complex task as polyps come in various shapes and sizes, where some (e.g., flat lesions) are barely detectable by even the most experienced endoscopists. Figure 1 shows some of the more difficult examples taken from EndoCV's development dataset [10] provided by the challenge organizers. The challenge presented two separate tasks, the *detection generalization challenge* and the *segmentation generalization challenge*. We participated in the segmentation generalization challenge, where we achieved the best results among 13 other competitors in both rounds. The code for the experiments presented in this paper is available on GitHub[1].

This paper summarizes our approaches to the EndoCV2021 challenge. In particular, we developed the *TriUNet* segmentation model combining three separate UNet models, and the *DivergentNets* that combines TriUNet with an ensemble of the well-known segmentation models, namely UNet++, FPN, DeepLabv3, and DeepLabv3+. The rest of the paper is structured as follows. Section 2 present our approach to this year's challenge, where we use two unique models that achieve state-of-the-art performance on the EndoCV dataset. Section 3 gives a description of the implementation details on the models and training procedure and how the data was split and prepared. Section 4 presents the preliminary and official results for our tested models and performs a qualitative analysis on some of the predicted masks. Lastly, Section 5 concludes this paper with a summary and plans for future work.

---

[1]https://github.com/vlbthambawita/divergent-nets
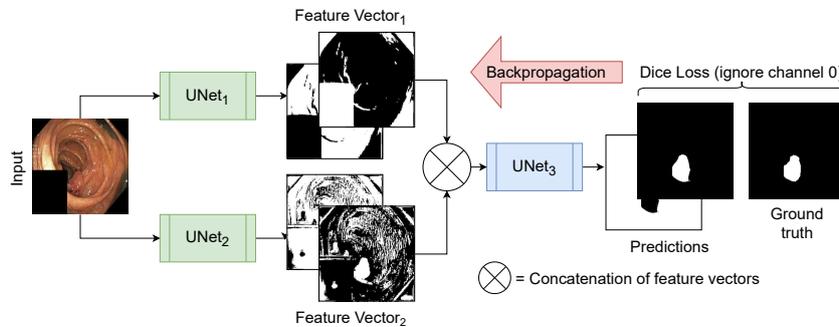
Figure 2: An illustration of the TriUNet architecture. First, the image is passed through two separate UNets in parallel, which produce the feature vectors V_1 and V_2, respectively. These two vectors are then concatenated before being passed through a third UNet that predicts the final segmentation mask. The loss is calculated by taking the Dice coefficient of the mask corresponding to the main class and the ground truth, which is then back-propagated through the entire model.

2. Approach

In this section, we introduce three approaches that we developed for the segmentation generalization challenge at EndoCV 2021, which are two new architectures, TriUNet and DivergentNet, and a modified loss function.

2.1. TriUNet

TriUNet is a convolutional neural network (CNN) architecture that utilizes multiple UNet [14] architectures arranged in a triangular structure as depicted in Figure 2. The model takes a single image as input, which is passed through two separate UNet models with different randomized weights. The output of both models is then concatenated before being passed through a third UNet model to predict the final segmentation mask. Figure 2 also shows an example of the intermediate representations provided by the two initial UNet models. The loss is calculated and back-propagated through the whole model, meaning the entire network is trained in one go. From the intermediate representations, we clearly see that the different UNets learn different interpretations of the data, which then are combined in one final output.

2.2. DivergentNets

The DivergentNets network is inspired by the idea of ensembles made with multiple high-performing image segmentation architectures and the TriUNet architecture presented in the previous section. We constructed this DivergentNets assuming that cumulative decisions taken from multiple intermediate models should give a more precise decision than the predictions from a single network. The included models were selected based on what has previously been shown to produce good results on different segmentation tasks and some preliminary experiments
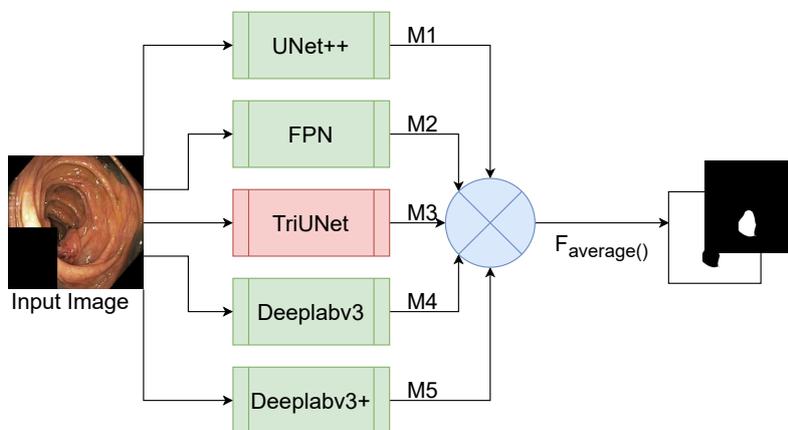
Figure 3: An illustration of the DivergentNets architecture. First, five different models are trained using the U-Net++, FPN, TriUnet, DeepLabv3, and DeepLabv3 architectures. Then, an image is passed through each model separately, which produces masks M_1 to M_5. Last, the masks are averaged to make the final segmentation mask.

using each model independently. Furthermore, the selection was limited by the hardware we had available.

As shown in Figure 3, our configuration comprises five intermediate models, namely UNet++, FPN, DeepLabv3, DeepLabv3+, and TriUNet. The five intermediate models are first trained for N number of epochs separately, where the best checkpoint of each model is selected to be combined in DivergentNet. This N should be selected using a preliminary experiment. In our case, we identified that $N = 200$ is enough to produce high-quality masks. However, training for more epochs may result in better checkpoints to use in DivergentNet. To produce the intermediate masks, the output of each model is passed through a *softmax2d* activation function. However, this should be changed based on the application. In our case, we predict masks for two classes, background and polyp, where no two categories may overlap. The masks produced by each intermediate model represent the divergent views on the data. The final output of DivergentNets is made by averaging the pixels between each intermediate mask and rounding to the nearest integer (either 0 or 1).

2.3. Single-channel Dice

All models were trained to predict masks for both polyps and background (mostly containing the mucosal wall lining the inside of the colon). As the primary focus of EndoCV is to segment colon polyps, we use a modified Dice loss to calculate the prediction error. We call this loss function *single-channel Dice loss* as it only considers one channel when calculating error. This

Table 1

An overview of how the data was split between training, validation, and testing.

Dataset	Partition	# Samples	# Polyp	# Non-Polyp
EndoCV	Training	1,754	1,329	435
EndoCV	Validation	2,756	1,400	1,356
HyperKvasir	Testing	1,000	1,000	0

is shown in Equation 1:

$$\text{Single-Channel Dice Loss} = \frac{2 \cdot |A_n \cap B_n|}{2 \cdot |A_n \cap B_n| + |B_n \backslash A_n| + |A_n \backslash B_n|} \tag{1}$$

where n represents the class for which loss should be calculated for. In this case, we only calculate loss for the polyp class and ignore the background.

3. Experiments

The experimenters can be categorized into two sub-groups, namely baseline experiments and experiments used for the challenge. The baseline experiments were used to benchmark common segmentation models. The baseline models tested were UNet [14], UNet++ [15], FPN [16], DeepLabv3 [17], and DeepLabv3+ [18]. In turn, we used these networks to design the TriUNet and DivergentNets architectures. This section describes the experimental setup, including how the data was prepared, training procedures, architecture implementations, and specifics on what hyperparameters were used.

3.1. Data details and preparation

The development dataset provided by the organizers was split between several directories, primarily one part consisting of a five-way center-wise split (directories *C1* through *C5*) containing a diverse set of data [10], and one part consisting of pure sequence data (directories *seq1* through *seq15*). For this challenge, we decided to use a standard three-way split of the data into training, validation, and testing datasets. The training data was made up of all the data contained within the center-wise split for training data, in addition to a few sequences only containing negative samples. For validation, we used the remaining sequence data. Table 1 gives an overview of how each directory was split between training, validation, and test datasets. All samples contained an image, a segmentation mask, bounding box coordinates, and the image with the bounding-box superimposed over it. As we were only participating in the segmentation generalization challenge, we only used the images and segmentation masks.

3.2. Implementation details

All models were implemented in PyTorch and trained on an Nvidia DGX-2. The Nvidia DGX-2 consists of 16 Tesla V100 GPUs, dual Intel Xeon Platinum 816 processors, and 1.5 terabytes of system memory. Despite that the system contains 16 GPUs, we only use one for training

Table 2
The results collected from the preliminary experiments on the internal validation dataset.

Model	All Classes				Polyp Class				Background Class			
	IoU	F1	REC	PREC	IoU	F1	REC	PREC	IoU	F1	REC	PREC
U-Net	0.973	0.985	0.985	0.985	0.774	0.802	0.831	0.926	0.984	0.991	0.996	0.988
U-Net++	0.972	0.984	0.984	0.984	0.787	0.815	0.847	0.918	0.983	0.991	0.995	0.989
FPN	0.973	0.985	0.985	0.985	0.778	0.810	**0.853**	0.904	0.984	0.991	0.995	0.989
DeepLabv3	0.971	0.984	0.984	0.984	0.764	0.798	0.842	0.902	0.983	0.991	0.994	0.989
DeepLabv3+	0.973	0.985	0.985	0.985	0.777	0.807	0.840	0.919	0.984	0.991	0.994	0.989
TriUNet	0.970	0.983	0.983	0.983	0.775	0.802	0.846	0.903	0.982	0.990	0.992	**0.989**
DivergentNets	**0.976**	**0.986**	**0.986**	**0.986**	**0.795**	**0.823**	0.844	**0.937**	**0.986**	**0.992**	0.997	0.989

so that we can train multiple models in parallel. For the baseline experiments, we used the implementations and pre-trained weights available in the *Segmentation Models* [19] library. These networks were also used as the basis for our proposed TriUNet and DivergentNets. Each model was implemented SE-ResNeXt-50-32x4D [20] as the encoder, which was initialized with ImageNet [21] weights. Images and masks were resized to 256×256 and resized back to the original resolution using bilinear interpolation. The final prediction was produced by passing the output through a two-dimensional softmax function. For training, all models started with a learning rate of 0.0001 and reduced to 0.00001 after 50 epochs. The model error was calculated using the proposed single-channel Dice for the polyp class (as explained in Section 2.3), and the weights were optimized using Adam [22].

As the size of the development dataset is relatively small, we use a series of different image augmentations to make the model more generalizable. These augmentations include horizontal flip, shift scale rotation, resizing, additive Gaussian noise, perspective shift, contrast limited adaptive histogram equalization (CLAHE), random brightness, random gamma, random sharpen, random blur, random motion blur, random contrast, and hue saturation. The augmentations were implemented using the Python library *Albumentations* [23]. No augmentations were applied to the validation and testing data.

4. Results and Discussion

In this section, we discuss the preliminary and official results of our approach to the EndoCV 2021 challenge. We also perform a qualitative analysis of the models, showing how the different modes diverge to a final prediction.

4.1. Preliminary results

Table 2 and Table 3 show the initial results on the provided development validation and testing datasets. Overall, we see that all models perform well on segmenting the polyp class, with the DivergentNets architecture achieving the best performance and UNet++ at a close second place on both the validation and test datasets. Comparing UNet and TriUNet, we see that TriUNet performs slightly better on the polyp class, however, UNet++ outperforms both. With these results, it would be natural to assume that a TriUNet++ architecture would perform even better

Table 3
The results collected from the preliminary experiments on the internal testing dataset.

Model	All Classes				Polyp Class				Background Class			
	IoU	F1	REC	PREC	IoU	F1	REC	PREC	IoU	F1	REC	PREC
U-Net	0.941	0.967	0.967	0.967	0.823	0.883	0.876	0.938	0.959	0.977	0.988	0.970
U-Net++	0.945	0.969	0.969	0.969	0.834	0.894	0.882	0.942	0.961	0.979	0.988	0.972
FPN	0.944	0.968	0.968	0.968	0.824	0.887	0.870	0.943	0.961	0.978	**0.990**	0.970
DeepLabv3	0.942	0.968	0.968	0.968	0.821	0.885	0.874	0.935	0.959	0.977	0.988	0.970
DeepLabv3+	0.942	0.968	0.968	0.968	0.823	0.886	0.883	0.931	0.823	0.886	0.883	0.931
TriUNet	0.941	0.967	0.967	0.967	0.829	0.890	**0.891**	0.928	0.959	0.977	0.983	**0.975**
DivergentNets	**0.949**	**0.972**	**0.972**	**0.972**	**0.840**	**0.899**	0.886	**0.946**	**0.964**	**0.980**	0.990	0.973

Table 4
The official results provided by the EndoCV organizers. *Score* is an average score of F1-score, F2-score, PPV, and Recall provided by the organizers, and *SD* is the standard deviation of the metrics.

Round	Model	Score	SD
I	UNet++	0.917	0.168
	TriUNet	*0.925*	*0.152*
II	TriUNet	0.796	0.047
	DivergentNets	*0.823*	*0.043*

than TriUNet. However, due to hardware limitations (specifically GPU memory), we were unable to test this configuration and move this to future work.

4.2. Official results

The official evaluation was split into two rounds, where *Round I* used a subset of the testing data that was fully used for *Round II*. For both rounds, we were limited by the number of submissions that could be delivered per day. This limit started at five-per-day for *Round I* and was reduced to two-per-day for *Round II*. Due to this limitation, only a subset of the aforementioned models was submitted as official runs. Models were selected based on their performance on a different test dataset that we chose, namely the well-known and established HyperKvasir [24] dataset, in ascending fashion. From the HyperKvasir dataset, we only used the images with segmentation masks as an independent test set to determine the best generalizable model. It was not used in any way as training or validation data. Table 4 shows the official results for *Round I* and *Round II*. Note that the DivergentNets model was not part of *Round I* as it was developed during *Round I* and used in *Round II* once it was finished. From the results, we see that TriUNet achieved the best score for *Round I*, and DivergentNets achieved the best score for *Round II, i.e.,*both winning their respective rounds of the competition.

4.3. Qualitative analysis

Figure 4 shows some example masks predicted by our best performing model (DivergentNets) together with masks produced by the intermediate models. We see that each intermediate model learns slightly different features, making an overall more precise segmentation mask when

302
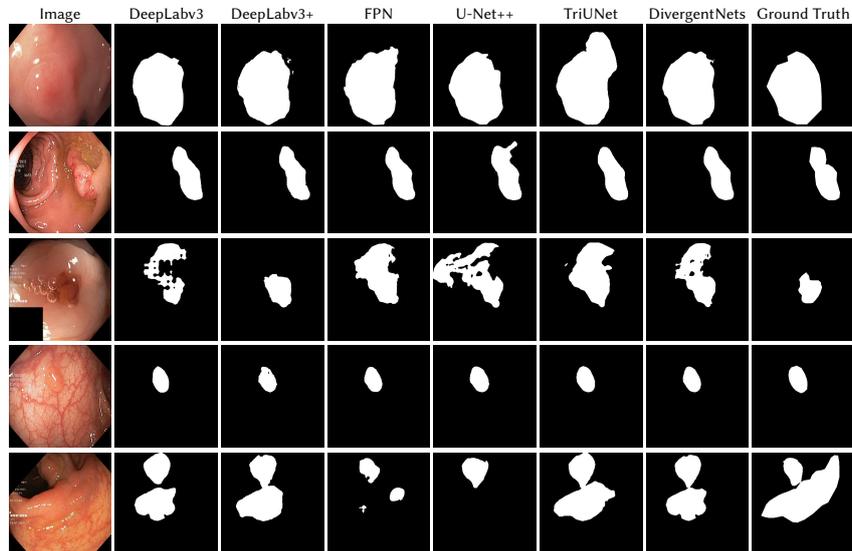
Figure 4: Some predicted mask examples taken from the divergent network and its five intermediate models. The images are taken from HyperKvasir.

combined. For example, the first row of Figure 4 shows the predicted masks and ground truth of a large polyp. We see that each model predicts slightly different masks for the same input and that TriUNet over-estimates the size of the polyp. After averaging the predicted masks for DivergentNets' final output, this area is smoothed out by the predictions from the other intermediate models.

Even though DivergentNets primarily produces more accurate masks than any single model, there are cases where masks from the intermediate model better match the ground truth. We see this in row three, where DeepLabv3+ produces a more precise mask than all other intermediate models, making the averaged output less accurate.

5. Conclusion and future work

In this paper, we presented our approaches to the EndoCV 2021 challenge. We trained a series of baseline models and two models based on novel architectures using a slightly modified Dice loss, which achieved the overall best score in both rounds of the generalization segmentation challenge. For the first round, we developed TriUNet, which reached an average score of 0.925 on the official testing dataset. For the second round, we developed the DivergentNets architecture, which combines the baseline models with the TriUNet to gain an average score of 0.823 on the official training dataset. Due to a limitation on time and computational resources,

we could not experiment with another improved version where the UNet architectures are replaced with UNet++ architectures.

For future work, we plan to explore different configurations of TriUNet, such as implementing TriUNet++ and testing different architectures for three architectures that make up the TriUNet architecture, for example, combining the UNet, FPN, and DeepLabv3 as TriUNet nodes. We would also like to explore different configurations for the DivergentNets architecture with different networks for each node. Another idea could be to use a neural network to produce the final prediction instead of the current averaging technique, similar to the approach discussed in [25]. Further testing the approaches with datasets from other medical fields can help to identify the generalizability of our approach.

Acknowledgments

References

[1] O. I. Alirr, Deep learning and level set approach for liver and tumor segmentation from ct scans, Journal of Applied Clinical Medical Physics 21 (2020) 200–209.

[2] S. Pang, A. Du, M. A. Orgun, Z. Yu, Y. Wang, Y. Wang, G. Liu, Ctumorgan: a unified framework for automatic computed tomography tumor segmentation, European Journal of Nuclear Medicine and Molecular Imaging 47 (2020) 2248–2268. doi:10.1007/s00259-020-04781-3.

[3] N. Yamanakkanavar, J. Y. Choi, B. Lee, Mri segmentation and classification of human brain using deep learning for diagnosis of alzheimer's disease: a survey, Sensors 20 (2020). doi:10.3390/s20113243.

[4] L. Clarke, R. Velthuizen, M. Camacho, J. Heine, M. Vaidyanathan, L. Hall, R. Thatcher, M. Silbiger, Mri segmentation: Methods and applications, Magnetic Resonance Imaging 13 (1995) 343–368.

[5] P. M. Scheikl, S. Laschewski, A. Kisilenko, T. Davitashvili, B. Müller, M. Capek, B. P. Müller-Stich, M. Wagner, F. Mathis-Ullrich, Deep learning for semantic segmentation of organs and tissues in laparoscopic surgery, Current Directions in Biomedical Engineering 6 (2020). doi:10.1515/cdbme-2020-0016.

[6] O. Schoppe, C. Pan, J. Coronel, H. Mai, Z. Rong, M. I. Todorov, A. Müskes, F. Navarro, H. Li, A. Ertürk, B. H. Menze, Deep learning-enabled multi-organ segmentation in whole-body mouse scans, Nature Communications 11 (2020) 5626. doi:10.1038/s41467-020-19449-7.

[7] T. Sun, W. Zhang, Z. Wang, L. Ma, Z. Jie, Image-level to pixel-wise labeling: From theory to practice, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2018, pp. 928–934. doi:10.24963/ijcai.2018/129.

Appendix A. Published Articles

[8] A. Bansal, X. Chen, B. Russell, A. Gupta, D. Ramanan, Pixelnet: Representation of the pixels, by the pixels, and for the pixels, arXiv preprint arXiv:1702.06506 (2017).

[9] J. Amann, A. Blasimme, E. Vayena, D. Frey, V. I. Madai, Explainability for artificial intelligence in healthcare: a multidisciplinary perspective, BMC Medical Informatics and Decision Making 20 (2020) 1–9.

[10] S. Ali, D. Jha, N. Ghatwary, S. Realdon, R. Cannizzaro, M. A. Riegler, P. Halvorsen, C. Daul, J. Rittscher, O. E. Salem, D. Lamarque, T. de Lange, J. E. East, Polypgen: A multi-center polyp detection and segmentation dataset for generalisability assessment, arXiv (2021).

[11] S. Ali, M. Dmitrieva, N. Ghatwary, S. Bano, G. Polat, A. Temizel, A. Krenzer, A. Hekalo, Y. B. Guo, B. Matuszewski, M. Gridach, I. Voiculescu, V. Yoganand, A. Chavan, A. Raj, N. T. Nguyen, D. Q. Tran, L. D. Huynh, N. Boutry, S. Rezvy, H. Chen, Y. H. Choi, A. Subramanian, V. Balasubramanian, X. W. Gao, H. Hu, Y. Liao, D. Stoyanov, C. Daul, S. Realdon, R. Cannizzaro, D. Lamarque, T. Tran-Nguyen, A. Bailey, B. Braden, J. E. East, J. Rittscher, Deep learning for detection and segmentation of artefact and disease instances in gastrointestinal endoscopy, Medical Image Analysis 70 (2021) 102002. doi:10.1016/j.media.2021.102002.

[12] S. Ali, F. Zhou, B. Braden, A. Bailey, S. Yang, G. Cheng, P. Zhang, X. Li, M. Kayser, R. D. Soberanis-Mukul, S. Albarqouni, X. Wang, C. Wang, S. Watanabe, I. Oksuz, Q. Ning, S. Yang, M. A. Khan, X. W. Gao, S. Realdon, M. Loshchenov, J. A. Schnabel, J. E. East, G. Wagnieres, V. B. Loschenov, E. Grisan, C. Daul, W. Blondel, J. Rittscher, An objective comparison of detection and segmentation algorithms for artefacts in clinical endoscopy, Scientific Reports 10 (2020) 2748. doi:10.1038/s41598-020-59413-5.

[13] D. Jha, S. Ali, N. K. Tomar, H. D. Johansen, D. Johansen, J. Rittscher, M. A. Riegler, P. Halvorsen, Real-time polyp detection, localization and segmentation in colonoscopy using deep learning, IEEE Access 9 (2021) 40496–40510.

[14] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer International Publishing, Cham, 2015, pp. 234–241.

[15] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, in: Deep learning in medical image analysis and multimodal learning for clinical decision support, Springer, 2018, pp. 3–11. doi:10.1007/978-3-030-00889-5_1.

[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2017, pp. 2117–2125.

[17] L.-C. Chen, G. Papandreou, F. Schroff, H. Adam, Rethinking atrous convolution for semantic image segmentation, arXiv preprint arXiv:1706.05587 (2017).

[18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.

[19] P. Yakubovskiy, Segmentation models pytorch, 2020.

[20] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), 2018, pp. 7132–7141.

305

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, International Journal of Computer Vision (IJCV) 115 (2015) 211–252. doi:10.1007/s11263-015-0816-y.

[22] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[23] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: fast and flexible image augmentations, Information 11 (2020) 125.

[24] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen, et al., Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy, Scientific Data 7 (2020) 1–14. doi:10.1038/s41597-020-00622-y.

[25] V. Thambawita, D. Jha, H. L. Hammer, H. D. Johansen, D. Johansen, P. Halvorsen, M. A. Riegler, An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification, ACM Trans. Comput. Healthcare 1 (2020). doi:10.1145/3386295.

A.19 Paper XIX - A Self-learning Teacher-student Framework for Gastrointestinal Image Classification

Authors: Henrik L. Gjestang, Steven A. Hicks, **Vajira Thambawita**, Pål Halvorsen, Michael A. Riegler

Abstract: We present a semi-supervised teacher-student framework to improve classification performance on gastrointestinal image data. As labeled data is scarce in medical settings, this framework is built specifically to take advantage of vast amounts of unlabeled data. It consists of three main steps: (1) train a teacher model with labeled data, (2) use the teacher model to infer pseudo labels with unlabeled data, and (3) train a new and larger student model with a combination of labeled images and inferred pseudo labels. These three steps are repeated several times by treating the student as a teacher to relabel the unlabeled data and consequently train a new student. We demonstrate that our framework can classify both video capsule endoscopy (VCE) and standard endoscopy images. Our results indicate that our teacher-student framework can significantly increase the performance compared to traditional supervised-learning-based models, i.e., an overall increase in the F_1-score of 4.7% for the Kvasir-Capsule VCE dataset and 3.2% for the HyperKvasir colonoscopy dataset. We believe that our framework can use more of the data collected at hospitals without the need for expert labels, contributing to overall better models for medical multimedia systems for automatic disease detection.

Candidate contributions: Vajira contributed to the conception and designing of the study presented in this paper. He contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective I, Sub-objective III

A self-learning teacher-student framework for gastrointestinal image classification

Henrik L. Gjestang*, Steven A. Hicks*†, Vajira Thambawita*†, Pål Halvorsen*†, and Michael A. Riegler*‡

*SimulaMet, Norway   †Oslo Metropolitan University, Norway   ‡UIT The Arctic University of Norway

Abstract—We present a semi-supervised teacher-student framework to improve classification performance on gastrointestinal image data. As labeled data is scarce in medical settings, this framework is built specifically to take advantage of vast amounts of unlabeled data. It consists of three main steps: (1) train a teacher model with labeled data, (2) use the teacher model to infer pseudo labels with unlabeled data, and (3) train a new and larger student model with a combination of labeled images and inferred pseudo labels. These three steps are repeated several times by treating the student as a teacher to relabel the unlabeled data and consequently train a new student. We demonstrate that our framework can classify both video capsule endoscopy (VCE) and standard endoscopy images. Our results indicate that our teacher-student framework can significantly increase the performance compared to traditional supervised-learning-based models, i.e., an overall increase in the F_1-score of 4.7% for the Kvasir-Capsule VCE dataset and 3.2% for the HyperKvasir colonoscopy dataset. We believe that our framework can use more of the data collected at hospitals without the need for expert labels, contributing to overall better models for medical multimedia systems for automatic disease detection.

Index Terms—Teacher-student framework, capsule endoscopy, colonoscopy, self-training, deep learning, machine learning, computer vision

I. INTRODUCTION

Numerous abnormal mucosal findings ranging from minor annoyances to highly lethal diseases can be found in the human gastrointestinal (GI) tract. According to the International Agency for Research on Cancer [1], GI cancer globally accounts for about 3.5 million new cases each year. These types of cancer usually have combined mortality of about 63% and 2.2 million deaths per year [2], [3], [4]. In this context, endoscopy is currently the gold-standard procedure for examining the GI tract for cancer precursors like polyps, but its effectiveness is considerably limited by the variation in operator performance [5], [6], [7]. The consequence is an average of 20% polyp miss-rate in regular colonoscopies [8]. In video capsule endoscopy (VCE) analysis, essential findings are also missed due to lack of concentration, insufficient experience and knowledge [9], [10], [11]. Thus, improved endoscopic performances, high-quality clinical examinations, and systematic screening are significant factors in preventing GI disease-related morbidity and deaths.

To assist clinicians, computer-aided diagnosis (CAD) systems have recently received a lot of attention where supervised machine learning models detect and classify lesions. Despite there being a lot of data gathered at hospitals, labeled data is scarce due to the time-consuming, tedious, and expensive
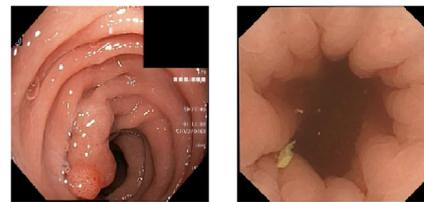
(a) HyperKvasir   (b) Kvasir-Capsule

Fig. 1: Example images from both datasets.

process of having qualified medical personnel do manual labeling work. In this respect, semi-supervised methods using unlabeled data have shown improvements and been successfully applied in medical image analyses [12]. Instead of learning from a large set of annotated data, algorithms learn from sparsely labeled and unlabeled data. Self-learning [13], [14] and neural graph learning [15] are examples of using unlabeled data in addition to a small amount of labeled data to extract additional information [16], [14], [13]. In an area with scarce data, such algorithms might be the technology needed to make AI truly useful for medical applications.

In this work, we present a semi-supervised teacher-student framework using the classification of GI images as a case study. The semi-supervised learning framework trains a model on labeled data, uses this model to predict image labels, called pseudo labels, from a corpus of unlabeled images, then finally trains a new model on the combination of labeled images and pseudo labels. This type of self-learning framework is called a teacher-student framework because we first train a model on the labeled data (the teacher), and then use the teacher to train a student, which eventually becomes better than the teacher. Moreover, we have used two open datasets, HyperKvasir [17] and Kvasir-Capsule [18] to demonstrate the potential of our framework for both colonoscopy and VCE data (example images shown in Figure 1. Our results indicate that our teacher-student model can significantly increase the performance compared to traditional supervised-learning-based models, i.e., compared to EfficientNet [19] as a representative example of supervised models, we observe overall increase in the F_1-score of 4.7% for the Kvasir-Capsule VCE dataset and 3.2% for the HyperKvasir colonoscopy dataset. We believe that this proves, using two different datasets, that our

framework has the potential to be a useful addition to existing medical multimedia systems for automatic disease detection as it can make use of the unlabeled data collected at hospitals.

II. Background and Related Work

In recent years, there have been many proposed methods to use deep learning to produce better and more efficient health care systems [20], [21]. Many of these methods are considered state-of-the-art within the fields of deep learning [22], [23]. As neural networks developed for medicine usually deal with high-risk decision-making, all data should be labeled by a doctor or other professional personnel before it is used for training. We propose a method that takes advantage of unlabeled data, which is more readily available and cheaper to produce.

Examination of the colon using conventional endoscopy or VCE produces a lot of video data. Too much data for any medical professional to manually annotate every frame. The result is that only a handful of frames are labeled, leaving a vast number of unlabeled images. In cases like this, semi-supervised models that can utilize the unlabeled data have found success in the area of medical image analyses [12], where self-learning [13], [14] and neural graph learning [15] are example solutions where additional information is extracted from the unlabeled data [16], [14], [13].

Self-training is one of the most common semi-supervised methods used. Self-training means to use a trained, supervised model, called a teacher, to assign pseudo labels on a subset of the unlabeled data by using a threshold on the predictions. This pseudo-labeled data is then combined with the original labeled dataset, and used to train a new model, called a student model. This is repeated multiple times until the system converges, and thereby fully utilizing the unlabeled data to increase model performance.

Xie et al. [24] proposed a self-training framework better suited to work well at scale, and their model achieved 88.4% top-1 accuracy on ImageNet [25], which is 2.0% better than the previous state of the art model [24]. They found that for self-training to work well at scale, noise should be inserted into the student model during training while no noise should be input into the teacher model when generating pseudo labels. Noisy students improve self-training in two distinct ways: (1) it makes the student model deeper and wider than, or at least equal to, the teacher so the student can learn from a larger dataset, and (2) it adds noise to the student, forcing the student to better generalize on the unlabeled dataset, and thereby learn more. The authors used multiple types of noise to improve the student model's ability to generalize, such as RandAugment data augmentation [26] as input noise, stochastic depth [27] as model noise, and dropout [28]. Injecting noise on the input data has the benefit of enforcing local smoothness in the decision function on both labeled and unlabeled images. The student must be able to correctly classify images with random data augmentations, which helps the student model to learn beyond the teacher and make predictions on more difficult data. When model noise such

as stochastic depth and dropout are used, the teacher behaves like an ensemble while it generates pseudo labels, whereas the student is forced to mimic a more powerful ensemble model. Below is the noisy student algorithm in more detail. In the following algorithm, $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ in steps 1 and 2 are the labeled images and their respective label, and $\{(\tilde{x}_1, \tilde{y}_1), (\tilde{x}_2, \tilde{y}_2)..., (\tilde{x}_m \tilde{y}_m)\}$ in step 3 is the unlabeled images and their respective pseudo label.

1) Learn a teacher model θ_*^t which minimizes the cross-entropy loss on a labeled set of images.

$$\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t)) \qquad (1)$$

2) Use the teacher model to generate soft or hard pseudo labels for unlabeled images.

$$\tilde{y}_i = f(\tilde{x}, \theta_*^t), \forall i = 1, ..., m \qquad (2)$$

3) Learn an equal-or-larger student model θ_*^s which minimizes the cross-entropy loss on labeled images and pseudo labels with noise added to the student model.

$$\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m} \sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s)) \qquad (3)$$

4) Use the student model as a teacher and repeat steps 2 through 4 until the student stops improving.

To train the teacher and student models, Xie et al. [24] use EfficientNet [19] as the baseline model and further scale up and down the model to achieve network architectures which are deeper and wider, as well as networks which are shallower and thinner.

The previously discussed teacher-student-based methods are complex, difficult to train, and hard to set up. In this work, we focus on a simple but efficient framework with a particular emphasis on the evaluation process specifically targeted towards medical applications. Related work usually follows a *set and forget* training strategy, which only produces one final model, while our framework stores every model and its corresponding evaluations to be considered in the entire training process.

III. Iterative Teacher-Student Framework

We developed a semi-supervised teacher-student-based image classification system, depicted in Figure 2, to take advantage of the vast amounts of unlabeled medical data and thereby reduce the estimated cost of creating medical classification models.

The first step in our semi-supervised framework is to train a model, called a teacher model, on labeled data. The next step is to use the trained model to infer pseudo labels from the unlabeled dataset, then train a student model on the combination of the pseudo-labeled data and the original labeled dataset. Finally, we switch the teacher with the student and repeat in an iterative process. This process is shown in Figure 2. This framework is heavily dependent on the initial
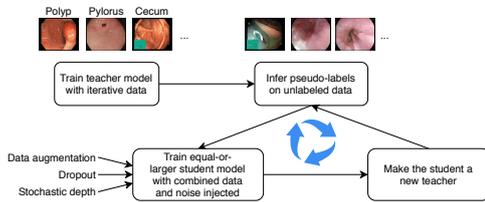
Fig. 2: Illustration of the Noisy student method.

Fig. 3: The pipeline used for training our models. All code is available on GitHub[1].

teacher model, which must capture at least some of the features of each class to later produce useful pseudo labels in later iterations. Therefore, we set up a data pipeline with integrated monitoring of model metrics, as demonstrated in Figure 3.

For our teacher and student models, we use architectures based on EfficientNet [19] with weights pre-trained on the ImageNet dataset [25]. EfficientNet was chosen due to its compound scaling, which uniformly scales network width, depth, and resolution to create the optimal network that captures all fine-grained features of an image. Furthermore, we can also use the architecture itself to add stochastic depth as model noise to the student model.

The labeled images are downscaled to 256×256 pixels and the pixel values are normalized to be between 0 and 1. The framework was tested with the original resolution of the VCE device, 336×336 pixels, but we found no additional benefits that outweighed the increased time needed for training. To produce a uniform number of images from each class, we resample the dataset by oversampling the minority classes and undersampling the majority classes. For the student models, we randomly augment the data using a series of image augmentations (rotation, flipping, skewing, cropping, and adjust image brightness, saturation, and contrast), and add model noise in the form of stochastic depth and dropout. We generate pseudo labels for the unlabeled data by running every image through our predictive model. For the first iteration, the teacher model makes the predictions, and the next time the student. This is the most time-consuming process of our system, and depending on the size of the unlabeled dataset and the depth of the model, it can take from half an hour to many hours for each run through the dataset on an Nvidia Volta 100 GPU. For HyperKvasir, which has approximately $100,000$ unlabeled images, and the shallowest EfficientNet model (EfficientNetB0), the process takes roughly forty minutes. The model predicts a probability distribution over the set of classes for each image in the unlabeled data. We empirically set a threshold to include the image if the prediction confidence is above 90%. If above this threshold, the image is marked with a pseudo label and incorporated into the training data for the next iteration. If confidence is below the threshold, it is assumed to be out-of-domain and rejected.

To prevent growing memory usage, we add a threshold
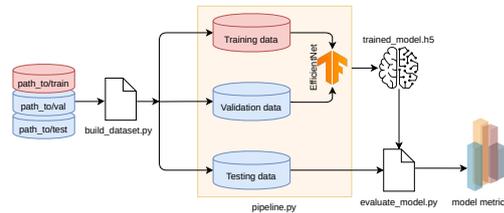
for the max number of pseudo labels added to each class. This threshold was set based on the number of images in the majority class in the initial training data and remained constant throughout all iterations of the framework. In Figure 4, an example of the number of samples per class for two iterations of the teacher-student framework is shown. The majority class in this example is *BBPS-2-3* with 803 samples, and during the run, no more pseudo labels are added.

The framework stores the model metrics for each trained model at the end of the run. In this metric folder, we can inspect the pseudo labels, its distribution of samples per class, accuracy and loss for both training data and validation data, classification report of precision, recall, F_1-score, confusion matrix, and more. When testing the framework, we monitored the pseudo labels added to the training data of the later models to verify that they were aligned with the annotated images for the given class. In Figure 5, we see one such output of pseudo labels for a subset of the classes from the HyperKvasir dataset, which contains the class *hemorrhoids*, one of the minority classes. We see that the teacher models can produce good in-domain pseudo labels for the minority class. The correctness of the pseudo labels was verified by a professional endoscopist.

We rely on data augmentation and model noise to help the student model generalize and not overfit on the training data. This is done when preparing the data by using data transformations, which applies a user-defined function to each element of the input dataset to artificially inflate the training dataset with label-preserving transformations. Because the samples are independent of one another, the process can be run in parallel across multiple CPU cores for efficiency. These transformations include commonly used ones such as horizontal and vertical flipping of the image, skewing, cropping, and rotating. By doing this, the augmented data is able to simulate a variety of subtly different data points, as opposed to just duplicating the same data over and over. We shuffle the data twice to reduce variance and to make sure the model remains general and overfits less.

Our framework incorporates two methods for dampening the negative effect of training on a highly skewed dataset: (1) weighing the classes by the number of samples per class when calculating loss, and (2) sampling from the training dataset during model training, which ensures the model is fed with
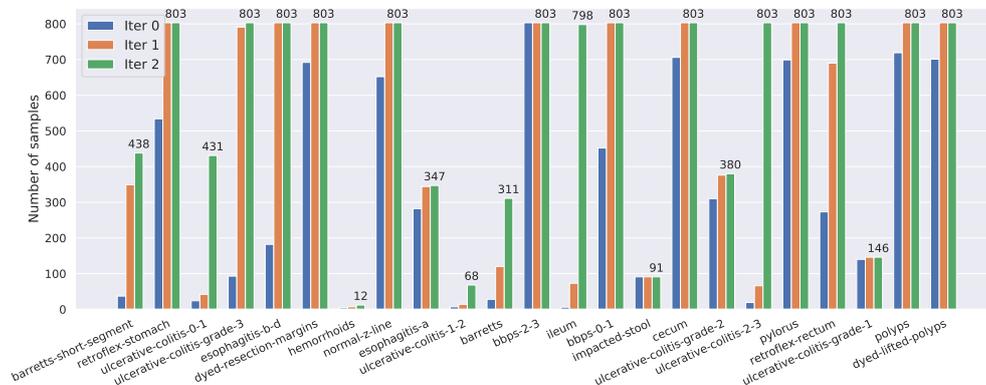
Fig. 4: Class distribution of labeled images and pseudo labels after two iterations of adding pseudo labels from the HyperKvasir dataset. The first iteration represents the original number of labeled images. The threshold for the max number of samples added to each class is set by the majority class, which is *BBPS-2-3* with 803 original samples.
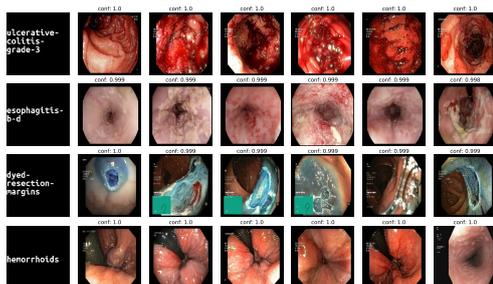

Fig. 5: Generated pseudo labels from the *ulcerative colitis grade 3*, *esophagitis B-D*, *dyed resection margins*, and *hemorrhoids* classes, from a subset of the 23 classes in HyperKvasir dataset. Above each pseudo label is the models confidence score which is generated by a teacher model on first iteration.

images from a uniform distribution of the overall training data.

IV. EXPERIMENT DETAILS

A. Datasets

The framework was tested on two public datasets. The first is the **HyperKvasir** dataset [17], which is one of the largest endoscopy datasets available, containing $110,079$ images, of which $10,662$ images are labeled, and 373 videos of anatomical landmarks and pathological findings, resulting in more than 1.1 million images and video frames altogether. The dataset contains four parts, labeled images, unlabeled images, segmented images, and videos. The labeled images are categorized into 23 classes, with a varying number of images per class. The classes contain a mix of pathological findings,

normal mucosa and anatomical landmarks, and degrees of bowel cleanliness. The second dataset is the **Kvasir-Capsule** dataset [18], which is a large VCE dataset collected from routine clinical examinations at Hospitals in Norway. The dataset consists of 118 videos which can be used to extract a total of $4,820,739$ image frames. This includes a total of 44,228 labeled, medically verified frames with a bounding box around the detected anomalies. There are 13 different classes of anomalies with a skewed number of samples per class, as some findings are more scarce than others. The classes contain images from anatomical landmarks, quality of mucosa view, and pathological findings.

Because Kvasir-Capsule contains frames extracted from videos, we had to consider what parts of the data went into each respective split (train and validation) to avoid near-duplicates between them. Suppose the data is split arbitrarily. In that case, there might be instances of the same finding spread across the different data splits, which would give an incorrect depiction of the model's actual performance. To avoid this, we split the data so that no frames of one specific finding are part of both the training and validation split.

B. Model evaluation

To make full use of the available data, we used two-fold cross-validation to evaluate our teacher-student framework. We ran our framework using one split as training data and the other for validation, and in the next run, we swapped the training data with the validation data and vice versa. The final performance metrics were calculated by averaging the metrics produced for each split, which is visualized in Figure 6 as a graph with metrics for each iteration of the teacher-student framework for HyperKvasir and Kvasir-capsule datasets. The metrics used to evaluate the framework are calculated by av-
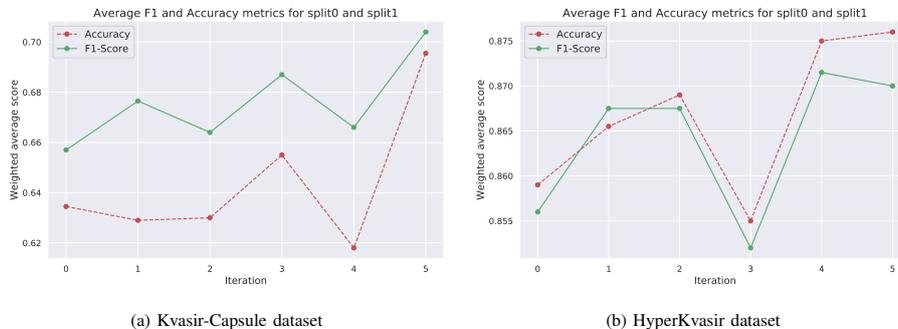
Fig. 6: Averaged accuracy and F_1-score for both splits after three iterations of switching out the teacher with the student. The pseudo label threshold was set to a max of 1,500 labels per class.
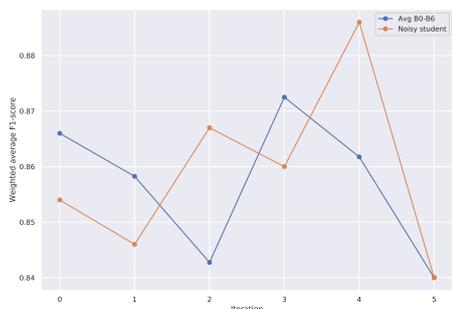

Fig. 7: A comparison of the performance on the HyperKvasir dataset of how the teacher-student framework performs when adding noise to the student. The blue line shows the results without adding noise, which is averaged over four runs using EfficientNet B0, B2, B4, and B6 for both the teacher and student models. The orange line shows the results with adding noise to the student.

Dataset	Architecture	Accuracy	F_1-score	Sensitivity	Precision	Specificity
Kvasir-Capsule	EfficientNetB0	0.634	0.657	0.635	0.717	0.599
	Teacher-student	**0.695**	**0.704**	**0.696**	**0.734**	**0.626**
HyperKvasir	EfficientNetB0	0.855	0.854	0.855	0.858	0.992
	Teacher-student	**0.893**	**0.886**	**0.893**	**0.890**	**0.993**

TABLE I: The results using our teacher student framework compared to standard classification training on the Kvasir-Capsule dataset and the HyperKvasir dataset. The results are averaged over two-fold validation.

see that when the teacher-student framework is run without injecting noise into the student, the models perform about the same as the teachers. When noise is added to the student model, we see a clear improvement in the overall performance, until in the last student iteration, the performance drop due to over-saturation of pseudo labels.

B. Iterative training

By putting the student back as the teacher and repeating the process, we found that the models generally perform better after mixing in more pseudo-labeled data with the labeled data. This allows the student to train on more images and therefore generalize better. In Table I, we present the results comparing our teacher-student framework using EfficientNetB0 and EfficientNetB6 against a EfficientNetB0 trained on only the labeled images. Running the framework for more than 3 or 4 iterations gave diminishing returns, or in some cases, an abrupt performance drop due to the over-saturation of pseudo labels in the training data. For this reason, the framework should incorporate an automatic look-back system to capture the best iteration and exit the training at an appropriate time. This was done manually in our work, but it should be implemented in future work.

eraging the weighted metric with regards to the class support, and by averaging the model metrics for each split of the data.

V. RESULTS

A. Adding noise to the student

Our experiments found that the student model performed better than the teacher. This is because the teacher tends to overfit on the data, while the student uses a model with more parameters, dropout, and data augmentation, which forces the student to better learn each class feature. For the teacher, we used EfficientNetB0 with no dropout and no augmentation. For the student model, we used EfficientNetB6 with a dropout rate of 30% and a range of image augmentations. The effect of adding noise to the student is shown in Figure 7, where we

311

VI. Conclusions

This paper presented a teacher-student-based framework for automatically classifying findings in the GI tract using video frames from VCE and standard endoscopies. The results show the potential of the proposed framework in utilizing unlabeled data, specifically by increasing the F_1-score by 3.2% for the HyperKvasir dataset and 4.7% for the Kvasir-Capsule dataset. As most medical data is collected without being annotated, we expect this type of self-learning paradigm will have a profound affect on the future of computer-assisted diagnoses in medicine. However, despite this framework being tested on two medical datasets, the framework should be tested on different datasets from different medical domains to better understand its generalizability and performance.

References

[1] Freddie Bray, Jacques Ferlay, Isabelle Soerjomataram, Rebecca L. Siegel, Lindsey A. Torre, and Ahmedin Jemal, "Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, 2018.

[2] H. Brenner, M. Kloor, and C. P. Pox, "Colorectal cancer," *The Lancet*, vol. 383, no. 9927, pp. 1490–502, 2014.

[3] Lindsey A. Torre, Freddie Bray, Rebecca L. Siegel, Jacques Ferlay, Joannie Lortet-Tieulent, and Ahmedin Jemal, "Global cancer statistics, 2012," *A Cancer Journal for Clinicians*, vol. 65, no. 2, 2015.

[4] World Health Organization - International Agency for Research on Cancer, "Estimated Cancer Incidence, Mortality and Prevalence Worldwide in 2012," 2012.

[5] David G Hewett, Charles J Kahi, and Douglas K Rex, "Efficacy and effectiveness of colonoscopy: how do we bridge the gap?," *Gastrointestinal Endoscopy Clinics*, vol. 20, no. 4, pp. 673–684, 2010.

[6] Si Hyung Lee, Byung Ik Jang, Kyeong Ok Kim, Seong Woo Jeon, Joong Goo Kwon, Eun Young Kim, Jin Tae Jung, Kyung Sik Park, et al., "Endoscopic experience improves interobserver agreement in the grading of esophagitis by los angeles classification: conventional endoscopy and optimal band image system," *Gut and liver*, 2014.

[7] Sascha C Van Doorn, Y Hazewinkel, James E East, Monique E Van Leerdam, Amit Rastogi, Maria Pellisé, Silvia Sanduleanu-Dascalescu, Barbara AJ Bastiaansen, Paul Fockens, and Evelien Dekker, "Polyp morphology: an interobserver evaluation for the paris classification among international experts," *The American journal of gastroenterology*, vol. 110, no. 1, pp. 180, 2015.

[8] Michal F Kaminski, Jaroslaw Regula, Ewa Kraszewska, Marcin Polkowski, Urszula Wojciechowska, Joanna Didkowska, Maria Zwierko, Maciej Rupinski, Marek P Nowacki, and Eugeniusz Butruk, "Quality indicators for colonoscopy and the risk of interval cancer," *New England Journal of Medicine*, vol. 362, no. 19, pp. 1795–1803, 2010.

[9] YuanPu Zheng, Lauren Hawkins, Jordan Wolff, Olga Goloubeva, and Eric Goldberg, "Detection of lesions during capsule endoscopy: physician performance is disappointing," *American Journal of Gastroenterology*, vol. 107, no. 4, pp. 554–560, 2012.

[10] Stefania Zammit Chetcuti and Reena Sidhu, "Capsule endoscopy–recent developments and future directions," *Expert Review of Gastroenterology & Hepatology*, pp. 1–11, 2020.

[11] Emanuele Rondonotti, Marco Soncini, Carlo Maria Girelli, Antonio Russo, Giovanni Ballardini, Guglielmo Bianchi, Paolo Cantù, Laura Centenara, Pietro Cesari, Claudio Camillo Cortelezzi, et al., "Can we improve the detection rate and interobserver agreement in capsule endoscopy?," *Digestive and Liver Disease*, 2012.

[12] Veronika Cheplygina, Marleen de Bruijne, and Josien PW Pluim, "Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis," *Medical Image Analysis*, vol. 54, pp. 280–296, 2019.

[13] Olivier Henaff, "Data-efficient image recognition with contrastive predictive coding," in *Proceedings of the International Conference on Machine Learning (PMLR)*, 2020, pp. 4182–4192.

[14] Ishan Misra and Laurens van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 6707–6717.

[15] Thang D. Bui, Sujith Ravi, and Vivek Ramavajjala, "Neural graph learning: Training neural networks using graphs," in *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, 2018, pp. 64–71.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738.

[17] Hanna Borgli, Vajira Thambawita, Pia H Smedsrud, Steven Hicks, Debesh Jha, Sigrun L Eskeland, Kristin R Randel, Konstantin Pogorelov, Mathias Lux, Duc T D Nguyen, Dag Johansen, Carsten Griwodz, Håkon K Stensland, Enrique Garcia-Ceja, Peter T Schmidt, Hugo L Hammer, Michael A Riegler, Pål Halvorsen, and Thomas de Lange, "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Springer Nature Scientific Data*, vol. 7, 2020.

[18] Pia H Smedsrud, Vajira Thambawita, Steven Hicks, Henrik Gjestang, Oda O Nedrejord, Espen Næss, Hanna Borgli, Debesh Jha, Tor Jan Derek Berstad, Sigrun L Eskeland, Mathias Lux, Håvard Espeland, Andreas Petlund, Duc Tien Dang Nguyen, Enrique Garcia-Ceja, Dag Johansen, Peter T Schmidt, Ervin Toth, Hugo L Hammer, Thomas de Lange, Michael A Riegler, and Pål Halvorsen, "Kvasir-capsule, a video capsule endoscopy dataset," *Springer Nature Scientific Data*, 2021.

[19] Mingxing Tan and Quoc Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceeding of the International Conference on Machine Learning (PMLR)*, 2019, pp. 6105–6114.

[20] Bradley J. Erickson, Panagiotis Korfiatis, Zeynettin Akkus, and Timothy L. Kline, "Machine Learning for Medical Imaging," *RadioGraphics*, vol. 37, no. 2, pp. 505–515, 2017.

[21] Kun-Hsing Yu, Andrew L Beam, and Isaac S Kohane, "Artificial intelligence in healthcare," *Nat Biomed Eng*, vol. 2, no. 10, pp. 719–731, 2018 10 2018.

[22] Geert Litjens, Francesco Ciompi, Jelmer M. Wolterink, Bob D. de Vos, Tim Leiner, Jonas Teuwen, and Ivana Išgum, "State-of-the-art deep learning in cardiovascular image analysis," *JACC: Cardiovascular Imaging*, vol. 12, no. 8_Part_1, pp. 1549–1565, 2019.

[23] Xin He, Shihao Wang, Shaohuai Shi, Xiaowen Chu, Jiangping Tang, Xin Liu, Chenggang Yan, Jiyong Zhang, and Guiguang Ding, "Benchmarking deep learning models and automated model design for covid-19 detection with chest ct scans," *medRxiv*, 2020.

[24] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10687–10698.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[26] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020, pp. 702–703.

[27] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger, "Deep Networks with Stochastic Depth," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 646–661.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

A.20 Paper XX - Using Preprocessing as a Tool in Medical Image Detection

Authors: Mathias Kirkerød, **Vajira Thambawita**, Michael Riegler, Pål Halvorsen

Abstract: In this paper, we describe our approach to gastrointestinal disease classification for the medico task at MediaEval 2018. We propose multiple ways to inpaint problematic areas in the test and training set to help with classification. We discuss the effect that preprocessing does to the input data with respect to removing regions with sparse information. We also discuss how preprocessing affects the training and evaluation of a dataset that is limited in size. We will also compare the different inpainting methods with transfer learning using a convolutional neural network.

Published: In the Proceedings of MediaEval 2020.

Candidate contributions: Vajira contributed to the conception and design of the study discussed in this manuscript. He guided the first author (master student) of this manuscript and contributed to analyzing the results of this study. Vajira contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective III, Sub-objective IV

Using preprocessing as a tool in medical image detection

Mathias Kirkerød [1,3], Vajira Thambawita[1,2], Michael Riegler[1,2,3], Pål Halvorsen[1,3]

[1]Simula Research Laboratory, Norway
[2]Oslo Metropolitan University
[3]University of Oslo

mathias.kirkerod@gmail.com,vajira@simula.no,michael@simula.no,paalh@simula.no

ABSTRACT

In this paper we describe our approach to gastrointestinal disease classification for the medico task at MediaEval 2018. We propose multiple ways to inpaint problematic areas in the test and training set to help with classification. We discuss the effect that preprocessing does to the input data with respect to removing regions with sparse information. We also discuss how preprocessing affects the training and evaluation of a dataset that is limited in size. We will also compare the different inpainting methods with transfer learning using a convolutional neural network.

1 INTRODUCTION

Medical image diagnosis is a challenging task in the industry of computer vision. In the last couple of years, as computing power has increased, machine learning has become a tool in the task of image detection, segmentation and classification. In this paper we are looking in depth how to use machine learning to help solve classification tasks on the data-set from the Medico task [8]. The Medico task focuses on image classification in the gastrointestinal (GI) tract. The data is divided in to 16 different classes.

Similar to other parts of image detection, the Medico dataset encounter the challenges that the amount of data is too small, or that the training data does not cover the full distribution of the data in the test case. The main goal of this task is to classify medical images. Our proposal is to use unsupervised machine learning for removal of the green corners that are in the Medico dataset. The details of the task are described in [5, 7].

2 APPROACH

Our approach is divided in to two steps: first preprocessing, then classifying. Our focus is mainly on the preprocessing of the data to remove the green corners in the medical images.

After the preprocessing the dataset we run it through a Convolutional Neural Network (CNN) based on transfer learning. We chose the CNN model based on the top 5 and top 1 accuracy of the pre-trained networks on the Keras documentation pages.

In our approach we use the InceptionResNetV2 [9] network. We also remove the top layer and replace it with a global average pooling layer and a dense 16 layer output, to match the number of classes wanted. In addition, we do not freeze any layers of the model. The five submissions that we run is with the same hyperparameters in the transferlearning model. This means that the difference in
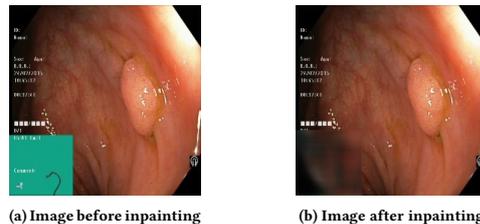
(a) Image before inpainting   (b) Image after inpainting

Figure 1: Differences of images after inpainting

results should only come from the different training datasets we use.

The medical data has 1 main feature that we focus on during the preprocessing, namely the green square in the bottom left corner. A neural network often struggle with areas with really sparse information. Our hypothesis is that just replacing the green area with a similar black area will not yield a better result.

We have a dataset that we use as a base case. This dataset was not augmented, other than shrinking the size of every image to a fixed resolution. The other datasets were augmented in a way that would cover up the green square in one way or another.

Our hypothesis it that if we recreate the areas as they would look like without any sparse areas, the classifier can focus on the right features for classifications. We propose 4 different methods on how to inpaint the corner area of the medical images.
An autoencoder [4], a context conditional generative adversarial network[2, 3], a context encoder [6], and a simple crop of the image.

2.1 Autoencoder

For the autoencoder approach, we created and trained a custom autoencoder from scratch. Our autoencoder consist of a encoder-decoder network, with 2D convolutions as well as rectified linear units as activation functions. In the layer between the encoder and the decoder we included a 25% dropout. [1]

To preprocess the medical data we feed the whole image through the encoder-decoder network. We take the loss of the whole reconstructed image, but only keep the inpainted part. Under training, the goal is to minimize the loss: $L(x, g(f(\tilde{x})))$ Where x is an image without a green corner, and $\tilde{x}$ is the same image with an artificial green corner. In theory we can replace any part of the image with this method.

A.20. Paper XX - Using Preprocessing as a Tool in Medical Image Detection

Kirkerød et al.

Table 1: Validation set' results

Method	REC	PREC	SPEC	ACC	MCC	F1
Autoencoder	0.929	0.929	0.981	0.929	0.923	0.928
CC-GAN	0.931	0.932	1.000	**0.931**	**0.926**	0.931
Contextencoder	0.926	0.928	0.945	0.926	0.920	0.926
Clipping	0.903	0.904	0.980	0.903	0.895	0.903
Non-augmenteted	0.925	0.927	0.981	0.925	0.919	0.924

2.2 Context encoder

For the context encoder approach, we created a new encoder-decoder network. Here the encoder has a similar structure to the autoencoder, but our decoder is only making outputs at the size of the desired area to inpaint. In addition to the loss generated from taking a MSE loss[6]:

$L(\hat{x}, g(f(x)))$ *Where $\hat{x}$ is an image with an artificial green corner, and x is the part that was replaced by the corner,* we include an adversarial loss, as described in [6].

With the context encoder we feed images without a green corner in to the encoder-decoder network. The output of the network is the same size as the area we want to fill.

2.3 Context conditional generative adversarial network

For the generative adversarial approach, we create a similar structure as the autoencoder. We have a constant 10% dropout at each layer in the discriminator. As with the autoencoder we have the same size input as output, but we only decide to keep the parts we want to inpaint.

We use the same type of loss as the context encoder, with 15% of the loss coming from a MSE loss, and the remaining 85% coming from the adversarial loss.

2.4 Clipping instead of inpainting

The last method was just to crop the images in a way that excluded the green corner. Since every image is scaled down to 256x256 px during preprocessing, the same is done with the clipped version (after the clip the size was reduced to 256x256).

The clipping was done in a way so that we had the most amount of center frame, and minimal amount of the bottom left corner, without sacrificing to much of the image.

3 RESULTS AND ANALYSIS

We made the augmented datasets before we trained the preprocessing model. This means that the transferlearning model did not augment the images at runtime. We split the data into a 70% train set, and a 30% validation set.

Our results on the test set are tabulated in Table 1. The official Results on the test set are tabulated in Table 2. Table 3 shows the confusion matrix from the CC-GAN from the official test set.

The results show that the CC-GAN got the highest MCC score with 0.926, and also the most realistic inpaintings. The context encoder had the lowest MCC score with 0.920, and also the worst inpainted areas. The official result did have the same pattern in

Table 2: Official Results

Method	REC	PREC	SPEC	ACC	MCC	F1
Autoencoder	0.915	0.915	0.994	**0.989**	0.910	0.915
CC-GAN	0.915	0.915	0.994	**0.989**	0.910	0.915
Contextencoder	0.910	0.910	0.994	0.988	0.905	0.910
Clipping	0.904	0.904	0.993	0.988	0.898	0.904
Non-augmented	0.917	0.917	0.994	**0.989**	**0.911**	0.917

Table 3: Confusion Matrix

A:ulcerative-colitis , **B**:esophagitis , **C**:normal-z-line , **D**:dyed-lifted-polyps , **E**:dyed-resection-margins , **F**:out-of-patient , **G**:normal-pylorus , **H**:stool-inclusions , **I**:stool-plenty , **J**:blurry-nothing , **K**:polyps , **L**:normal-cecum , **M**:colon-clear , **N**:retroflex-rectum , **O**:retroflex-stomach , **P**:instruments

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
A	510	0	1	0	1	0	1	0	69	0	5	24	0	3	0	13
B	3	401	68	0	1	0	5	0	0	0	0	0	0	0	1	0
C	0	153	489	0	0	0	3	0	0	0	0	0	0	0	0	0
D	0	0	0	502	39	0	0	0	0	0	3	0	1	0	0	45
E	0	0	0	46	517	1	0	0	0	0	1	0	0	0	0	15
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	2	2	3	0	0	0	547	0	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0	0	486	35	0	0	0	0	0	0	0
I	3	0	0	0	2	0	0	1	1857	0	3	1	0	0	0	3
J	1	0	0	0	0	0	1	0	1	36	0	0	1	0	0	0
K	8	0	1	5	2	3	4	0	0	0	349	17	0	2	1	55
L	11	0	1	2	1	0	1	0	1	1	11	542	0	0	0	3
M	2	0	0	0	0	0	0	18	2	0	1	0	1064	0	1	3
N	2	0	0	1	1	0	0	0	0	0	1	0	0	183	4	5
O	0	0	0	0	0	0	0	0	1	0	0	0	0	2	389	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	131

MCC score, though the base case got the best result. In both cases the clipping gave significantly worse result.

As expected, most of the images was classified correctly, but we had some problems distinguishing between esophagitis and normal-z-line. We also had a few cases of instruments where there were none.

4 CONCLUSION

In general, when training on a dataset that is homogeneous, the preprocessing is less valuable. We want to remove areas with sparseness, and areas that has nothing to do with the classification.

In our example we used 3 different methods to do this, and we had no improvements in the results. As we can see from the validation set, we saved under a percent on the best method, and we got a worse score on the official results.

We conclude that preprocessing the Medico dataset is not worth the hassle. The effort put in to preprocess the images yields little to no improvement to the result. We recommend that the time is used to find the right network, with the right hyper-parameters instead. A reason to lackluster results might be caused that the training and the test set have the same green squares in the same classes. We suspect that the similarity in the test and train set makes the squares an essential part of the image. We believe that the result would be much better if the test set would be completely without the squares, as they would if they were "real time" images.

In a future test we would also recommend removing the four black edges too. With the images being round, this might be a challenge, since there are no full-resolution images (without zoom) that captures the edges. With the medico dataset, this method will probably not give a better score, on the basis that every image in the dataset has the same four black corners.

Appendix A. Published Articles

316

REFERENCES

[1] Aaron Courville Yoshua Bengio David Warde-Farley, Ian J. Goodfellow. 2013. An empirical analysis of dropout in piecewise linear networks. abs/1609.05158 (2013). arXiv:1312.6197v2 https://arxiv.org/pdf/1312.6197v2

[2] Emily L. Denton, Sam Gross, and Rob Fergus. 2016. Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks. *CoRR* abs/1611.06430 (2016). arXiv:1611.06430 http://arxiv.org/abs/1611.06430

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[4] Y. Kamp H. Bourlard. 1988. Auto-Association by Multilayer Perceptrons and Singular Value Decomposition. (1988). http://ace.cs.ohio.edu/~razvan/courses/dl6890/papers/bourlard-kamp88.pdf

[5] Pål Halvorsen Thomas de Lange Kristin Ranheim Randel Duc-Tien Dang-Nguyen Mathias Lux Konstantin Pogorelov, Michael Riegler. 2018. Mediaeval information. http://multimediaeval.org/mediaeval2018/medico/. (2018). Accessed: 2018-10-16.

[6] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context Encoders: Feature Learning by Inpainting. *CoRR* abs/1604.07379 (2016). arXiv:1604.07379 http://arxiv.org/abs/1604.07379

[7] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. 2017. KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference (MMSys'17)*. ACM, New York, NY, USA, 164–169. https://doi.org/10.1145/3083187.3083212

[8] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Thomas De Lange, Kristin Ranheim Randel, Duc-Tien Dang-Nguyen, Mathias Lux, and Olga Ostroukhova. 2018. Medico Multimedia Task at MediaEval 2018. (2018).

[9] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR* abs/1602.07261 (2016). arXiv:1602.07261 http://arxiv.org/abs/1602.07261

A.21 Paper XXI - Unsupervised Preprocessing to Improve Generalisation for Medical Image Classification

Authors: Mathias Kirkerød, Rune Johan Borgli, **Vajira Thambawita**, Steven Hicks, Michael Alexander Riegler, Pål Halvorsen

Abstract: Automated disease detection in videos and images from the gastrointestinal (GI) tract has received much attention in the last years. However, the quality of image data is often reduced due to overlays of text and positional data. In this paper, we present different methods of preprocessing such images and we describe our approach to GI disease classification for the Kvasir v2 dataset. We propose multiple approaches to inpaint problematic areas in the images to improve the anomaly classification, and we discuss the effect that such preprocessing does to the input data. In short, our experiments show that the proposed methods improve the Matthews correlation coefficient by approximately 7% in terms of better classification of GI anomalies.

Published: In proceedings of 13th International Symposium on Medical Information and Communication Technology (ISMICT), 2019.

Candidate contributions: Vajira contributed to the conception, design of this study, and analysis of the results of this manuscript. Vajira contributed to drafting and revising this extended version of the working notepaper: "Using preprocessing as a tool in medical image detection".

Thesis objectives: Sub-objective III, Sub-objective IV

Appendix A. Published Articles

Unsupervised preprocessing to improve generalisation for medical image classification

Mathias Kirkerød, Rune Johan Borgli
Simula Research Laboratory, Norway
University of Oslo, Norway
mathiaki@ifi.uio.no, rune@simula.no

Vajira Thambawita, Steven Hicks, Michael Alexander Riegler, Pål Halvorsen
SimulaMet - Simula Metropolitan Center for Digital Engineering, Norway
{vajira, steven, michael, paalh}@simula.no

Abstract—Automated disease detection in videos and images from the gastrointestinal (GI) tract has received much attention in the last years. However, the quality of image data is often reduced due to overlays of text and positional data. In this paper, we present different methods of preprocessing such images and we describe our approach to GI disease classification for the Kvasir v2 dataset. We propose multiple approaches to inpaint problematic areas in the images to improve the anomaly classification, and we discuss the effect that such preprocessing does to the input data. In short, our experiments show that the proposed methods improve the Matthews correlation coefficient by approximately 7% in terms of better classification of GI anomalies.

Index Terms—Machine learning, GAN, Autoencoder, Inpainting

I. INTRODUCTION

In the field of computer vision, image-based disease detection has become a popular area of research. For example, algorithms based on deep neural networks have been used to automatically analyse the human digestive system for anomalies such as polyps, lesions and other common illnesses. This is important as the detection and removal of colon polyps is the main prevention method of colorectal cancer, which ranks within the top-three terminal cancer types for both men and woman [1]. Automatically detecting this disease goes a long way of aiding doctors to perform a more thorough analysis of their patients, and has the potential of saving lives. In addition to gastroenterology, we continue to see machine learning based classification systems appear in nearly every branch of medicine.

In recent years, deep learning based algorithms have become a popular method for solving these problems. Aided by the rapid advancement of computational power due to the efficiency of GPUs, deep learning has shown state-of-the-art performance across numerous fields, including medicine. However, deep neural networks are only as good as the data used to train them. Thus, data which contains artefacts such as text and overlays may negatively impact the performance of models trained on this data. This is particularly problematic in medicine, as the selection of datasets is often limited, and the datasets available may include artefacts from the software

Fig. 1: Example image from the Kvasir dataset with included overlays and black borders.

the doctors use to analyse the images/videos (e.g., overlays, text, and other information).

In this work, we look at improving the quality of a publicly available endoscopy dataset called Kvasir [2], which contains several of the artefacts previously mentioned (example shown in Figure 1). We hope that this shows that there are more ways of improving the performance of a deep neural network than increasing its number of training samples. This work can be seen as an extension of our approach to this years MediaEval Medico task [3], where we presented a similar technique, albeit to a much lesser extent [4]. Additionally, a recent study using Kvasir for training deep learning based models showed that these artefacts directly impacted the classification performance of said models, showing that there is potential room for improvement [5].

The main contributions of this paper are (i) we present different methods for preprocessing data to be able to create better generalisable models, (ii) a detailed cross-dataset evaluation of the methods used and (iii) we report classification performance across different datasets.

318

II. RELATED WORK

As mentioned in the introduction, medical image classification has been a heavily researched area. Research gathered by Lu and Weng [6] give current practices, problems, and prospects of image classification.

Our methods for inpainting bears a resemblance to context-encoder made by Pathak et al. [7] who introduce an encoder-decoder network in style close to our proposed generative adversarial network. However, a big difference is the use of a channel-wise fully connected layer in their model to share information around in the image space. This part was not necessary for us, given the homogeneity of the medical images coupled with the use of a non-random filter for inpainting.

Denton et al. [8] presented a model for inpainting close to the context-conditional adversarial network presented by Pathak et al. that is also trained on non-medical images, with random filter placement during training and evaluation. Their results showed that their generative adversarial network (GAN) model was capable of producing semantically meaningful inpaintings in a diverse set of images.

Previously, Hicks et al. [5] applied various preprocessing steps to Kvasir based on analysis conducted on common CNN architectures. Using heat maps and saliency maps, they discovered a common issue where artefacts such as text, black borders, and green navigation boxes were directly correlated to the misclassification of some images. In an attempt to correct this issue, they applied various preprocessing steps to the training data, namely cropping black borders and blacking out the green navigation box. Their results revealed improvement in all cases of data preprocessing, and in the best case, they achieved an increase of Matthews correlation coefficient (MCC) by approximately 3%.

In this paper, we aim to improve on this work by not simply removing borders and green navigation boxes. We also try to replace the artefacts using ideas from GAN inpainting to generate an automatically generated mask which attempts to replicate what would have been there if not for said overlay artefacts.

III. APPROACH

By using machine learning, we aim to classify medical images from the gastrointestinal (GI) tract correctly. With this approach, it is common to use a dataset for training and validation, with a separate set for testing. In practice the dataset we test on is never seen by the model before its evaluation. This is the main reason why we often struggle to get the same level of accuracy when evaluating our model if the data originates from different sources. In our case, the test data from the CVC dataset differs from the training data in both the image content and size. When this problem arises, it is practice to use domain-specific knowledge to help training, and if the amount of training data is small, methods like K-fold cross-validation [9] can also be used to improve the results.

For this paper, we focus on inpainting as a form of generalised preprocessing. We do this to remove dataset specific overlays for better classification on new datasets no matter the source of the dataset. Furthermore, we have also chosen to use

(a) Original image

(b) Square inpainted

(c) Edges inpainted

(d) Corner and edges inpainted
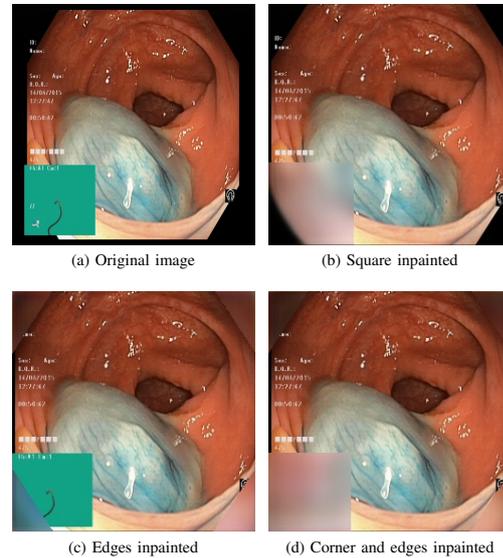
Fig. 2: Here we have a sample of what we want to achieve. (a) Original from the Kvasir dataset. Here we also see extended edges that we can cut away without any machine learning. (b) Same image without edges and the green square. (c) Same image with new corners, (d) Same image with both new corners and new area for green square

the same Bayesian optimisation techniques as in the Borgli et al. paper [10] to find the optimal network for classification. With both hyperparameter optimisation and inpainting, our goal is to get the highest classification score on the CVC datasets.

A. Preprocessing

As discussed, the Kvasir dataset has some unwanted artefacts that are present in a good portion of the data. Some of the unwanted artefacts are Kvasir specific, and some are general artefacts when capturing images from the colon. First, the camera used in colonoscopies has an exceptionally wide lens. This setup takes good medical images but comes with the drawback that the images are not rectangular. Because of this, the camera needs to add black corners and borders to save the images. Another unwanted artefact that is Kvasir specific is an unwanted additional overlay added to the images. They are added post-image-capture by the medical staff, and they show essential information about the patient. As we can see from this, we have multiple areas in the images with pixels not originating from the patient, and subsequently contains no information relevant for classification.

A neural network will also often struggle with areas with really sparse information. Because of this, we believe that just replacing the green area with a similar black area will not yield

TABLE I: Details of all datasets used in the experiments.

BC: Black corner. **GS**: Green square. **BC+GS**: Black corner and Green square

Dataset labels	Size	Inpainted area	Generator network used
D-I	256x256 px	-	-
D-II	256x256 px	BC	Autoencoder
D-III	256x256 px	GS	Autoencoder
D-IV	256x256 px	BC+GS	Autoencoder
D-V	256x256 px	BC	GAN
D-VI	256x256 px	GS	GAN
D-VII	256x256 px	BC+GS	GAN
D-VIII	512x512 px	-	-
D-IX	512x512 px	BC	Autoencoder
D-X	512x512 px	GS	Autoencoder
D-XI	512x512 px	BC+GS	Autoencoder
D-XII	512x512 px	BC	GAN
D-XIII	512x512 px	GS	GAN
D-XIV	512x512 px	BC+GS	GAN

TABLE II: Details of experiments.

Test	Training datasets	Testing dataset	Network model
T1	D-I - D-VII	Kvasir V2	DenseNet121
T2	D-I - D-VII	CVC-12k	DenseNet121
T3	D-I - D-VII	CVC-356	DenseNet121
T4	D-I - D-VII	CVC-356	InceptionResnetV2
T5	D-VIII - D-XIV	CVC-356	DenseNet121

choose Densenet121 [14] as our default architecture for training our new datasets. We are also interested in the accuracy compared to a more general classification network. We ran model D-I - D-VII with the pretrained InceptionResNetV2 [15] network. We chose this network because of its high accuracy on the Keras websites [16], and thus we hypothesise that the model will be generally good without hyperparameter optimisation. In both cases, we remove the top layer and replace it with a global average pooling layer and a dense eight layer output to match the number of classes in the training dataset.

Our focus is the comparison between the generated datasets and the baseline; hence we do not change the hyperparameters after they are chosen. We believe this sets up a valid comparison since the only difference in score should come from the differences in the dataset and not the classification model. An overview of our experiments are shown in Table II, where Models T1 - T3 is a direct comparison on how well we have generalised our model, while Models T4 & T5 show how changing models will affect the results. Below, we give brief a description of the three datasets used.

a) The Kvasir V2 dataset [2]: The Kvasir V2 dataset consists of 8,000 images from the GI tract. Several of these images contain artefacts such as navigation boxes (green box as seen Figure 1), overlayed text, black borders, and black edges. With our first hypothesis in mind, we assume that the dataset with the inpainted rounding corners (D-II & D-IV) will do slightly better than the baseline (D-I). This is because the training and test data is from the same set, and subsequently our generalisation will not help. That leaves us with the only way to improve the result is to remove sparseness.

b) The CVC-356 dataset [17]: The CVC-356 dataset consists of 2,285 images from the lower GI tract. CVC-356 does not have images with green boxes. It does have images with black borders, and rounded black edges. As stated in our second hypothesis; the inpainting of the green square will presumably give the best result. This is because, as stated, the CVC-356 images has the same black rounded corners as Kvasir, but lacks the green squares.

c) The CVC 12k dataset [17]: The CVC-12k dataset consists of 11954 images from the lower GI tract, with a resolution of primarily 288x384 px. Given the similarity with the CVC-356 dataset, this will presumably follow our second hypothesis stating that the inpainting of the green square would give the best result. Given that the CVC-12k images has the same black rounded corners as Kvasir, but lacks the green squares.

the best result. However, we expect improvement if we instead try to inpaint both the green corner and the black edges with data gathered from similar images. Furthermore, by removing areas that are specific for that dataset, we believe the model will be far better at generalising to other datasets within the same domain. In our case, the area we will be inpainting is the green area, since it is not present in the CVC datasets, and most other medical datasets are also without it.

With our two hypotheses, we have two different features that we believe will make the classification harder. We first aim to inpaint both areas separately to see how each of them affects classification. We also want to try to collectively remove both areas to see if a combined mask will yield a better or worse result.

With this in mind, we use two different methods for inpainting the desired areas. First, an autoencoder (AE) [11] as a lightweight way to generate new data, and second we use a GAN [12] as a more sophisticated generator. Both methods are unsupervised learning methods to generate new data within the distribution of the original dataset.

For our experiments, we scale our data to a constant resolution. We run four experiments with 256x256 pixels (px) resolution, and four experiments at 512x512 px. Our change in resolution is to compare the effect it has compared to our standard 256x256 px. With this configuration, we end up with 14 augmented datasets shown in table I.

B. Classification

Our research from the 2018 MediaEval workshop showed less desirable result compared to other projects that researched on the same dataset [13] [10]. Therefore one of our goals is to make our model more realistic by using a model that works better on the augmented Kvasir dataset. Using the Bayesian hyperparameter optimiser on our newly created datasets, we

IV. PREPROCESSING TOOLS

The networks used for inpainting are based on the network presented in the Mediaeval conference [4]. Both networks are using on masking, where only the parts of the image corresponding to a mask was inpainted.

A. Autoencoder

The first approach we created and trained was a custom autoencoder [11] from scratch. Our autoencoder consists of an encoder-decoder network, with 2D convolutions as well as rectified linear units as activation functions, and a 25% dropout between the encoder and decoder. The network used is a modification of the network presented in [4]. The modifications are a smaller batch size and a more consistent filter size throughout the network. These modifications were made to make more credible results, and to get a lower error during training. The loss function was also modified to solely train on parts of the images that were modified. This lead to a larger and more accurate gradient descent, which also contributed to a better reconstruction.

B. Context conditional generative adversarial network

For the GAN approach, we create a similar structure to the autoencoder. We have a generator-discriminator network that serves much of the same functionality as the encoder-decoder network in the autoencoder. As with the autoencoder, we have the same size input as output, but we only decide to keep the parts we want to inpaint. The model we ended up with is closely inspired to the model made by Denton et al. in [18]. The main differences are the number of layers used, and the lack of a low-resolution image as an extra input.

V. RESULTS

We divide our results into two sections, preprocessing and classification. In our preprocessing section, we discuss the appearance of the dataset, and how close the results are to the ground truth. In our classification section, we discuss the rate of generalisation and rate of success.

A. Preprocessing

Since there are no specific metrics associated with the training of Autoencoders and GANs, we used the mean square error of the ground truth as a metric of our progress. Figure 3 from the z-line shows how the two different models perform on the two different sizes. This is a typical case where both the GAN and the AE are fairly similar, except for more features added by the GAN. The features are most present in the smaller images, as the images are easier to train on, and subsequently easier to add complex local features too.

B. Classification

We evaluated our model on both the Kvasir and the CVC dataset as described in the classification section (III-B). When presenting our results, our main point of comparison is the MCC [19]. In addition to the MCC score, we use F1, precision

(a) GAN generated image at 512x512 px

(b) Autoencoder generated image at 512x512 px

(c) GAN generated image at 256x256 px

(d) Autoencoder generated image at 256x256 px
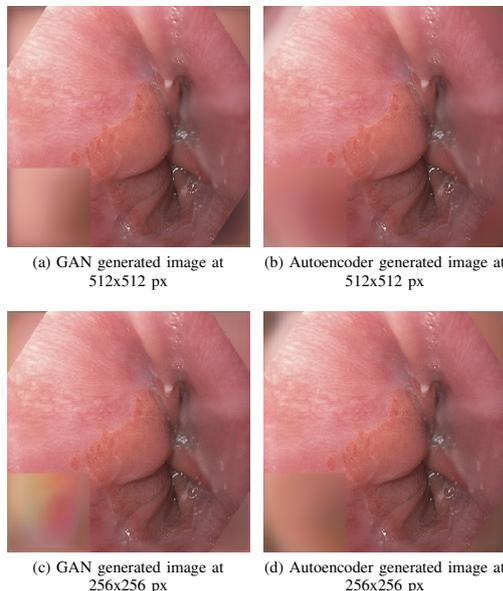
Fig. 3: Same image from the z-line with four different inpainting attempts. Each image is re-sized to fit in the figure.

and recall as metrics when presenting our results. In addition to the best MCC score, we present the average MCC score as an indicator of the general success of the method in question.

Since our task was to improve classification and cross-dataset generalisability through inpainting, each table has its first row as the dataset without any inpainting, followed by the rest of the datasets. The first column is the maximum MCC score of the runs. Then we give the maximum F1 score followed by the maximum precision and recall. The last column gives us the average MCC of all four runs for each model.

First, we evaluated our results on the three datasets: Kvasir, CVC-356 and CVC-12k. Here our goal was to see the general improvement based only on inpainting and dataset. Then we evaluated the InceptionResNetV2 network on the CVC-356 dataset, and lastly, re-evaluated the CVC-356 network, at double image size.

a) Kvasir, Test T1: These are our results from training and evaluating on the Kvasir v2 dataset with the 5,600 image training set, 800 image validation set, and 1,600 image test set split. Table III shows the highest value for each of the six methods compared to the highest baseline.

As we can see in the results shown in Table III, we got the highest MCC score on the baseline dataset. Both the best and average scores were highest for the baseline, but the average was consistently high for all methods. As we recall, we predicted that we expected a higher MCC score for the Autoencoder inpainting the black corner and the GAN

Appendix A. Published Articles

TABLE III: Test T1, Kvasir dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	MCC (AVG)	MCC (SD)
D-I	0.9307	0.9394	0.9396	0.9394	0.9163	0.0166
D-II	0.9150	0.9254	0.9303	0.9250	0.9053	0.0102
D-III	0.9212	0.9310	0.9347	0.9306	0.9040	0.0167
D-IV	0.9187	0.9287	0.9298	0.9288	0.9105	0.0057
D-V	0.9208	0.9308	0.9316	0.9306	0.9108	0.0067
D-VI	0.9096	0.9204	0.9226	0.9206	0.9055	0.0038
D-VII	0.8960	0.9094	0.9174	0.9081	0.8926	0.0049

TABLE IV: Test T2, CVC-12k dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	MCC (AVG)	MCC (SD)
D-I	0.2897	0.5558	0.6968	0.6067	0.2723	0.0329
D-II	0.3031	0.5413	0.7148	0.5927	0.2675	0.0250
D-III	0.3197	0.6152	0.7050	0.6600	0.2649	0.0374
D-IV	0.2956	0.4663	0.7632	0.5156	0.2733	0.0225
D-V	0.2967	0.5451	0.7072	0.5965	0.2523	0.0440
D-VI	0.2803	0.4548	0.7571	0.5038	0.2244	0.0410
D-VII	0.2225	0.5740	0.6451	0.6236	0.1984	0.0195

TABLE V: Test T3, CVC-356 dataset on DenseNet121

Dataset	MCC	F1	Precision	Recall	MCC (AVG)	MCC (SD)
D-I	0.7070	0.9137	0.9132	0.9164	0.5904	0.1104
D-II	0.5153	0.7846	0.8153	0.8065	0.4861	0.0307
D-III	0.7325	0.9402	0.9535	0.9348	0.6465	0.0978
D-IV	0.6631	0.9264	0.9410	0.9194	0.5637	0.1011
D-V	0.5714	0.8387	0.8487	0.8516	0.4557	0.1002
D-VI	0.7150	0.9214	0.9206	0.9225	0.6334	0.0819
D-VII	0.7466	0.9370	0.9391	0.9356	0.4576	0.1941

TABLE VI: Test T4, CVC-356 dataset on InceptionResNetV2

Dataset	MCC	F1	Precision	Recall	MCC (AVG)	MCC (SD)
D-I	0.4038	0.8851	0.9130	0.8678	0.2999	0.0841
D-II	0.2221	0.7957	0.7958	0.7955	0.1227	0.0900
D-III	0.0745	0.4489	0.5535	0.5131	0.0299	0.0374
D-IV	0.3147	0.7793	0.7730	0.7916	0.1636	0.1197
D-V	0.1802	0.5434	0.6201	0.5985	0.0446	0.0923
D-VI	0.3276	0.8372	0.8429	0.8323	0.2234	0.0826
D-VII	0.2738	0.6754	0.6938	0.7106	0.1417	0.1230

inpainting the black corner. The results do not show a clear indication that the baseline was the best method, nor that there are any good ways to inpaint this dataset.

b) CVC-12k, Test T2: The T2 test case was trained on the Kvasir v2 dataset with the 5,600 image training set and the 800 image validation set, then evaluating on the CVC-12k dataset. Table IV shows the highest value for the six methods compared to the highest baseline, with four runs each.

As we can see in the results, shown in Table IV, we got the highest MCC score on the dataset with the inpainted green square made by the autoencoder. Also, the average score was consistently higher for the autoencoder datasets compared to the GAN datasets. The results give a small indication that inpainting the green area with an autoencoder might give a better result compared to the baseline.

c) CVC-356, Test T3: The T3 test case was, as test case T2, trained on the Kvasir v2 and evaluated on the CVC-356 dataset. The table V shows the highest value for each of the six methods compared to the highest baseline, with four runs each.

As we can observe in the results shown in Table V, we got the highest MCC score on the dataset with the inpainted green square made by the autoencoder and the GAN. We can also see a constant higher value for both datasets inpainting the green area. The highest value was from the dataset with both corner and square inpainting, but this is most likely just a lucky result, given the low average MCC. The results give a reasonable indication that inpainting the green area will give a better result compared to the baseline.

d) InceptionResNetV2, Test T4: These are our results from training on the Kvasir v2 dataset with the 5,600 image training set and the 800 image validation set, then evaluating on the CVC-365 dataset. The table VI shows the highest value

for each of the six methods compared to the highest baseline, with four runs each. In this run we used the InceptionRes-NetV2 network to train our model.

As we can see from the results shown in Table VI, we got the highest MCC score on the baseline dataset. From our tests, it looked like the overall scores were much lower here compared to our DenseNet121 models, and in general, we got more unpredictable scores.

e) Double image size, Test T5: These are the results from training on the Kvasir v2 dataset with the 5600 image training set and the 800 image validation set, then evaluating on the CVC-365k dataset. The table VII shows the highest value for each of the six methods compared to the highest baseline, with four runs each. Here we have doubled the size of the images for the training and evaluation set to see how size affects the results.

On the CVC-356 dataset at 512x512 px resolution, we see a generally lower MCC score compared to the same dataset at 256x256 px. Our best average results came from the dataset with both inpainted corners and inpainted squares, but it looks like the more inpainting, the better. The results give a small indication that inpainting large areas with sparse information might give a better result compared to the baseline, at least compared to smaller areas.

Overall, we can observe through all experiments that inpainting can both improve and worsen the results. In general, inpainting works best when applied in dataset specific artefacts that are not present in the test set.

VI. Discussion

Our first hypothesis was that removal of the black edges and corners around the images would result in a better classification and better generalisation. Our results also show that training and testing on the same dataset gave approximately

TABLE VII: Test T5, CVC-356 dataset with double resolution

Dataset	MCC	F1	Precision	Recall	MCC (AVG)	MCC (SD)
D-VIII	0.5865	0.8711	0.8702	0.8770	0.4696	0.1560
D-IX	0.6447	0.8992	0.8980	0.9015	0.4775	0.1142
D-X	0.4346	0.8894	0.9157	0.8735	0.3754	0.0709
D-XI	0.6449	0.8998	0.8986	0.9019	0.5935	0.0402
D-XII	0.7189	0.9294	0.9311	0.9282	0.4499	0.2110
D-XIII	0.5956	0.8891	0.8880	0.8905	0.5547	0.0604
D-XIV	0.7234	0.9235	0.9228	0.9247	0.5737	0.1173

the same MCC score, with and without corners. In addition, we observed that the removal of areas within the images with no relevant information did not give any better results, given the same training and test distribution. This was not the case when the images were up-scaled above their original size, as we saw a much better result when the areas were inpainted. We also observed that by removing the corners on the Kvasir set during training, the testing on the CVC-sets we did not get any better results in general. This was as expected since all the images had black edges, and removing them from training would make the datasets less alike. Our second hypothesis was concerning the removal of the green squares in the training set. With this, we wanted to see how the inpainted training sets affected to the test set that did not originate from the original distribution. We observed good results for both the CVC-12k set and the CVC-356 set. For the set, we deemed most realistic, namely the CVC-356 set, we saw that our score consistently was higher both for the average and the max MCC. Lastly, using a non-optimised network gives a lower MCC score when inpainting. In general, we see that inpainting to only remove sparseness will often worsen the results when the test and training set is from different sources. The same goes for excessive inpainting.

VII. Conclusions

Our two main hypotheses regarding types of inpainting for this paper were about how it would affect classification. We tested this on various datasets with different models at different sizes to see how the datasets affected the classification score. From our experiments, we can see that inpainting can help when generalising the training data to other datasets. In our GI anomaly classification experiments, our models show an average increase of at least 7% MCC score when using an optimal network for testing on images that are not from the same domain as the training data, shown in VII. When working with bigger size images, and subsequently larger areas with sparse information, it seems that inpainting does a better job, compared to smaller images. The results coincide with the previous work done [4].

References

[1] B. Stewart and C. Wild, *International Agency for Research on Cancer. World Cancer Report 2014 (International Agency for Research on Cancer)*. World Health Organization, 2014.

[2] K. Pogorelov, K. R. Randel, C. Griwodz, S. L. Eskeland, T. de Lange, D. Johansen, C. Spampinato, D.-T. Dang-Nguyen, M. Lux, P. T. Schmidt, M. Riegler, and P. Halvorsen, "Kvasir: A multi-class image dataset for computer aided gastrointestinal disease detection," in *Proceedings of the 8th ACM on Multimedia Systems Conference*, ser. MMSys'17. New York, NY, USA: ACM, 2017, pp. 164–169. [Online]. Available: http://doi.acm.org/10.1145/3083187.3083212

[3] K. Pogorelov, M. Riegler, P. Halvorsen, S. Hicks, K. Randel, D.-T. Dang-Nguyen, M. Lux, O. Ostroukhova, and T. Lange, "Medico multimedia task at mediaeval 2018." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[4] M. Kirkerød, V. Thambawita, M. Riegler, and P. Halvorsen, "Using preprocessing as a tool in medical image detection." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[5] S. Hicks, M. Riegler, P. Konstantin, K. V. nonsen, T. de Lange, D. Johansen, M. Jeppsson, K. R. Randel, S. Eskeland, and P. Halvorsen, "Dissecting deep neural networks for better medical image classification and classification understanding," 2018.

[6] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, 2007. [Online]. Available: https://doi.org/10.1080/01431160600746456

[7] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.278

[8] E. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," 2016.

[9] S. M., "Cross-validatory choice and assessment of statistical predictions." *Journal of the Royal Statistical Society*, no. 36(2), pp. 111–147, 1974.

[10] R. J. Borgli, P. Halvorsen, M. Riegler, and H. K. Stensland, "Automatic hyperparameter optimization in keras for the mediaeval 2018 medico multimedia task." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[11] Y. K. H. Bourlard, "Auto-association by multilayer perceptrons and singular value decomposition," 1988. [Online]. Available: http://ace.cs.ohio.edu/ razvan/courses/dl6890/papers/bourlard-kamp88.pdf

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] S. Hicks, P. H. Smedsrud, P. Halvorsen, and M. Riegler, "Deep learning based disease detection using domain specific transfer learning." CEUR Workshop Proceedings (CEUR-WS.org), 2018.

[14] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.243

[15] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[16] F. Chollet, "Applications - keras documentation. [online]," https://keras.io/applications/, 2019, accessed: 2019-01-07.

[17] J. Bernal and H. Aymeric, "Miccai endoscopic vision challenge polyp detection and segmentation," 2017, accessed: 2019-01-07. [Online]. Available: https://endovissub2017-giana.grand-challenge.org/home/

[18] E. L. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *CoRR*, vol. abs/1611.06430, 2016. [Online]. Available: http://arxiv.org/abs/1611.06430

[19] B. Matthews, "Comparison of the predicted and observed secondary structure of t4 phage lysozyme," *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442 – 451, 1975. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0005279575901099

A.22 Paper XXII - GANEx: A Complete Pipeline of Training, Inference and Benchmarking GAN Experiments

Authors: Vajira Thambawita, Hugo Lewi Hammer, Michael Riegler, Pål Halvorsen

Abstract: Deep learning (DL) is one of the standard methods in the field of multimedia research to perform data classification, detection, segmentation and generation. Within DL, generative adversarial networks (GANs) represents a new and highly popular branch of methods. GANs have the capability to generate, from random noise or conditional input, new data realizations within the dataset population. While generation is popular and highly useful in itself, GANs can also be useful to improve supervised DL. GAN-based approaches can, for example, perform segmentation or create synthetic data for training other DL models. The latter one is especially interesting in domains where not much training data exists such as medical multimedia. In this respect, performing a series of experiments involving GANs can be very time consuming due to the lack of tools that support the whole pipeline such as structured training, testing and tracking of different architectures and configurations. Moreover, the success of generative models is highly dependent on hyper-parameter optimization and statistical analysis in the design and fine-tuning stages. In this paper, we present a new tool called GANEx for making the whole pipeline of training, inference and benchmarking GANs faster, more efficient and more structured. The tool consists of a special library called FastGAN which allows designing generative models very fast. Moreover, GANEx has a graphical user interface to support structured experimenting, quick hyper-parameter configurations and output analysis. The presented tool is not limited to a specific DL framework and can be therefore even used to compare the performance of cross frameworks.

Candidate contributions: Vajira contributed to the conception and the design of this manuscript. He implemented a new GUI-based tool named GANEx to train, do

inference, and evaluate Generative Adversarial Networks (GANs) using pre-defined
GAN implementations. Vajira has published this tool in a Github repository (https://github.com/v
to use in the research community who need to train GANs without coding. He con-
tributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective IV

GANEx: A complete pipeline of training, inference and benchmarking GAN experiments

Vajira Thambawita
SimulaMet
Norway
vajira@simula.no

Hugo Lewi Hammer
Oslo Metropolitan University
Norway
hugoh@oslomet.no

Michael Riegler
SimulaMet
Norway
michael@simula.no

Pål Halvorsen
SimulaMet, Norway
Oslo Metropolitan University, Norway
paalh@simula.no

Abstract—**Deep learning (DL) is one of the standard methods in the field of multimedia research to perform data classification, detection, segmentation and generation. Within DL, generative adversarial networks (GANs) represents a new and highly popular branch of methods. GANs have the capability to generate, from random noise or conditional input, new data realizations within the dataset population. While generation is popular and highly useful in itself, GANs can also be useful to improve supervised DL. GAN-based approaches can, for example, perform segmentation or create synthetic data for training other DL models. The latter one is especially interesting in domains where not much training data exists such as medical multimedia. In this respect, performing a series of experiments involving GANs can be very time consuming due to the lack of tools that support the whole pipeline such as structured training, testing and tracking of different architectures and configurations. Moreover, the success of generative models is highly dependent on hyper-parameter optimization and statistical analysis in the design and fine-tuning stages.**

In this paper, we present a new tool called GANEx for making the whole pipeline of training, inference and benchmarking GANs faster, more efficient and more structured. The tool consists of a special library called FastGAN which allows designing generative models very fast. Moreover, GANEx has a graphical user interface to support structured experimenting, quick hyper-parameter configurations and output analysis. The presented tool is not limited to a specific DL framework and can be therefore even used to compare the performance of cross frameworks.

Index Terms—**GANs, Neural Networks, Graphical User Interface, GAN Experiments, GAN Library, GAN Statistics**

I. INTRODUCTION

Generative models have become an active research area in recent years as a result of the introduction of generative adversarial networks (GANs) [1], [2]. Research such as deep convolution GAN [3], conditional GAN [4], coupled GAN [5], cycle GAN [6] and many more generative models [7] based on the original GAN idea have been published in recent years. These generative models are actively used in multimedia research because of the capabilities for generating images, sounds, texts and videos from noise or conditional inputs. Most of the GAN architectures follow the same set of logical training procedures, generative and adversarial network architectures and closely related optimization procedures. Researchers are wasting valuable time to implement the same

logical flow of GANs over and over again implementing already available GAN architectures from scratch. In addition, they are facing problems in organizing deep learning (DL) experiments and experiment data.

In this context, our GANEx tool is a solution to perform GAN based research more effectively and efficiently. This tool is a complete pipeline for training, inference and analysis of generative models for saving the time of researchers and saving valuable data of experiments. The GANEx tool is enriched with real-time training analysing tools to support researchers to get early-stage decisions such as stopping the unstable training processes, detecting the unstable hyperparameters and other decisions which are more important to take early before starting the long training process of DL. Moreover, this tool is capable to handle GAN experiments in structured way and perform advanced analysis in the inference stage of GAN experiments.

As depicted in Figure 1, our main GANEx tool consists of three components; 1) a graphical user interface (GUI), 2) a library called FastGAN and 3) a DL library (Pytorch). In this paper, we discuss only the first two sections (our contributions) because the last section is implemented using the well known DL library Pytorch where details are found in [8]. Based on this, the main contributions of the presented tools are:

- The FastGAN library, which is introduced to develop, fine-tune, perform experiments and analyse generative models or GANs efficiently and effectively.
- The GANEx GUI, which is introduced to perform GAN experiments in a structured way and benchmarking them quickly for saving valuable time and experiment results such as parameters of GAN models, output data and other analysed statistical data.

In the next section (section 2), we discuss the available GUI based tools for running DL experiments. Section 3 covers the
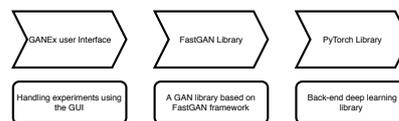
Fig. 1: Overview of the GANEx execution flow

concept and architecture of the FastGAN library before we discuss the GANEx GUI in section 4. We present some ideas of how GANEx can be expanded in section 5 and finally, in section 6, we give a description of the proposed demo.

II. RELATED WORK

NVIDIA DIGITS [9] is one of the popular tools among researchers for running and analysing DL experiments in computer vision problems such as classification, detection and segmentation. Recently they have experimented how to run basic GAN experiments using this tool and they have added basic GAN training capabilities. However, NVIDIA DIGITS capabilities of the current version are not enough for analysing and performing advanced GAN experiments because it has general purpose DL capabilities and it does not have mechanisms to mange experiments and doing advanced statistical analysis. Moreover, this kind of web-based solutions are slower than stand-alone applications because it has limitations to work directly with OS functionalities. Therefore, our solution is designed using stand-alone application development concepts to keep good performance and reliability.

GAN lab [10] is another web-based tool for understanding the main GAN architecture. Users can play with simple data generation problems using this tool. It visualizes how the GAN changes the input noise to the target distribution. They clearly emphasize gradient changes of model parameters using visualization to give a deeper understanding of the GAN architecture. This web-page was designed using the Tensorflow Javascript [11] library, and they have not targeted any researchers who are doing new advanced generative model experiments.

Weka [12] is a popular tool among machine learning researchers from beginners to advanced users. However, handling DL experiments using the Weka tool is limited. Because it mainly targets statistical machine learning algorithms, it has restrictions for doing advanced DL experiments using the popular DL frameworks such as Pytorch and TensorFlow. But, the concepts for managing experiments underpinning the Weka tool are helpful for developers of GUI-based experiment tools. They use standalone application development methods with Java for maintaining reliability of the tool with OS interactions.

The TensorBoard [13] visualization tool comes with the Tensorflow [14] DL library, and it is rich with more visualization tools for model parameters and training process. This tool is also powered by web technologies. Moreover, this tool can be used to visualize the main components of any DL model. However, GAN-specific statistical analysis is more difficult with this tool as a result of a generalized tool with Tesorflow. Structured experiment handling and data handling capabilities have not been designed in this solution.

Deep learning studio [15] is another tool which is closer to the concept of NVIDIA DIGITS. This tool has more capabilities in designing deep neural networks compared to the designing capabilities of NVIDIA DIGITS tool. In contrast to these benefits, Deep learning studio suffers from a lack of capabilities for visualization of model learning patterns and reasoning of inference. However, this tool can be identified as a tool including all the steps like designing a model, training a model and inference from a pre-trained model. For example, an autoencoder [16] which is one of the generative models is demonstrated using this tool. The autoencoder model can be designed easily using this tool because the input of this type of model is an image while the output is also an image. In contrast to autoencoder experiments, GAN experiments' input data depend on latent spaces, images as well as labels while they generate several output data such as images and labels. Some generative models generate indirect inference parameters like standard deviation and mean of probability distributions. Therefore, statistical data analysing and parameter handling are more important for GAN experiments.

Deep learning studio and all other tools discussed above are lacking of GAN specific modifications, statistical analysis tools and developments procedures. Therefore, we address these issues in our proposed solution, the GANEx tool which is a complete GAN specific pipeline for training, inference and doing advanced statistical analysis.

III. THE FASTGAN LIBRARY

The FastGAN implementation can be categorized as a core library because it opens paths for developers to define there own GAN experimental tools using other DL frameworks. The FastGAN library implementation is the core of the GANEx GUI tool. This library consists of high-end abstract logical flow which can be used to implement GAN based experiments very easily and quickly in a few steps. The main structure of the library is depicted in Figure 2.

The FastGAN library can be defined on top of available DL libraries such as Tensorflow, Pytorch, Microsoft Cognitive Toolkit or any other DL libraries available today. However, our first FastGAN library implementation is accomplished using the Pytorch library. This back-end dependency is depicted at the bottom of Figure 2. Visualization and 2D/3D plotting tools have to be provided alongside the main DL library for advanced statistical analysis of GAN experiments. The first FastGAN library uses "pyqtgraphs" [17] for this purpose because of the enrichment of plotting tools which are based on statistical and engineering applications.

The second level of this library consists of three main sections; FastGAN Nets, FastGAN Trainer and FastGAN Analyser. The FastGAN Nets should be packed with state of the art generative networks, discriminative networks, decoder and encoder networks which are used in generative model implementations. The FastGAN Trainer defines the logical training flows of generative models. It should have all possible training mechanisms of generative model training and adversarial generative model training. These mechanisms can be implemented as methods which are applicable for all generative models. For example, a training discriminator with real labels, a training discriminator with fake labels and a training generator with fake labels as real labels, can be identified as sub-components of the main training of basic
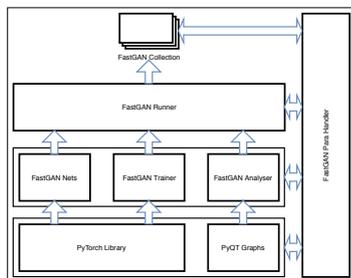
Fig. 2: The FastGAN library structure

Fig. 3: GUI and JSON flow of GANEx

GAN architectures. The FastGAN Analyzer is the next main module at the same level. This analyzer consists of various analysis tools for generative models. These analysis tools can include metrics for generative model comparison, plots of training and inference stages, parameter representations, input-output analysis, probability distribution representations and many more engineering and statistical analysis algorithms that are commonly used for generative models. The FastGAN analyzer can be implemented as an independent module which can be used for statistical analysis of any DL architecture.

The third level of the FastGAN library is named FastGAN Runner which defines how to execute training, re-training and inference of generative models based on the components of the bottom levels. This level includes data pre-processing mechanisms, parameters initialization methods for network parameters and noises, initializing analysis mechanisms of a specific GAN and interconnecting routings of all the components of the bottom levels.

The top level of the FastGAN library can be implemented by collecting implementations of all the state of the art generative models using the components of the bottom levels. Then, we can allow researchers to define their own parameters and input data without considering developments of generative models. If researchers are interested in more advanced modifications, then they can go through the levels of FastGAN from top to bottom as they required. This allows researchers to do GAN experiments from simple modifications to more advanced modifications.

The last component, the FastGAN Parahandler or parameter flow, is defined using the JSON library and dictionary data structures of Python in our test implementation. However, researchers or developers can use any mechanism such as a relational database to handle parameters and store parameters via all the levels.

IV. THE GANEx GUI TOOL

The GANEx GUI implementation is developed on top of the FastGAN library, and it enables a structured way for researchers to train, re-train, save, analyse and manage experiments and experiment data. The GUI of this tool is designed using PyQT5 library which is based on the well known Qt [18] project. In this GANEx tool, we designed a mechanism to save all experimental details and data automatically to avoid losing them as a result of forgetting to save. In contrast to this, if the researcher wants to delete experiments or projects, they can use the delete option of our tool compared to the saving options of general GUI tools. The main motivation for this concept is that every experimental details may have valuable point in future and therefore it should be a reference or logged records to easily track previous experiments. The main GUI flow of our GANEx is depicted in Figure 3. This flow is defined from the top to the bottom; the top level is the starting point of the tool while the bottom level represents endpoints of the GUI flow. Four boxes in the bottom layer shows different user interfaces for configuring and visualizing experiments.
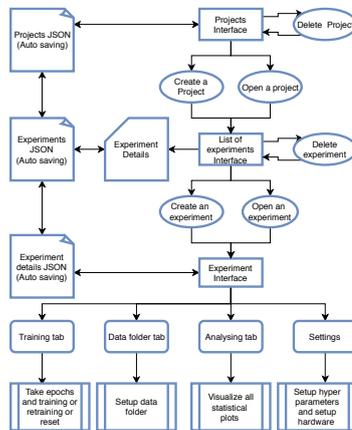
The top level of GANEx creates projects which can be consist of several GAN experiments. These experiments may have any type of FastGAN implementations. Then, GANEx enables users to organize their experiments in a structured way. As depicted in the left side of Figure 3, the GANEx tool uses its own JSON recorder to record project details.

The next main window is "list of experiments interface" which allows users to create different experiments based on different type of generative models. This window summarizes all the details about previous experiments and if researchers want it is possible to continue to the previous experiments. The user can create a new experiment by selecting a generative model type within the wizard window. Then, the GANEx GUI enables the main experiment window which has all the functionalities to control a specific GAN experiment. This experiments window is capable of handling several experiments of different types of generative models at the same time. Therefore, doing comparative generative model-based experiments are straightforward. The experiments JSON file organizes all details related to experiments of the current project.

(a) Real-time plotting

(b) Advanced analysis of generated images
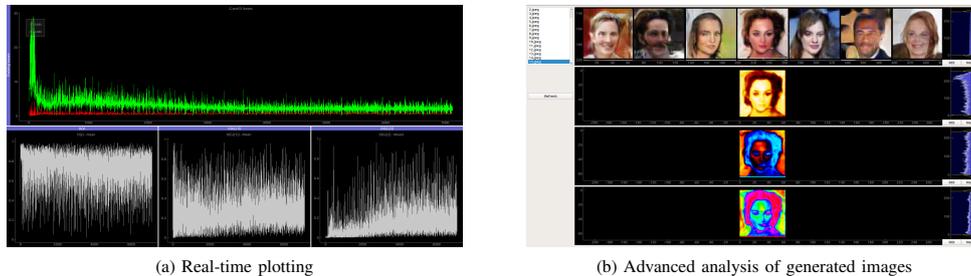
Fig. 4: Sample screenshots of the GANEx tool; (a) - this screen shows real-time loss value plots of a GAN architecture to monitor the failures of GAN training process, (b) - in this window, the user can use different heat maps and color histograms to understand the generated outputs

The current implementation of the main experiment window has capabilities to train a generative model, handle input data sources, set or tune hyperparameters of generative models and analyze generative models by visualizing input data, generated data, training and re-training behaviour and many more statistical and engineering analysing mechanisms. Example screen shots of real application windows and plots are presented in Figure 4.

V. EXPANDABLE GANEX

This GANEx implementation opens doors to a wide range of directions for expansions. In the future, the GUI-based GANEx tool can be improved to design complex GAN architectures from scratch using drag and drop components while implementing training and inference via GUI-based flow diagrams. In adddition, the analysis functionalities can be expanded based on the state of the art findings without affecting them with the base implementation because we keep the analysing part as an independent section. Moreover, our tool can be upgraded with hardware resources monitoring for researchers who are dealing with performance improvements. Furthermore, using the concepts behind this tool, GANEx shows direction to implement advanced tools for other DL mechanisms also.

VI. DEMO

In this demo, participants can get hands-on knowledge of GANEx, and they will experience the power of the tool. They will be able to get an idea about how GANEx organize experiments and experimental details. In this session, users can train a pre-designed simple GAN model from scratch using simple datasets like MNIST (handwritten digits) [19] and CelebA (low resolution celebrity images) [20]. Then, they can analyse the training process in real-time and generated images using the analysis window of GANEx.

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[2] I. J. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *Computing Research Repository (CoRR)*, vol. abs/1701.00160, 2016.

[3] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.

[4] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *Computing Research Repository (CoRR)*, vol. abs/1411.1784, 2014.

[5] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 469–477.

[6] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2017.

[7] A. Hindupur. (2017) Deep hunt: The gan zoo. [Online]. Available: https://deephunt.in/the-gan-zoo-79597dc8c347

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[9] L. Yeager, J. Bernauer, A. Gray, and M. Houston, "Digits: the deep learning gpu training system," in *Proceeding of Automation of Machine learning (AutoML)*, 2015.

[10] M. Kahng, N. Thorat, D. H. P. Chau, F. B. Viégas, and M. Wattenberg, "Gan lab: Understanding complex deep generative models using interactive visual experimentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 310–320, 2019.

[11] A. Paler, "Surfbraid: A concept tool for preparing and resource estimating quantum circuits protected by the surface code," *arXiv preprint arXiv:1902.02417*, 2019.

[12] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*, 4th ed. Morgan Kaufmann, 2016.

[13] G. LLC. (2019) Tensorboard. [Online]. Available: https://www.tensorflow.org/guide/summaries_and_tensorboard

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[15] I. Deep Cognition. (2018) Deep learning studio. [Online]. Available: https://deepcognition.ai/features/deep-learning-studio/

[16] P. Baldi, "Autoencoders, unsupervised learning and deep architectures," in *Proceedings of the International Conference on Unsupervised and Transfer Learning Workshop (UTL)*. JMLR.org, 2011, pp. 37–50.

[17] L. Campagnola. (2011) Pyqt graphs. [Online]. Available: http://www.pyqtgraph.org/

[18] Q. G. N. H. QTCOM). (2019) Qt. [Online]. Available: https://www.qt.io/

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.

[20] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

A.23 Paper XXIII - Vid2Pix - A Framework for Generating High-Quality Synthetic Videos

Authors: Oda O. Nedrejord, **Vajira Thambawita**, Steven A. Hicks, Pål Halvorsen, Michael A. Riegler

Abstract: Data is arguably the most important resource today as it fuels the algorithms powering services we use every day. However, in fields like medicine, publicly available datasets are few, and labeling medical datasets require tedious efforts from trained specialists. Generated synthetic data can be to future successful healthcare clinical intelligence. Here, we present a GAN-based video generator demonstrating promising results.

Published: In proceedings of IEEE International Symposium on Multimedia (ISM), 2020.

Candidate contributions: Vajira contributed to designing and implementing the theoretical models discussed in this paper. He contributed to evaluating the results (generated synthetic data) critically using dense optical flow calculations which can be used to identify temporal feature differences between frames in a video. Vajira also contributed to drafting and revising this manuscript.

Thesis objectives: Sub-objective IV

Vid2Pix - A Framework for Generating High-Quality Synthetic Videos

Oda O. Nedrejord*†, Vajira Thambawita*‡, Steven A. Hicks*‡, Pål Halvorsen*‡, and Michael A. Riegler*
*SimulaMet, Norway   †University of Oslo, Norway   ‡Oslo Metropolitan University, Norway

Abstract—**Data is arguably the most important resource today as it fuels the algorithms powering services we use every day. However, in fields like medicine, publicly available datasets are few, and labeling medical datasets require tedious efforts from trained specialists. Generated synthetic data can be to future successful healthcare clinical intelligence. Here, we present a GAN-based video generator demonstrating promising results.**

Index Terms—**Deep learning, generative adversarial networks, data up-sampling, video generation.**

I. INTRODUCTION

Data-driven technology has become ingrained in all areas of modern society, and healthcare is no exception. For example, machine learning-based systems have shown tremendous results in automatic detection of gastrointestinal (GI) anomalies for colonoscopies (e.g., [1], [2]). Despite these impressive results, these methods do not generalize well [3]. This is mostly due to a lack of training data as making medical data public is difficult, i.e., due to legal restrictions, patient privacy, and a manual time-consuming, tedious labelling task for trained medical experts.

Generated "fake" synthetic data can be the key to successful clinical and business intelligence [4], [5]. Therefore, in this paper, we present our Vid2Pix system that takes existing datasets and generates synthetic videos using a generative adversarial network (GAN). As an initial use-case, we use data collected from GI colonoscopies where anomalies are often missed and overlooked. We limit our scope to polyp videos, but the presented method should generalize well to other domains as well. The realism of the generated data is evaluated by two medical doctors, and quantitative measurements. The results suggest that the generated synthetic data is sometimes indistinguishable from real data and can, in the future, be used as training data for machine learning-based algorithms.

II. THE PROPOSED METHOD: VID2PIX

Using a dataset collected from two hospitals in Norway containing $83,088$ video frames, downsized to 128×128, we developed a system that can create more data from data we already have. Specifically, we aim to generate artificial videos of colon polyps by using real videos of colon polyps. Our system can be broken down into three distinct steps:

1) Skip Frames using Dense Optical Flow (step 1): A high frame rate combined with inconsistent camera movements causes inconsistencies in the videos. To address this problem, we first process the videos by using dense optical flow. Since the movement direction is not critical to solve our problem, we only consider the magnitude of the motion to decide whether to keep or to skip a frame using a threshold of 20% above the average magnitude between each continuous frame in a video. We create each video with a fixed length of 8 frames. If the difference in frame numbers are larger than 10 frames, we create a new video to avoid large jumps in the videos. With the method, we managed to optimize the dataset by removing duplicate frames and large jumps between frames.

2) Future Frame Generation with Vid2Pix (step 2): Our proposed architecture is a conditional GAN [6] that uses a generator and discriminator based on Pix2Pix [7]. Pix2Pix was developed to translate an *image* in one domain to an image in another domain. However, we are trying to learn past *image sequences* (videos) in one domain to generate future sequences in the same domain. Thus, our Vid2Pix system is a generative model that predicts a future frame conditioned on the past frames in a sequence.

We first add an additional dimension in order to use the temporal dimension to generate realistic motions. The additional dimension leads to a replacement of 2D with 3D convolutions and deconvolutions. The 3D convolutions extract features from the temporal dimension as well as the spatial dimension. Instead of using 2D convolutions as Pix2Pix does for down- and upsampling, we use 3D convolutions for both operations (to ensure support features). We use the additional dimension to input temporal information by stacking frames through that dimension. The height, width, and channels dimensions are used to input spatial features of input frames. The discriminator outputs a downsampled feature map from either a concatenation of the input sequence and a generated image or from a concatenation of the input sequence and the ground truth. The discriminator is a PatchGAN [7] .

3) Pipeline of predicting frame sequence (step 3): In order to generate a video in Vid2Pix, we need to iterate over the model with shifted input several times. Figure 1 shows how we generate a video from generated images. The Vid2pix model generates one image at a time as depicted in Figure 2.

III. QUALITY EVALUATION OF GENERATED VIDEOS

As an initial step in evaluating our GAN-based system, we generated videos consisting of four frames and calculated a dense optical flow visualization between each frame (Figure 3). An **initial inspection** suggests that they look realistic, and the dense optical flow proves that the model also learned to capture the correct movement of the videos.
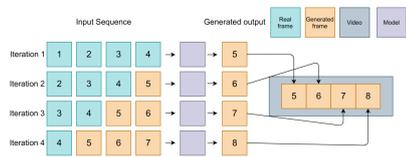
Appendix A. Published Articles

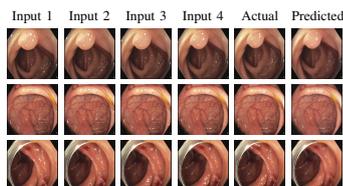Fig. 1: Frame pipeline to create a video from Vid2Pix.

Fig. 2: Videos with four input frames representing the stacked input to the model, the ground truth and the predicted output.
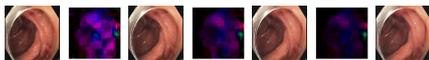

Fig. 3: Generated four frames and the corresponding dense optical flow between consecutive frames. Hue values represent the direction and the amount of motion.

Furthermore, to assess the realism of the generated content, we conducted a **subjective human assessment** to evaluate the generated videos. We recruited two medical doctors with endoscopy data experience, i.e., a medical doctor with two years of experience, and a gastroenterologist with extensive experience. The assessment is divided into two sessions, where we for each provided detailed information. The *classification* session involves classifying videos into two classes, either *real* or *fake*, where ten were artificial, and ten were real. Our results show that a total of six real videos were miss-classified as fake, and six videos were miss-classified as real when they were fake. Moreover, in a *grading* assessment, the reviewers assessed 31 fake videos. For each video, they were asked to give scores from one to five where one is least real and five is most real. Figures 4a and 4b show the grading results. The average grade from the first reviewer (the junior doctor) is 3.4, and the average grade from the second reviewer (the senior doctor) is 2.8. Overall, the doctors found many examples where it was hard to differentiate between real and fake as the shapes and colors appeared realistic, but the differences indicate that there is room for improvement, especially since the participants found some examples of strange motions and tissues.

Finally, we assessed the system using objective **similarity measures**. Using the generated frame and the corresponding ground truth on all generated videos we calculated the mean square error (MSE), peak signal to noise ratio (PSNR) and structural similarity (SSIM) values. Using 626 videos, we achieved respective PSNR, MSE, and SSIM averages
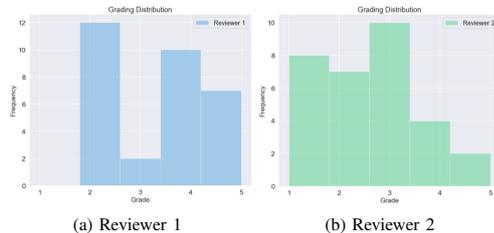
Fig. 4: Subjective grading distributions of 31 generated videos.

of 72.1301, 0.0050, and 0.8011 for Pix2Pix, and 73.3718, 0.0042, and 0.8409 for Vid2Pix. From these numbers, we observe that when we modify the original Pix2Pix model by using our intermediate experiments, such as predicting four frames at once, the SSIM and PSNR values first decrease, and MSE increases. Finally, SSIM shows good values for our last model modification, where we changed the model to predict one image instead of a sequence of images and reduced the discriminator complexity. We conclude that reducing the discriminator complexity and changing the output dimension has a positive effect on the quality of generated output.

IV. CONCLUSION

We have developed a conditional GAN to generate "fake" synthetic future frames using real videos as input. The key parts of the model were the 3D convolutional and deconvolutional layers creating realistic-looking spatio-temporal features. Moreover, to improve quality, we implemented a dense optical flow-based preprocessing framework, which could filter away stationary frames of a video. From our quantitative measurements, the MSE, PSNR, and SSIM metrics show that the Vid2Pix model outperforms the Pix2Pix model for artificial video generation. We also found that experienced medical doctors struggle to differentiate between real and synthetic videos, which indicates that synthetic videos look real. Still, there is a large room for improvement, and we currently work on model enhancements and trying different use-cases.

REFERENCES

[1] G. Urban *et al.*, "Deep Learning Localizes and Identifies Polyps in Real Time With 96% Accuracy in Screening Colonoscopy," *Gastroenterology*, vol. 155, no. 4, pp. 1069–1078.e8, Oct. 2018.
[2] M. Riegler *et al.*, "EIR - Efficient computer aided diagnosis framework for gastrointestinal endoscopies," in *Proc. of CBMI*, Jun. 2016, pp. 1–6.
[3] V. Thambawita *et al.*, "An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification," *ACM Trans. Comput. Healthcare*, vol. 1, no. 3, Jun. 2020.
[4] A. Rebhan, "How - and why - health organizations are using synthetic health care data," Advisory Board, Nov. 2019. [Online]. Available: https://www.advisory.com/research/health-care-it-advisor/it-forefront/2019/11/synthetic-health-data
[5] B. Siwicki, "Is synthetic data the key to healthcare clinical and business intelligence?" Healthcare IT News, Feb. 2020. [Online]. Available: https://www.healthcareitnews.com/news/synthetic-data-key-healthcare-clinical-and-business-intelligence
[6] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *CoRR*, vol. abs/1411.1784, 2014.
[7] P. Isola *et al.*, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proc. of CVPR*, Jul. 2017, pp. 5967–5976.

A.24 Paper XXIV - DeepFake Electrocardiograms: the Beginning of the End for Privacy Issues in Medicine

Authors: **Vajira Thambawita**, Jonas L. Isaksen, Steven A. Hicks, Jonas Ghouse, Gustav Ahlberg, Allan Linneberg, Niels Grarup, Christina Ellervik, Morten Salling Olesen, Torben Hansen, Claus Graff, Niels-Henrik Holstein-Rathlou, Inga Strümke, Hugo L. Hammer, Molly Maleckar, Pål Halvorsen, Michael A. Riegler, Jørgen K. Kanters

Abstract: Recent global developments underscore the prominent role big data have in modern medical science. Privacy issues are a prevalent problem for collecting and sharing data between researchers. Synthetic data generated to represent real data carrying similar information and distribution may alleviate the privacy issue. In this study, we present generative adversarial networks (GANs) capable of generating realistic synthetic DeepFake 12-lead 10-sec electrocardiograms (ECGs). We have developed and compare two methods, WaveGAN* and Pulse2Pulse GAN. We trained the GANs with 7,233 real normal ECG to produce 121,977 DeepFake normal ECGs. By verifying the ECGs using a commercial ECG interpretation program (MUSE 12SL, GE Healthcare), we demonstrate that the Pulse2Pulse GAN was superior to the WaveGAN to produce realistic ECGs. ECG intervals and amplitudes were similar between the DeepFake and real ECGs. These synthetic ECGs are fully anonymous and cannot be referred to any individual, hence they may be used freely. The synthetic dataset will be available as open access for researchers at OSF.io and the DeepFake generator available at the Python Package Index (PyPI) for generating synthetic ECGs. In conclusion, we were able to generate realistic synthetic ECGs using adversarial neural networks on normal ECGs from two population studies, i.e., there by addressing the relevant privacy issues in medical datasets.

Published: Submitted for publication, Preprint is available at medRxiv.

Candidate contributions: Vajira contributed to the conception and design of the deep-fake ECG generation study. He implemented a novel GAN architecture named Pulse2pulse that can generate realistic synthetic ECGs with the properties of real

"Normal" ECGs. Vajira conducted all GAN experiments and evaluated using MUSE reports (ECG evaluation reports generated from a real system using in hospitals). Vajira published his work on GitHub to make it reproducible for other ECG datasets. He generated and published the largest synthetic ECG dataset (around 120,000 ECGs) as a replacement to a restricted real ECG dataset. He contributed to drafting the manuscript and revising it.

Thesis objectives: Sub-objective II, Sub-objective IV

335

DeepFake electrocardiograms: the beginning of the end for privacy issues in medicine

Vajira Thambawita[1,2,+], Jonas L. Isaksen[3,+], Steven A. Hicks[1,2], Jonas Ghouse[3], GustavAhlberg[3], Allan Linneberg[3,4], Niels Grarup[3], Christina Ellervik[3], Morten Salling Olesen[3], Torben Hansen[3], Claus Graff[5], Niels-Henrik Holstein-Rathlou[6], Inga Strümke[1], Hugo L. Hammer[1,2], Molly Maleckar[1,2], Pål Halvorsen[1,2], Michael A. Riegler[1+*], and Jørgen K. Kanters[3+*]

[1]SimulaMet, Oslo 0167, Norway

[2]Oslo Metropolitan University, Oslo 0167, Norway

[3]University of Copenhagen, DK2200 Copenhagen N, Denmark

[4]Bispebjerg and Frederiksberg Hospital, DK2400 Copenhagen NV, Denmark

[5]Aalborg University, Aalborg, Denmark

[6]Novo-Nordisk Foundation, Copenhagen, Denmark

Corresponding Authors *jkanters@sund.ku.dk and michael@simula.no

+these authors contributed equally to this work

Appendix A. Published Articles

336

Summary

Big data is needed to implement personalized medicine, but privacy issues are a prevalent problem for collecting data and sharing them between researchers. A solution is synthetic data generated to represent real dataset carrying similar information.

Here, we present generative adversarial networks (GANs) capable of generating realistic synthetic DeepFake 12-lead 10-sec electrocardiograms (ECGs). We have developed and compare two methods, namely WaveGAN* and Pulse2Pulse GAN. We trained the GANs with 7,233 real normal ECG to produce 121,977 DeepFake normal ECGs. By verifying the ECGs using a commercial ECG interpretation program (MUSE 12SL, GE Healthcare), we demonstrate that the Pulse2Pulse GAN was superior to the WaveGAN to produce realistic ECGs. ECG intervals and amplitudes were similar between the DeepFake and real ECGs. These synthetic ECGs are fully anonymous and cannot be referred to any individual, hence they may be used freely. The synthetic dataset will be available as open access for researchers at OSF.io and the DeepFake generator available at the Python Package Index (PyPI) for generating synthetic ECGs.

In conclusion, we were able to generate realistic synthetic ECGs using adversarial neural networks on normal ECGs from two population studies, i.e., there by solving the relevant privacy issues in medical datasets.

336

Introduction

The use of artificial intelligence has increased in medicine over the past years. The overall aim of artificial intelligence in medicine is to aid clinicians based on complex data with decisions that are more accurate and improving personalized medicine. The prerequisite and foundation for artificial intelligence is the large amount of high-quality clinical data.

With updates of the General Data Protection Regulation (GDPR) regulative in the EU, the free flow of data has been restricted to ensure patient consent and anonymity[1]. Even anonymized deidentified data cannot be shared between research groups in different countries, because combining a few variables in an anonymized dataset, may allow for individual identification[2]. For example, knowing the zip code, birthday and sex is enough to identify 87% of US citizens[3]. On the other hand, large-scale, publicly available open-access medical datasets are required for personalized medicine to improve data-heavy machine learning solutions in medicine.

A solution to the privacy issue may be generation of synthetic realistic data. Synthetic data are data, which contain all the desired characteristics of a specific population, but without any sensitive content, making it impossible to identify individuals. Therefore, properly generated synthetic data are a solution to the privacy problem and will enable data sharing between research groups.

In this paper, we showcase synthetic electrocardiograms (ECG) as a complex example of medical data. An ECG is a voltage time series reflecting the electric currents within the heart, a widely used easy applicable and inexpensive clinical screening procedure to detect cardiac diseases. Using multiple electrodes, 3D propagation of cardiac electric impulses can be obtained and plotted as a standard 10-sec 12-lead ECG. Synthetic ECGs have been a topic of interest and research for many years. McSharry et al.[4] and Sayadi et al.[5] proposed mathematical dynamic models to generate continuous ECG signals, but these models were restricted to one

Appendix A. Published Articles

338

lead and did not reflect the distribution found in the normal population, nor did they give any insight in the mechanisms behind the disease.

Generative adversarial networks (GAN) were introduced in 2014 by Goodfellow et al. to generate synthetic data[6] using multi-layer perceptrons. Basically, a GAN consists of two neural networks: A generator network making signals (here ECGs) from random noise and a discriminator network evaluating whether the ECG is real or fake. During training, a mix of real ECGs and DeepFake ECGs are presented for the discriminator, which evaluates the ECGs assigning a score; high score for a likely real ECG, and low score for a supposed DeepFake ECG. As training proceeds, both the generator and discriminator improve until an equilibrium is reached[7]. Later, Radford et al.[8] developed a convolutional neural network GAN to generate synthetic images well suited for images like the ECG.

Since ECGs basically are time series, an initial approach was to use a WaveGAN[9] which is capable of generating sound signals. The classical WaveGAN is only able to output a single channel time series, so we modified the WaveGAN to generate all ECG channels (leads) (denoted WaveGAN*) instead of audio signals. We also introduce a novel DeepFake ECG U-net generative model, called Pulse2Pulse inspired by WaveGAN published by Donahue et al.[9] and compare our Pulse2Pulse generator to the WaveGAN generator.

In this paper, we present two GANs with the ability to generate an infinite number of 10-sec 12-leads synthetic "DeepFake" ECGs as a solution to overcome privacy issues related to real ECG data. These DeepFake ECGs can be openly distributed and freely downloaded as open access to be used by other scientists to develop ECG algorithms.

338

Results

We used ECGs from two population studies (GESUS[10] and Inter99[11]). To avoid chimeras between normal and abnormal ECG, we only trained the neural network with ECGs classified as normal by the MUSE 12SL. As shown in Table 1, both the WaveGAN* and Pulse2Pulse improved during training expressed as the percentage of DeepFake ECGs classified by the MUSE 12SLas normal ECGs. The Pulse2Pulse GAN trained faster than the WaveGAN* and had a better performance (expressed as fraction of ECGs classified as normal by the MUSE) than the WaveGAN* at their respective optimal number of training epochs (Table 1). Figure 1 shows a comparison of real and DeepFake ECGs, and the supplementary Figure S1 shows twenty randomly chosen DeepFake ECGs. Figure 2 shows the distribution of heart rates in the DeepFakes. By clinical definition Normal ECGs heart rates are between 60 and 99 beats per minute. The MUSE 12SL[12] classified 129 DeepFakes (0.5%) as sinus tachycardia (fast heart rate≥100) and 2863 (10.2%) as sinus bradycardia (slow heart rate<60). Figure 4 shows that cross correlation between as an example the QT interval and the RR interval were preserved. All cvariance structures can be seen in Supplementary Figure S2.

All DeepFake ECGs can be downloaded at OSF.io (https://osf.io/6hved/) with the corresponding ground truth parameters for the QT, RR, PR and QRS intervals and the P, STJ, R, and T amplitudes (see Figure 3 for ECG wave/interval naming terminology) delivered by the MUSE 12SL system (version 2.43). The DeepFake ECGs may be freely used for scientific use or commercial algorithm development if this paper is properly cited.)

Appendix A. Published Articles

340

Table 1. Quantitative difference between WaveGAN* and Pulse2Pulse GAN in the initial training for determining the optimal network and optimal number of epochs. The best values are bolded for each GAN.

	Fraction of ECGs classified as Normal (%)	
Checkpoint (epochs)	WaveGAN*	Pulse2Pulse
500	20.9	78. 7
1000	69.5	81.2
1500	71.2	78.8
2000	**72.5**	79.7
2500	71.3	**81.6**
3000	65.3	81.5

Using the Pulse2Pulse model from the optimal number of epochs (2500), we generated 150,000 DeepFake ECGs. To ensure, that these ECGs were realistic, we uploaded the 150,000 ECGs to the GE MUSE system and analyzed them using the 12SL algorithm. We found that 81.3% of the 150,000 DeepFake ECG were classified as "Normal ECG" (vs. 81.6 % in the initial training). Table 2 compares real vs. DeepFake ECGs using eight ECG properties (heart rate, P duration, QT interval, QRS duration, PR interval, STJ amplitude, R amplitude, and T amplitude extracted using MUSE 12SL. See Figure 3 for ECG nomenclature). The real data included all ECGs from GESUS and Inter99 classified as "Normal ECG" which were used for training. DeepFake ECGs are presented both as all 150.000 generated ECGs and the subset classified as Normal ECG. The supplementary Table S4 summaries the most common reasons for classifying DeepFake ECGs as Non-Normal ECGs.

340

Table 2. Mean, standard deviation (std), 2.5%, and 97.5% percentile for standard ECG parameters in real and fake ECGs. BPM = beats per minute

		Real – Normal (7,233)				Pulse2Pulse – Normal (121,977)				Pulse2Pulse – All (150,000)			
		Mean	std	2.5%	97.5%	Mean	std	2.5%	97.5%	Mean	std	2.5%	97.5%
Heart rate	BPM	70	8	60	90	70	7	60	88	70	8	60	89
P Duration	ms	105	12	82	130	117	17	86	152	118	17	84	152
QT Interval	ms	395	21	352	436	395	20	354	436	395	22	352	436
QRS Duration	ms	90	9	74	110	92	9	78	112	93	10	78	114
PR Interval	ms	156	19	120	198	158	17	126	192	159	19	124	194
STJ amplitude (V5)	µV	2	27	-44	58	18	33	-44	87	16	36	-54	87
R Amplitude (V5)	µV	1287	402	600	2163	1275	367	620	2026	1273	402	566	2094
T Amplitude (V5)	µV	343	137	126	664	366	135	156	668	361	141	141	673

A sample real ECG: A sample DeepFake ECG:
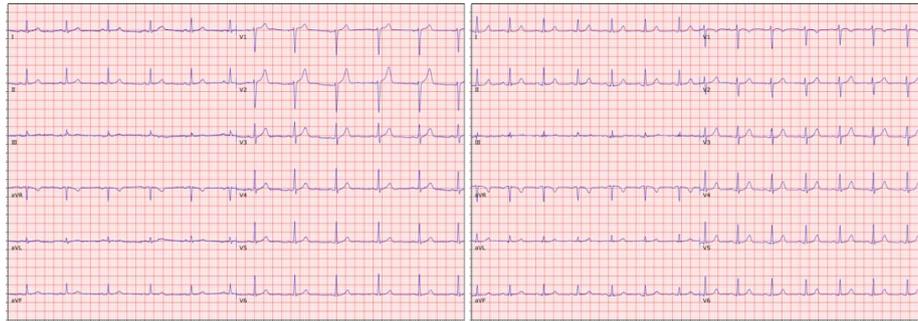
Figure 1. Comparison of examples of a real ECG (left lane) and a DeepFake ECG (right lane).

See supplementary Figure S1 for 20 more randomly chosen pairs of real and DeepFake ECGs.

Page 7 of 17

Appendix A. Published Articles

342

150K DeepFake ECG Distribution
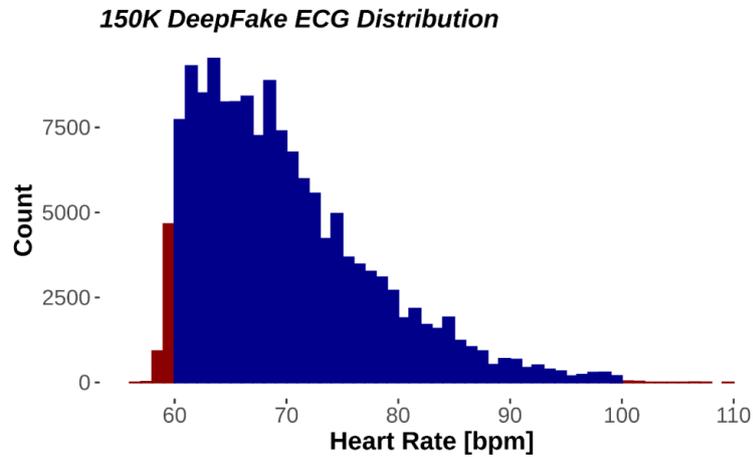
Figure 2. Distribution of heart rates in all 150.000 DeepFake electrocardiograms. Red fill denotes outside the normal heart rate range. Blue fill is within normal heart rate range (60-100).
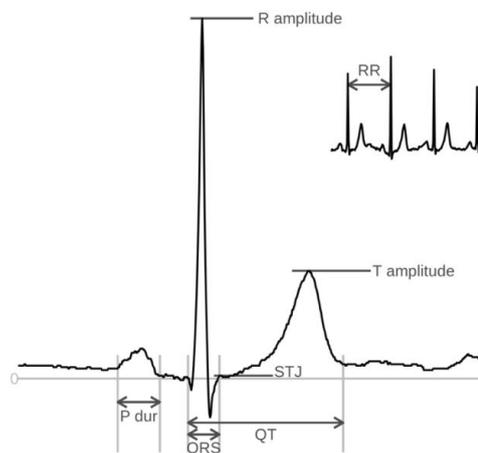
Figure 3. An ECG complex with the nomenclature of intervals (QT, QRS, P duration) and Amplitudes (STJ, R, T) and RR-interval (which can be converted to heart rate (HR) as HR=60/RR interval.

342

343
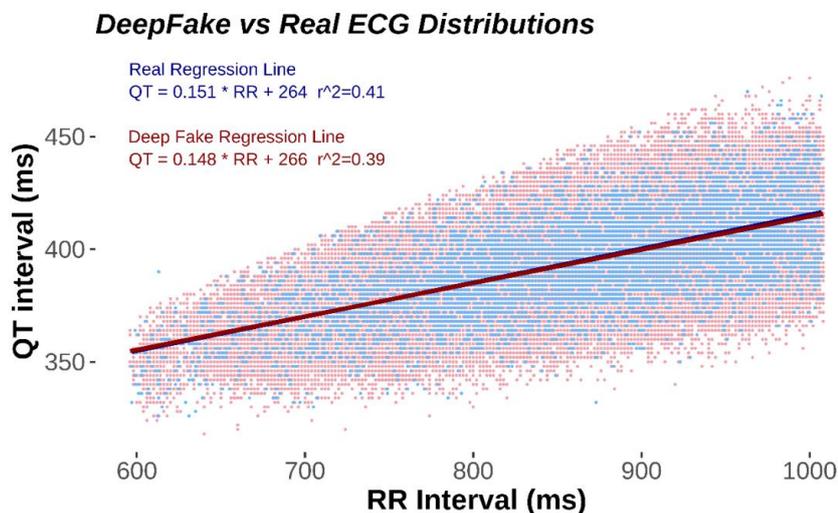
Figure 4. Scatter plot of the QT/RR interval relationship. Real ECG in blue and normal DeepFakes in red. DeepFake dots are nudged 1 ms to the left for visibility. Note that there are 121,977 normal DeepFakes and only 7,233 Real ECG making the DeepFake distribution more pronounced. As seen by the correlation coefficient r^2, the real and the fake DeepFake ECGs are similarly distributed.

Discussion

Although deep learning has been used for ECG analysis before[13,14], this study is the first study to generate realistic synthetic 10-sec 12-lead DeepFake ECGs. We demonstrate that the ECG characteristics from the real ECGs were similar to DeepFake ECGs. Hence, our DeepFake generator was able to construct synthesized ECG with similar intervals and amplitudes as the original population.

Nearly one fifth of the DeepFake ECGs were not recognized as Normal ECGs (Non-Normal) by the commercial MUSE 12SL ECG analyzer (No ECGs were rejected as being invalid).

Page 9 of 17

343

Appendix A. Published Articles

Many ECG parameters have hard boundaries in distinguishing between Normal and Non-Normal. For example, a normal heart rate is defined as between 60 to 99 bpm. Since we trained our model only on Normal ECGs, the input distribution for the GAN was a truncated asymmetric distribution. Thus, the clinically defined boundaries are skewed compared to the normal distribution of heart rates. The left truncation (at low heart rates) will discard more individuals than the right truncation (at high heart rates), and the final distribution of the real ECGs will be close to a truncated normal distribution with asymmetric truncations. The GAN will generally learn that heart rates outside 60-99 will not be valid, but small deviations will occur as seen in Figure 2 and Table 2. Since similar boundaries exist for many ECG parameters (for example PR interval >120 ms or QRS Interval<120 ms) sharp truncations would occur with several ECG parameters. This would lead to exclusion of some DeepFake ECGs, simply because the ECG intervals or amplitudes were just outside the normal range. Most ECG amplitudes and intervals were similar between real ECGs and DeepFake ECGs, but it was noteworthy that the STJ amplitude and the P duration had the greatest deviation between real ECGs and DeepFake ECGs. An explanation may be that both STJ and P amplitudes are small, and the network may tend to focus on larger waves such as the R and T waves. Following this theory, the network would to some extent neglect the smaller waves and features thereby introducing a larger uncertainty. Future networks may improve the ECG generation using conditional GANs to give more attention to smaller signal features. The Pulse2Pulse model was able to preserve the covariance structure between different ECG features, as seen in the most important relationship the QT/RR relationship which is known to have prognostic importance[15].

A challenging task is to define the optimal number of epochs for training. GANs tend to become unstable during the training process with the risk of the generator producing unrealistic output. To get an unbiased estimate how well the trained GAN performs, we used the commercial

345

MUSE 12SL system which automatically and reliably evaluates an ECG with a sensitivity of 99.9% and specificity of 100%[16]. Although the ECG discarded by the MUSE 12 SL may only have minimal abnormalities (like a heart rate of 59.9 where 60 is normal), the filtering of the DeepFake ECGs ensures, that the best epoch is chosen without bias, and the resulting ECGs are normal not only according to the discriminator, but also according to one of the most widely used ECG system in hospitals worldwide.

Personalized medicine will be dependent on big data, which demands international cooperation to ensure large datasets for researchers and the industry to work with. However, privacy and general data protection regulation rules are major obstacles for sharing data between researchers from different institutions and countries, or with the industry[17]. In conclusion, we show that we may overcome privacy and ethical[18] issues by constructing synthetic signals from real patients keeping the same clinical information as in the real dataset. The synthetic data generated by our Pulse2Pulse GAN makes it impossible to identify any patients, but still the ECGs remain useful for data scientists or industry to use for generating novel algorithms for ECG analysis. The approach is not limited to ECGs but should be expandable to all medical multichannel data, e.g., electroencephalography and electromyography. Therefore, the DeepFake ECGs generated from the Pulse2Pulse model can be used as a replacement to overcome the privacy constraints in real datasets.

Methods

The WaveGAN model is an evolution from the first GAN model introduced by Goodfellow et al.[6]. There are two deep neural networks named generator (G) and discriminator (D) to achieve the generation task from these GANs models. The main goal of the generator is to produce a data sample input (ECG(z)) from random noise (z) to the generator. The discriminator's task is to differentiate between real and fake data. We train the generator and discriminator together

Appendix A. Published Articles

346

as a competition (minmax game) between them. When a steady state is reached, the training halts, and the generator will generate realistic synthetic ECGs.

Data preparation

We have used two combined datasets: the Danish General Suburban Population Study[10] (GESUS) and the Inter99 study[11] (CT00289237, ClinicalTrials.gov). GESUS consists of 8,939 free-living subjects, and Inter99 consists of 6,667 free-living subjects with an available digital ECG. To avoid generation of hybrid ECG with mixed ECG abnormalities not occurring in real persons (e.g., to both be in sinus rhythm and atrial fibrillation at the same time which is impossible), we excluded ECGs who were not classified as normal (n=8,348) leaving 7,233 Normal ECGs for training.

A 12-lead 10-sec ECG consists only of 8 independent channels since 4 of the channels are simply trigonometric rotations. Therefore, the input ECG signal is 5,000x8 data points (corresponding to 10 sec with 500 samples per sec. x 8 channels). In addition to the up-scaling, we calculate the missing four channels with trigonometric functions to create the classic 12-channels ECG.

WaveGAN*: The input to WaveGAN* is a 1D random noise vector sampled from the uniform distribution (mean = 0, std = 1) with 100 x 1 passes through six deconvolution blocks to generate the desired output of 5000 x 8 samples. The deconvolution blocks are built from a series of four layers: an up-sampling layer, a constant padding layer, a 1D-convolution layer, and a ReLU activation function, consecutively. This implementation is deeper than the original architecture which use five deconvolution blocks used to generate synthetic music samples. Table S1 has comprehensive details of our WaveGAN* generator network.

Pulse2Pulse: The implementation of the Pulse2Pulse architecture (Figure 4) is inspired by the U-Net architecture[19] which is used for image segmentation. However, our Pulse2Pulse implementation is different than the original U-net implementation because the Pulse2Pulse implementation use 1D CNN for ECG signal generation rather than the 2D CNN used for original image segmentation task. The Pulse2Pulse network takes an 8×5000 noise vector which has the same dimension as the output dimension of a generated ECG. Then, we pass the noise through six down-sampling blocks followed by six up-sampling blocks as illustrated in Figure 3b. Each down-sampling block consists of a 1D-convolution layer followed by a Leaky ReLU activation. The up-sampling block is similar the deconvolution block used in WaveGAN*. In down-sampling, we have used Leaky ReLU instead of the ReLU layer used in the up-sampling to match the down-sampling operations to the discriminator. In addition to the up-sampling and down-sampling, the major modification is a bypass with down-sampling block features concatenating into the up-sampling block features represented by the black arrows in Figure 3b. To facilitate for this concatenation, we doubled the input size of up sampling blocks compared to WaveGAN* up sampling blocks. More details about Pulse2Pulse architecture are shown in the supplementary Table S1.

Discriminator: The same discriminator is used by WaveGAN* and Pulse2Pulse to discriminate between real and fake ECGs (Figure 3c). We used seven convolution layers (the original WaveGAN[9] has five layers), and each convolution layer is followed by a Leaky ReLU activation and the phase shuffle layer introduced in the original WaveGAN paper[9]. The discriminator takes an ECG as input (5000 samples * 8 channels) and outputs a score how close the ECG are to be determined fake or real.

Appendix A. Published Articles

348

(a) WaveGAN* generator.

(b) Pulse2pulse generator.

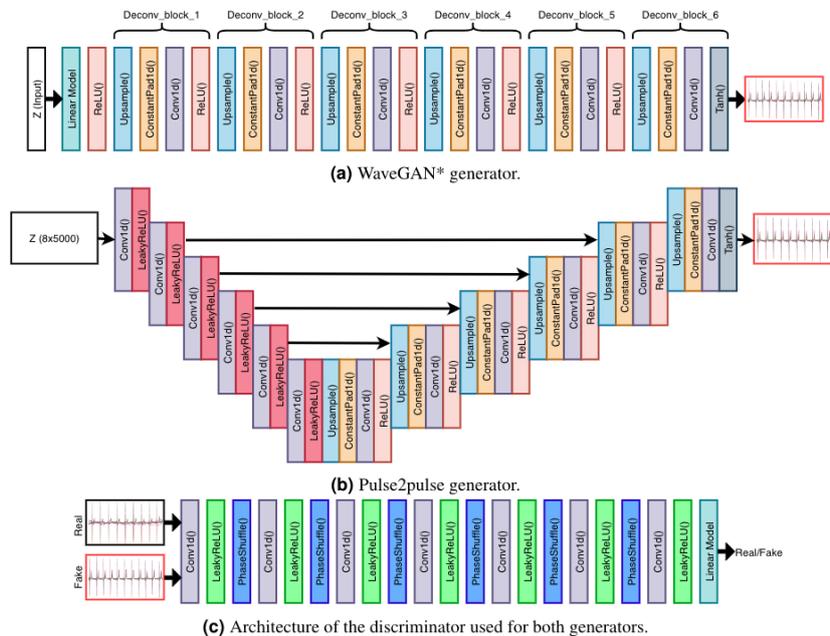(c) Architecture of the discriminator used for both generators.

Figure 4. Model architectures of the generators and the discriminator used to generate synthetic ECGs. WaveGAN* uses a 1D noise vector with 100 points. Pulse2Pulse uses a 2D noise vector with size of 8×5000 as input, same as the output ECG size.

Training: The models were trained on a Ubuntu workstation with a double Xeon processor and a GeForce NVIDIA RTX 2080 running the Pytorch deep learning framework[20]. We ran all our experiments (generators + discriminator) using the Adam[21] optimizer with a learning rate of 0.0001, β1value of 0.5, and β2 value of 0.9. As loss function, we used gradient clipping WGAN-GP[22], to ensure faster and better convergence. Similar to the audio generation paper of WaveGAN[9], we updated (backpropagated) the discriminator five times per update of the generator. We used a batch size of 32, which is half of the original batch size of 64 used in the original WaveGAN paper, as a result of using larger networks than the WaveGAN networks.

349

We kept the training process until 3000 epochs (~10 days computing time) because we experienced unstable training curves for both WaveGAN* and Pulse2Pulse afterwards.

DeepFake ECGs: For evaluation of our two GAN models, we initially generated 10,000 ECGs from every 500 epochs until 3000 epochs from each GAN model. The DeepFake ECGs were transferred to the MUSE system and evaluated by the MUSE 12SL algorithm v. 2.43[12] using the fraction of DeepFake ECGs described as Normal (similar to the Real ECGs used for training). Using the best epoch for the best GAN, we generated 150.000 DeepFake ECGs. These DeepFakes were similar evaluated by the MUSE 12SL.

Data Availability: The Normal DeepFake ECGs are available at OSF (https://osf.io/6hved/) with corresponding MUSE 12SL ground truth values freely downloadable and usable for ECG algorithm development. The DeepFake generative model is available at https://pypi.org/project/deepfake-ecg/ to generate only synthetic ECGs.

Code Availability: The complete source code of all networks discussed in paper are available at GitHub (https://github.com/vlbthambawita/deepfake-ecg).

References

1. Voigt, P. & von dem Bussche, A. *The EU General Data Protection Regulation (GDPR)*. (Springer International Publishing, 2017). doi:10.1007/978-3-319-57959-7.

2. de Montjoye, Y.-A., Radaelli, L., Singh, V. K. & Pentland, A. S. Identity and privacy. Unique in the shopping mall: on the reidentifiability of credit card metadata. *Science* **347**, 536–539 (2015).

3. El Emam, K., Jonker, E., Arbuckle, L. & Malin, B. A systematic review of re-identification attacks on health data. *PloS One* **6**, e28071 (2011).

Appendix A. Published Articles

350

4. McSharry, P. E., Clifford, G. D., Tarassenko, L. & Smith, L. A. A dynamical model for generating synthetic electrocardiogram signals. *IEEE Trans. Biomed. Eng.* **50**, 289–294 (2003).

5. Sayadi, O., Shamsollahi, M. B. & Clifford, G. D. Synthetic ECG generation and Bayesian filtering using a Gaussian wave-based dynamical model. *Physiol. Meas.* **31**, 1309–1329 (2010).

6. Goodfellow, I. *et al.* Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **27**, 9.

7. Osborne, M. J. & Rubinstein, a: *A Course in Game Theory*. (MIT Press, 1994).

8. Radford, A., Metz, L. & Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ArXiv151106434 Cs* (2016).

9. Donahue, C., McAuley, J. & Puckette, M. Adversarial Audio Synthesis. in *International Conference on Learning Representations* (2019).

10. Juhl, C. R., Miller, I. M., Jemec, G. B., Kanters, J. K. & Ellervik, C. Hidradenitis suppurativa and electrocardiographic changes: a cross-sectional population study. *Br J Dermatol* **178**, 222–228 (2018).

11. Ghouse, J. *et al.* Rare genetic variants previously associated with congenital forms of long QT syndrome have little or no effect on the QT interval. *Eur Heart J* **36**, 2523–2529 (2015).

12. GE Healthcare. Marquette™ 12SL™ ECG Analysis Program Physician's Guide 2056246-002 Revision C. (2015).

13. Attia, Z. I. *et al.* An artificial intelligence-enabled ECG algorithm for the identification of patients with atrial fibrillation during sinus rhythm: a retrospective analysis of outcome prediction. *Lancet* (2019) doi:10.1016/S0140-6736(19)31721-0.

14. Hicks, S. A. *et al.* Explaining Deep Neural Networks for Knowledge Discovery in Electrocardiogram Analysis. *MedRxiv* **2021.01.06.20248927**, (2021).

351

15. Jensen, B. T. *et al.* QT dynamics in risk stratification after myocardial infarction. *Heart Rhythm* **2**, (2005).

16. Froelicher, V. F., Marcus, R. & Heidenrich, P. Prognostic Value of Computer Electrocardiography in Veteran Outpatients. *Fed. Pract.* **21**, 11–20 (2004).

17. El Emam, K., Rodgers, S. & Malin, B. Anonymising and sharing individual patient data. *BMJ* **350**, h1139 (2015).

18. Ienca, M. *et al.* Considerations for ethics review of big data health research: A scoping review. *PloS One* **13**, e0204937 (2018).

19. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. in *International Conference on Medical image computing and computer-assisted intervention* 234–241 (Springer, 2015).

20. Paszke, A. *et al.* PyTorch: An Imperative Style, High-Performance Deep Learning Library. *ArXiv191201703 Cs Stat* (2019).

21. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs* (2017).

22. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. C. Improved training of Wasserstein GANs. *Adv. Neural Inf. Process. Syst.* **30**, 5767–5777 (2017).

A.25 Paper XXV - SinGAN-Seg: Synthetic Training Data Generation for Medical Image Segmentation

Authors: **Vajira Thambawita**, Pegah Salehi, Sajad Amouei Sheshkal, Steven A. Hicks, Hugo L. Hammer, Sravanthi Parasa, Thomas de Lange, Pål Halvorsen,Michael A. Riegler

Abstract: Processing medical data to find abnormalities is a time-consuming and costly task, requiring tremendous efforts from medical experts. Therefore, artificial intelligence (AI) has become a popular tool for the automatic processing of medical data, acting as a supportive tool for doctors. AI tools highly depend on data for training the models. However, there are several constraints to access to large amounts of medical data to train machine learning algorithms in the medical domain, e.g., due to privacy concerns and the costly, time-consuming medical data annotation process.

To address this, in this paper we present a novel synthetic data generation pipeline called SinGAN-Seg to produce synthetic medical data with the corresponding annotated ground truth masks. We show that these synthetic data generation pipelines can be used as an alternative to bypass privacy concerns and as an alternative way to produce artificial segmentation datasets with corresponding ground truth masks to avoid the tedious medical data annotation process. As a proof of concept, we used an open polyp segmentation dataset. By training UNet++ using both real polyp segmentation dataset and the corresponding synthetic dataset generated from the SinGAN-Seg pipeline, we show that the synthetic data can achieve a very close performance to the real data when the real segmentation datasets are large enough. In addition, we show that synthetic data generated from the SinGAN-Seg pipeline improving the performance of segmentation algorithms when the training dataset is very small. Since our SinGAN-Seg pipeline is applicable for any medical dataset, this pipeline can be used with any other segmentation datasets.

Candidate contributions: Vajira contributed to the conception and designing of this study. He developed the whole source code and tested the initial experiments. Moreover, he evaluated the performance of the model introduced in this paper critically by conducting several experiments. He has created and published the synthetic dataset, the corresponding generative models as a PyPI package, and the GitHub repository. Vajira also contributed to the drafting and revising of the article.

Thesis objectives: Sub-objectives I, Sub-objective II, Sub-objective III, Sub-objective IV

arXiv:2107.00471v1 [eess.IV] 29 Jun 2021

SinGAN-Seg: Synthetic Training Data Generation for Medical Image Segmentation

Vajira Thambawita[1,2], Pegah Salehi[1], Sajad Amouei Sheshkal[1], Steven A. Hicks[1,2], Hugo L. Hammer[2], Sravanthi Parasa[4], Thomas de Lange[3], Pål Halvorsen[1], and Michael A. Riegler[1]

[1]*SimulaMet, Oslo, Norway*
[2]*Oslo Metropolitan University, Oslo, Norway*
[3]*Department of Medical Research, Bærum Hospital, Gjettum, Norway*
[4]*Department of Gastroenterology, Swedish Medical Group, Seattle, WA, USA*

Abstract

Processing medical data to find abnormalities is a time-consuming and costly task, requiring tremendous efforts from medical experts. Therefore, artificial intelligence (AI) has become a popular tool for the automatic processing of medical data, acting as a supportive tool for doctors. AI tools highly depend on data for training the models. However, there are several constraints to access to large amounts of medical data to train machine learning algorithms in the medical domain, e.g., due to privacy concerns and the costly, time-consuming medical data annotation process.

To address this, in this paper we present a novel synthetic data generation pipeline called *SinGAN-Seg* to produce synthetic medical data with the corresponding annotated ground truth masks. We show that these synthetic data generation pipelines can be used as an alternative to bypass privacy concerns and as an alternative way to produce artificial segmentation datasets with corresponding ground truth masks to avoid the tedious medical data annotation process. As a proof of concept, we used an open polyp segmentation dataset. By training UNet++ using both real polyp segmentation dataset and the corresponding synthetic dataset generated from the SinGAN-Seg pipeline, we show that the synthetic data can achieve a very close per-

formance to the real data when the real segmentation datasets are large enough. In addition, we show that synthetic data generated from the SinGAN-Seg pipeline improving the performance of segmentation algorithms when the training dataset is very small. Since our SinGAN-Seg pipeline is applicable for any medical dataset, this pipeline can be used with any other segmentation datasets.

1 Introduction

AI has become a popular tool in medicine and has been vastly discussed in recent decades to augment performance of clinicians [1, 2, 3, 4]. According to the statistics discussed by Jiang et al. [1], artificial neural networks (ANNs) [5] and support vector machines (SVMs) [6] are the most popular machine learning (ML) algorithms used with medical data. These ML models learn from data; thus the medical data have a direct influence on the success of ML solutions in real applications. While the SVM algorithms are popular within regression [7, 8] and classification [9] tasks, ANNs or deep neural networks (DNNs) are used widely for all the types; regression, classification, detection and segmentation.

A segmentation model makes more advanced predictions than regression, classification, and detection as it performs pixel-wise classification of the input images. Therefore, medical image segmentation is

1

a popular application of AI in medicine, so it is used more widely with different kinds of medical image data [10, 11, 12]. Polyp segmentation is one of popular segmentation tasks that uses ML techniques to detect and segment polyps in images/videos collected from gastrointestinal tract (GI) screenings. Early identification of polyps in GI tract is critical to prevent colorectal cancers [13]. Therefore, many ML models have been investigated to segment polyps automatically in GI tract videos recorded from endoscopy [14, 15, 16] or PilCams examinations [17, 18, 19] to augment performance of doctors by detecting polyps missed by experts, thereby both decreasing the miss rates and reducing the observer variations.

Most of polyp segmentation models are based on convolutional neural networks (CNNs) and are trained using publicly available polyp segmentation datasets [20, 21, 22, 23, 24]. However, these datasets have a limited number of images with corresponding expert annotated masks. For examples, the CVC-VideoClinicDB [21] dataset has $11,954$ images from 10 polyp videos and 10 non-polyp videos, the PICCOLO dataset [24] has $3,433$ manually annotated images ($2,131$ white-light images and $1,302$ narrow-band images), and the Hyper-Kvasir [20] dataset has only $1,000$ segmented images, but also contains of $100,000$ unlabeled images.

We identified two main reasons for having small datasets in medical domain compared to other domains. The first reason is privacy concerns attached with medical data, and the second is the costly and time-consuming medical data annotation processes that the medical domain experts must perform.

The privacy concerns can vary from country to country and region to region according to data protection regulations introduced in the specific areas. For example, Norway should follow the rules given by the Norwegian data protection authority (NDPA) [25] and enforce the personal data act [26] in addition to following the general data protection regulation (GDPR) [27] guidelines being the same for all European countries. While there is no central level privacy protection guideline in the US like GDPR in Europe, US rules and regulations are enforced through other US privacy laws, such as

Health Insurance Portability and Accountability Act (HIPAA) [28] and California Consumer Privacy Act (CCPA) [29]. In Asian counties, they follow their own sets of rules, such as Japan's Act on Protection of Personal Information [30], the South Korean Personal Information Protection Commission [31] and the Personal Data Protection Bill in India [32].

If research is performed with such privacy restrictions, the papers published are often theoretical methods only. According to the analyzed medical image segmentation studies in [33], 30% have used private datasets. As a result, the studies are not reproducible. Researchers must keep datasets private due to medical data sharing restrictions. Furthermore, universities and research institutes that use medical domain data for teaching purposes use the same medical datasets for years, which affects the quality of education. In addition to the privacy concerns, the costly and time-consuming medical data labeling and annotation process [34] is an obstacle to producing big datasets for AI algorithms. Compared to other already time-consuming medical data labeling processes, a pixel-wise data annotation are far more demanding on the valuable medical experts' time. The experts in the medical domain can perform the annotations fully trustable in terms of correctness. If the data annotations by experts are not possible, the experts should do at least a review process to make the annotations correct before using them in AI algorithms. The importance of having accurate annotations from experts for medical data is, for example, discussed by Yu et al. [35] using a mandible segmentation dataset of CT images. In this regard, researching a way to produce synthetic segmentation datasets is important to overcome the timely and costly medical data annotation process. Therefore, researching an alternative way for medical data sharing, bypassing both the privacy and time-consuming dataset generation challenges, is the main objective of this study.

In this regard, the contributions of this paper are as follows.

- This study introduces the novel SynGAN-Seg pipeline to generate synthetic medical image and its corresponding segmentation mask using a modified version of the state-of-the-art SinGAN

2

architecture with a fine-tuning step using a style-transfer method. We use polyp segmentation as a case study, the SinGAN-Seg can be applied for all types of segmentation tasks.

- We have published the biggest synthetic polyp dataset and the corresponding masks at `https://osf.io/xrgz8/`. Moreover, we have published our generators as a python package at Python package index (PyPI) (`https://pypi.org/project/singan-seg-polyp/`) to generate an unlimited number of polyps and corresponding mask images as needed. To the best of our knowledge, this is the first publicly available synthetic polyp dataset and the corresponding generative functions as a PyPI package.

- We show that synthetic images and corresponding mask images can improve the segmentation performance when the size of a training dataset is limited.

2 Method

In the pipeline of SinGAN-Seg, there are as depicted in Figure 1 two main steps: (1) training novel SinGAN-Seg generative models and (2) style transferring. The first step generates synthetic polyp images and corresponding binary segmentation masks representing the polyp area. The novel four channels SinGAN-Seg, based on the vanilla SinGAN architecture [36], is introduced in this first step. The novel training process of four channels SinGAN-Seg models is presented in this step. Using a single SinGAN-Seg model, we can generate multiple synthetic images and masks from a single real image and the corresponding masks. Therefore this generation process can be identified as $1 : N$ generations, and it is denoted using $[img]_N$, where N represents the number of samples generated in the figure. The second step focuses on transferring styles such as features of polyps' texture from real images into the corresponding generated synthetic images. This second step is depicted in the Step 2 in Figure 1.

SinGAN-Seg is a modified version of SinGAN [36] which was designed to generate synthetic data from a generative adversarial network (GAN) trained only using a single image. The original SinGAN is trained using different scales of the same input image, the so-called image pyramid. This image pyramid is a set of images of different resolutions of a single image from low resolution to high resolution. SinGAN consists of a GAN pyramid, which takes the corresponding image pyramid. In this study, we build on the implementation and the training process used in SinGAN, except for the number of input and output channels. The original SinGAN implementation [36] uses a three-channel RGB image as the input and produces a three-channel RGB image as the output. However, our SinGAN-Seg uses four-channels images as the input and the output. The four-channels image consist of the input RGB image and the single channel ground truth mask by stacking them together as depicted in the SinGAN-Seg model in Figure 1. The main purpose of this modification is to generate four-channels synthetic output, which consists of a synthetic image and the corresponding ground truth mask.

In the second step of the SinGAN-Seg pipeline, we fine-tune the output of the four channels SinGAN-Seg model using the style-transfer method introduced by Leon et al. [37]. This step aims to improve the quality of the generated synthetic data by transferring realistic styles from real images to synthetic images. As depicted in Step 2 in Figure 1, every generated image G_M is enhanced by transferring style form the corresponding real image im_M. Then, the style transferred output image is presented using ST_M where $M = [0, 1, 2...999]$ in this study, representing the 1000 images in thr training dataset. In this process, a suitable *content* : *style* ratio should be found, and it is a hyper-parameter in this second stage. However, this step is a separate training step from the training step of the SinGAN-Seg generative models. Therefore, this step is optional to follow, but we strongly recommend this style-transferring step to enhance the quality of the output data from the first step.
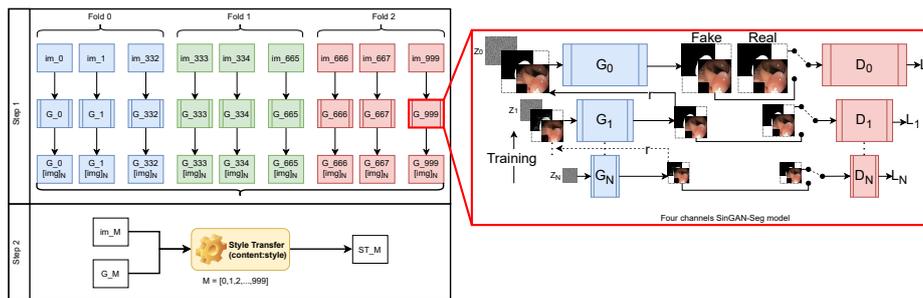
3

Figure 1: The complete pipeline of SinGAN-Seg to generate synthetic segmentation datasets. *Step 1*: represents the training of four channels SinGAN models. *Step 2*: represents fine tuning step using the neural style transfer [37]. *Four channels SinGAN*: Single training step of our four-channels SinGAN. Note the stacked input and output compared to the original SinGAN implementation [36] which input only single image with a noise vector and output only an image. In our SinGAN implementation, all the generators (from G_0 to G_{N-1}), except G_N, get four channels image (a polyp image and a ground truth) as the input in addition to the input noise vector. The first generator, G_N get only the noise vector as the input. The discriminators also get four channels images which consist of a RGB polyp image and a binary mask as input. The inputs to the discriminators can be either real or fake.

3   Experiments and results

This section demonstrates all the experiments and results collected using a polyp dataset as a case study. For all the experiments discussed in the following sections, we have used Pytorch deep learning framework [38].

3.1   Data

We have used a polyp dataset published with HyperKvasir dataset [20] which consists of polyp findings extracted from endoscopy examinations. This polyp dataset has 1000 polyp findings and a corresponding segmentation mask annotated by experts. We use only the polyp dataset as a case study because of the time and resource-consuming training process of the SinGAN-Seg pipeline. Furthermore, we use three-fold cross-validation, which is another time-consuming technique, for the experiments performed to find the validity of using synthetic data instead of real data.

A few sample images and the corresponding masks of the polyp dataset of HyperKvasir are depicted in Figure 2. The polyp images of the dataset are RGB images. The masks of the polyp images are single-channel images with white (255) for true pixels, which represent polyp regions, and black (0) for false pixels, which represent clean colon or background regions. In this dataset, there are different sizes of polyps. The distribution of polyp sizes as a percentage of the full image size is presented in the histogram plot in Figure 3. In this dataset, there are more relatively small polyps compared to larger polyps according to the plot presented in Figure 3. Additionally, this dataset was used to prove that the performance of segmentation models trained with small datasets can be improved using our SinGAN-Seg pipeline.

This dataset was used for two purposes.

1. To train SinGAN-Seg models to generate synthetic data.

2. To compare performance of real and synthetic

4

data for training segmentation ML models.

3.2 Training Generators

To use SinGAN-Seg to generate synthetic segmentation datasets to represent real segmentation datasets, we first trained SinGAN-Seg models one by one for each image in the training dataset. In our case study, there were 1000 polyp images and corresponding ground truth masks. Therefore, 1000 SinGAN-Seg models were trained. To train these SinGAN-Seg models, we have followed the same SinGAN settings used in the vanilla SynGAN paper [36]. Despite using the original training process, the input and output of SinGAN-Seg are four channels. After training each SinGAN-Seg by iterating 2000 epochs per scale of pyramidal GAN structure (see four channels SinGAN-Seg architecture in Figure 1 to understand this pyramidal GAN structure), we stored final checkpoints to generate synthetic data in the later stages from the each scale. The resolution of the training image of the SinGAN-Seg model is arbitrary because it depends on the size of the real polyp image. This input image is resized according to the pyramidal re-scaling structure introduced in the original implementation of SinGAN [36]. This rescaling pattern is depicted in the four channels SinGAN architecture in Figure 1. The re-scaling pattern used to train SinGAN-Seg models is used to change the randomness of synthetic data when pre-trained models are used to generate synthetic data. The models were trained on multiple computing nodes such as Google Colab with Tesla P100 16GB GPUs and a DGX-2 GPU server with 16 V100 GPUs because training 1000 GAN architectures one by one is a tremendous task. The average training time per SinGAN-Seg model was around 65 minutes.

After training SinGAN-Seg models, we have generated 10 random samples per real image using the input scale 0, which is the lowest scale that use a random noise input instead of a rescaled input image. For more details about these scaling numbers and corresponding output behaviors, please refer to the vanilla SinGAN paper [36]. Randomly selected three training images and the corresponding first 5

synthetic images generated using scale 0 are depicted in Figure 4. The first column of the figure represents the real images and the ground truth mask annotated from experts. The rest of the columns represents randomly generated synthetic images, and the corresponding generated mask.

In total, we have generated $10,000$ synthetic polyp images and the corresponding masks. SinGAN-Seg generates random samples with high variations when the input scale is 0. This variation can be easily recognized using the standard deviation (std) and the mean mask images presented in Figure 5. The mean and std images were calculated by stacking the 10 generated mask images corresponding to the 10 synthetic images related to a real image and calculating pixel-wise std and mean. Bright color in std images and dark color in mean images mean low variance of pixels. In contrast, dark color in std and bright color in mean images reflect high variance in pixel values. By investigating Figure 5, we can notice that small polyp masks have high variance compared to the large polyp mask as presented in the figure.

To understand the difference between the mask distribution of real images and synthetic images, we plotted pixel distribution of masks of synthetic $10,000$ images in Figure 6. This plot is comparable to the pixel distribution presented in Figure 3. The randomness of generations made differences in the distribution of true pixel percentages compared to the true pixel distribution of real masks of real images. However, the overall shape of synthetic data mask distribution shows a more or less similar distribution pattern to the real true pixel percentage distribution.

3.3 Style Transferring

After finishing the training of 1000 SinGAN-Seg models, the style transfer algorithm [37] was applied to every synthetic sample generated from SinGAN-Seg. In the style-transferring algorithm, we can change several parameters such as the number of epochs to transfer style from an image to another and the *content* : *style* weight ratio. This paper used a 1000 epoch to transfer style from a style image (real polyp image) to a content image (generated syn-
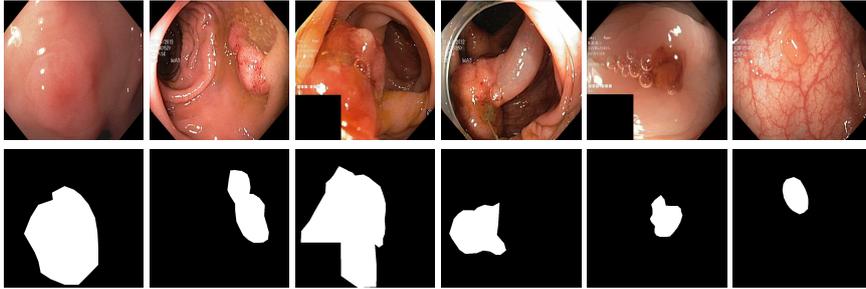
5

Figure 2: Sample images and corresponding masks from HyperKvasir [20] segmentation 1000 images.
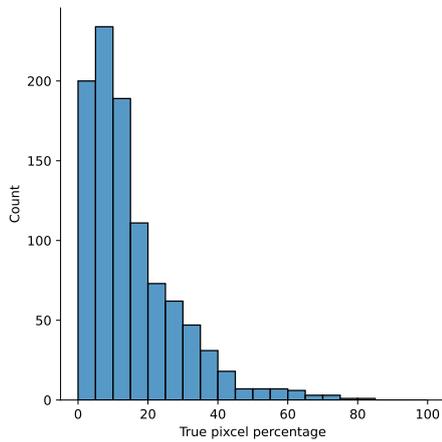

Figure 3: Distribution of true pixel percentages of the 1000 polyp masks of HyperKvasir [20] dataset.

thetic polyp). For performance comparisons, two $content : style$ ratios, 1 : 1 and 1 : 1000 were used. An NVIDIA GeForce RTX 3080 GPU took around 20 seconds to transfer style for a single image.

We have depicted visual comparison between pure generated synthetic images and style transferred images ($content : style = 1 : 1000$) in Figure 7. Samples with the style transfer ratio 1 : 1 are not depicted here because it is difficult to see the differences between 1 : 1 and 1 : 1000 visually. The first column of Figure 7 shows the real images used as content images to transfer styles. The rest of the images in the first row of each image shows synthetic images generated from SinGAN-Seg before applying the style transferring algorithm. Then, the second row of each image shows the style transferred synthetic images. Differences of the synthetic images before and after applying the style transfer method can be easily recognized from images of the second reference image (using 3rd and 4th rows in Figure 7).

3.4 Python package and synthetic data

Using all the pre-trained SinGAN-Seg checkpoints, we have published a PyPI package and the corresponding GitHub repository to make all the experiments reproducible. Additionally, we have published the first synthetic polyp dataset to demonstrate how to share synthetic data instead of a real dataset that may have privacy concerns. The synthetic dataset is available at `https://osf.io/xrgz8/`. Moreover, this is an example synthetic dataset generated using the SinGAN-Seg pipeline. Furthermore, this dataset is an example showing how to increase a segmentation dataset size without using the time-consuming and
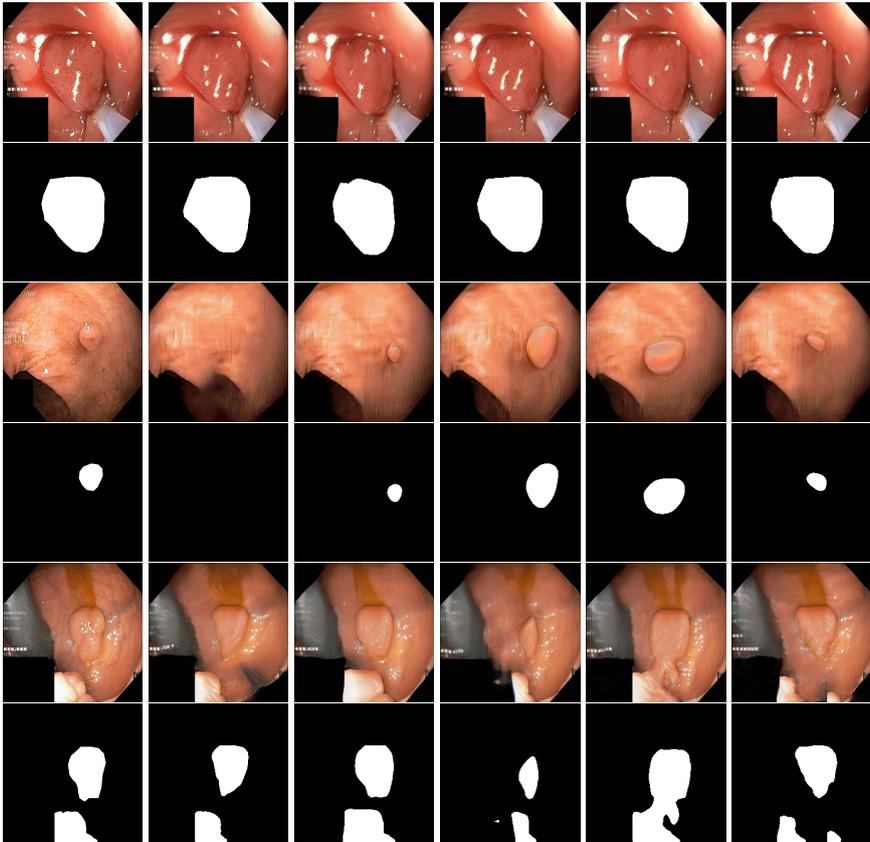
Figure 4: Sample real images and corresponding SinGAN generated synthetic GI-tract images with corresponding masks. The first column is illustrated with real images and masks. All other columns represent randomly generated synthetic data from SinGANs which were trained from the image on the first column.
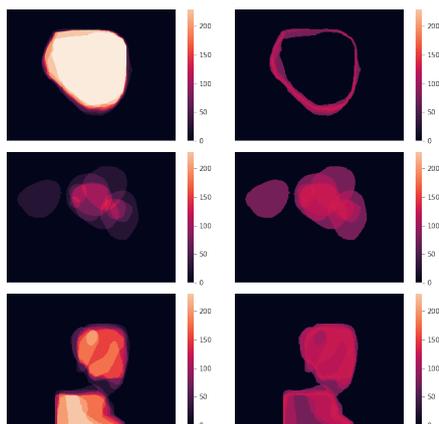
7

Figure 5: Mean and standard deviation calculated from 10 random mask generated from SinGAN-Seg. The corresponding real mask annotated from expertes can be seen in Figure 4.

costly medical data annotation process that needs experts' knowledge.

We named this PyPI package as `singan-seg-polyp` (pip install `singan-seg-polyp`) and it can be found here: `https://pypi.org/project/singan-seg-polyp/`. To the best of our knowledge, this is the only PyPI package to generate an unlimited number of synthetic polyps and corresponding masks. The corresponding GitHub repository is available at `https://github.com/vlbthambawita/singan-seg-polyp`. A set of functionalities were introduced in this package for end-users. Generative functions can generate random synthetic polyp data with their corresponding mask for a given image id from 1 to 1000 or for the given checkpoint directory, which is downloaded automatically when the generative functions are called. The style transfer function is in this package to transfer style from the real polyp images to the corresponding synthetic polyp images. In both functionalities, the relevant hyper-parameters can be

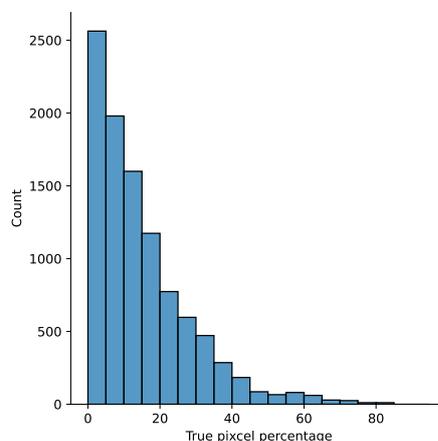Figure 6: Distribution of 10,000 masks of the synthetic generations. This 10,000 represent the 1000 real polyp images. From each real image, 10 synthetic samples were generated. The synthetic 10,000 dataset can be downloaded from `https://osf.io/xrgz8/`.
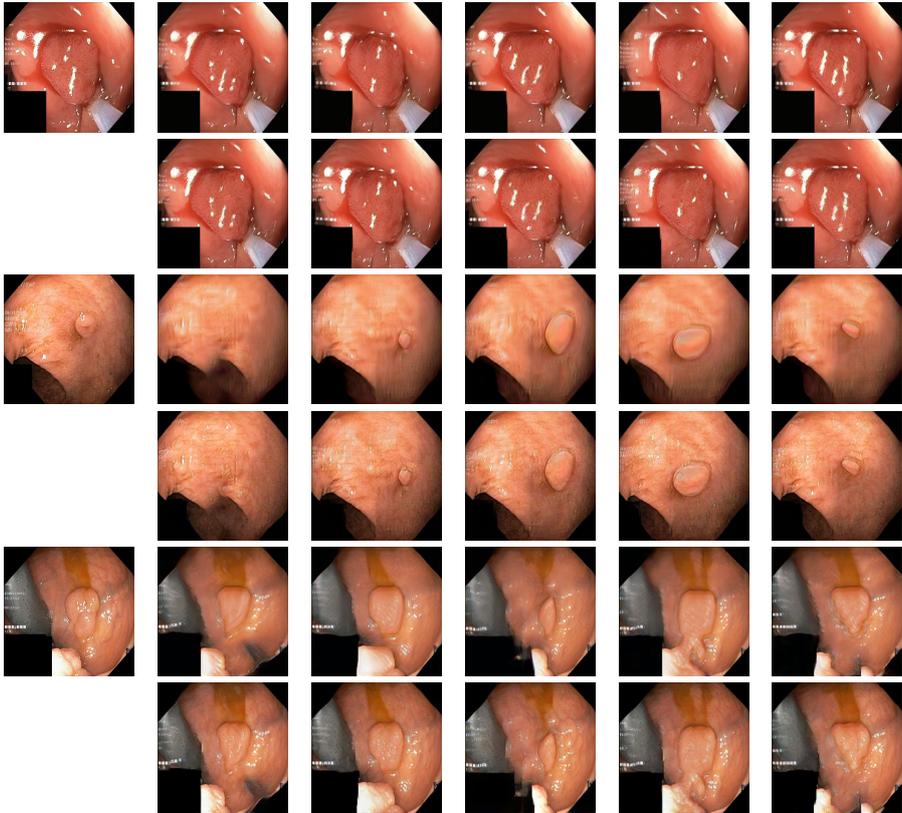
8

362

Figure 7: Direct generations of SinGAN-Seg versus Style transferred samples. The style transferring was performed using 1 : 1000 content to style ratio.

9

changed as needed to end-users of this PyPI package.

3.5 Baseline experiments

Two different sets of baseline experiments were performed for two different objectives. The first objective was to compare the quality of generated synthetic data over the real data. Using these baseline experiments, we can identify the capability of sharing SinGAN-Seg synthetic data instead of the real datasets for omitting privacy concerns. The second objective was to test how to use SinGAN-Seg pipeline to improve the segmentation performance when the size of training dataset of real images and masks are small. For all the baseline experiments, we selected Unet++ [39] as the main segmentation model according to the performance comparison done by the winning team at EndoCV 2021 [16]. The single-channel dice loss function used in the same study was used to train Unet++ polyp segmentation models. The `se_resnext50_32x4d` network as the encoder of the UNet++ model and `softmax2d` as the activation function of the last layer were used according to the result of the winning team at EndoCV 2021 [16].

Pytorch deep learning library was used as the main development framework for the baseline experiments also. Training data stream was handled using PYRA [14] data loader with Albumentations augmentation library [40]. The real images and the synthetic images were resized into 128×128 using this data handler for all the baseline experiments to save training time because we had to train multiple models for fair comparisons. We have used an initial learning rate of 0.0001 for 50 epochs and then change it to 0.00001 for the rest of the training epochs for all the training processes of UNet++. The UNet++ models used to compare real versus synthetic data were trained 300 epochs in total. On the other hand, the UNet++ models used to measure the effect of using SinGAN-Seg synthetic data for small segmentation datasets were trained only 100 epochs because the size of the data splits used to train the models are getting bigger when increasing the training data. In all the experiments, we have selected the best checkpoint using the best validation IOU score. Finally, dice loss, IOU score, F-score, accuracy, recall, and precision were calculated for comparisons using validation folds.

3.5.1 Synthetic data vs real data for segmentation

We have performed three-folds cross-validation to compare polyp segmentation performance using UNet++ when using real and synthetic data. First, we divided the real dataset (1000 polyp images and the corresponding segmentation masks) into three folds. Then, the trained SynGAN-Seg generative models and the corresponding generated synthetic data were also divided into the same three folds. These three folds are presented using three colors in Step I of Figure 1. In any of the experiments, training data folds and corresponding synthetic data folds were not mixed with the validation data folds. If mixed, it leads to a data leakage problem.

Then, the baseline performance of the UNet++ model was evaluated using the three folds of the real data. In this experiment, the UNet++ model was trained using two folds and validated using the remaining fold of the real data. In total, three UNet++ models were trained and calculated the average performance using dice loss, IOU score, F-score, accuracy, recall, and precision only for the polyp class because the most important class of this dataset is the polyp class. This three-fold baseline experiment setup is depicted on the left side of Figure 8.

The usability of synthetic images and corresponding masks generated from SinGAN-Seg was investigated using three-fold experiments as organized in the right side of Figure 8. In this case, UNet++ models were trained only using synthetic data generated from pre-trained generative models and tested using the real data folds, which were not used to train the generative models used to generate the synthetic data. Five different $N(N = [1, 2, 3, 4, 5])$ amount of synthetic data per image were used to train UNet++ models. This data organization process can be identified easily using the color scheme of the figure. To test the quality of pure generations, first, we used the direct output from SinGAN-Seg to train UNet++ models. Then, the style transfer method was applied with $1 : 1$ content to style ratio for all the synthetic
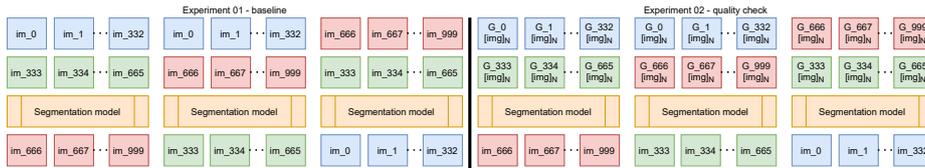
10

Figure 8: Three step experiment setup to analyze the quality of SinGAN output.

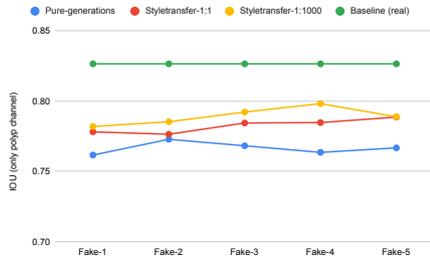
Figure 9: Real versus synthetic data performance comparison with UNet++ and the effect of applying the style-transferring post processing.

3.5.2 Synthetic segmentation data for real small datasets

The main purpose of these experiments are to find the effect of using synthetic data generated from the SinGAN-Seg pipeline instead of small real datasets because the SinGAN-Seg pipeline can generate an unlimited number of synthetic samples per real image. A synthetic sample consists of a synthetic image and the corresponding ground truth mask. Therefore, experts' knowledge is not required to annotate the ground truth mask. For these experiments, we have selected the best parameters of the SinGAN-Seg pipeline from the experiments performed under Section 3.5.1. First, we created small 10 real polyp datasets from the fold one such that each dataset contains R number of images and R can be one of the values of $[5, 10, 15, ..., 50]$. The corresponding synthetic dataset was created by generating 10 synthetic images and corresponding masks per real image. Then, our synthetic datasets consist of S number of images such that $S = [50, 100, 150, ..., 500]$. Then we have compared true pixel percentages of real masks and synthetic masks generated from SynGAN-Seg pipeline using histograms of bin size of 5. The histograms are depicted in Figure 10. The first row represents the histograms of real small detests, and the second row represents the histograms of corresponding synthetic datasets. Compare pairs (one from the top row and the corresponding one from the bottom) to get a clear idea of how the generated synthetic data improved the distribution of masks.

UNet++ segmentation models were trained using these real and synthetic datasets separately. Then we have compared the performance differences using

data. These style transferred images were used as training data and tested using the real dataset. In addition to the 1 : 1 ratio, 1 : 1000 was tested as a style transfer ratio for the same set of experiments.

Table 1 shows the results collected from the UNet++ segmentation experiments for the baseline experiment and the experiments conducted with synthetic data, which contains pure generated synthetic data and style transferred data using 1 : 1 and 1 : 1000. Differences in IOU scores of these three experiments are plotted in Figure 9 for easy comparison.

11

Table 1: Three-fold average of basic metrics to compare real vs synthetic performance with UNet++ and the effect of style-transfers performance

Train data	ST (cw:sw)	dice_loss	iou_score	fscore	accuracy	recall	precision
REAL	NA	0.1123	0.8266	0.8882	0.9671	0.8982	0.9161
FAKE-1	No ST	0.1645	0.7617	0.8357	0.9531	0.863	0.8793
	1:1	0.1504	0.7782	0.85	0.9572	0.8672	0.8917
	1:1000	0.1473	0.782	0.853	0.9591	0.8624	0.9005
FAKE-2	No ST	0.1549	0.7729	0.8453	0.9561	0.8692	0.8895
	1:1	0.155	0.7765	0.8453	0.9575	0.8729	0.8852
	1:1000	0.1477	0.7854	0.8525	0.9609	0.8647	0.9038
FAKE-3	No ST	0.161	0.7683	0.8391	0.9556	0.8568	0.8945
	1:1	0.1475	0.7845	0.8525	0.9585	0.8723	0.8936
	1:1000	0.1408	0.7923	0.8593	0.9629	0.8693	0.9078
FAKE-4	No ST	0.1649	0.7638	0.8352	0.9525	0.8669	0.878
	1:1	0.1464	0.7848	0.8537	0.9594	0.8713	0.8921
	1:1000	0.137	0.7983	0.863	0.9636	0.8653	0.9185
FAKE-5	No ST	0.1654	0.7668	0.8345	0.9563	0.8565	0.8919
	1:1	0.1453	0.7887	0.8547	0.961	0.8703	0.9
	1:1000	0.1458	0.7889	0.8543	0.962	0.8527	0.9211

validation folds. In this experiments, the training datasets were prepared using the fold one. The remaining two folds were used as the validation dataset. The collected results from UNet++ models trained with the real datasets and the synthetic datasets are tabulated in Table 2. A comparison of the corresponding IOU scores are plotted in Figure 11.

4 Discussion

The SinGAN-Seg pipeline has two steps. The first one is generating synthetic polyp images and the corresponding ground truth masks. The second is transferring style from real polyp images to synthetic polyp images to make them more realistic than the pure generations from the first step. We have developed this pipeline to achieve the main two goals. The first one is for sharing medical data when privacy concerns are to share real data. The second one uses is to improve the polyp segmentation performance when the size of training datasets are small.

4.1 SinGAN-Seg as data sharing technique

The SinGAN-Seg can generate unlimited synthetic data with the corresponding ground truth mask, representing real datasets. This SinGAN-Seg pipeline is applicable for any dataset with segmentation masks, particularly when the dataset is not sharable due to privacy concerns. However, in this study, we applied this pipeline to a public polyp dataset with segmentation masks as a case study. Assuming that the polyp dataset is private, we used this polyp dataset as a proof of concept medical dataset. In this case, we published PyPI package, `singan-seg-polyp` which can generate an unlimited number of polyp images and corresponding ground truth masks. If the real polyp dataset is restricted for public use, then this type of pip package can be published as an alternative dataset to represent the real dataset. Alternatively, we can publish a pre-generated synthetic dataset using the SinGAN-Seg pipeline, such as the

12
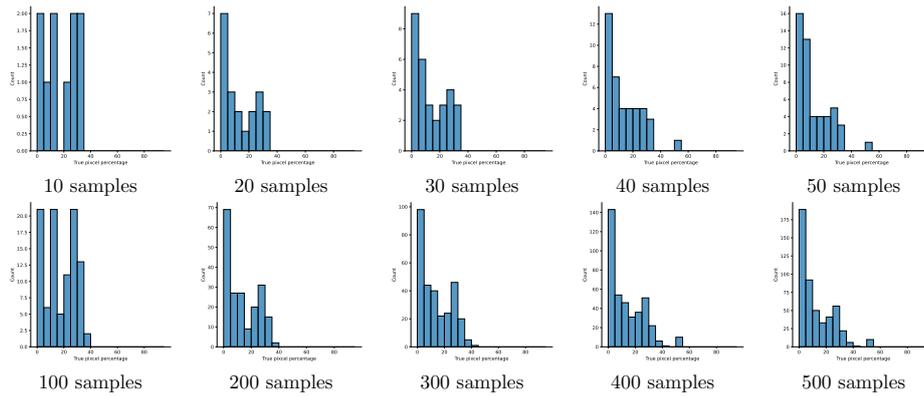
Figure 10: Distribution comparison between real and synthetic mask. Synthetic mask were generated using the SinGAN-Seg.

synthetic polyp dataset published as a case study at https://osf.io/xrgz8/.
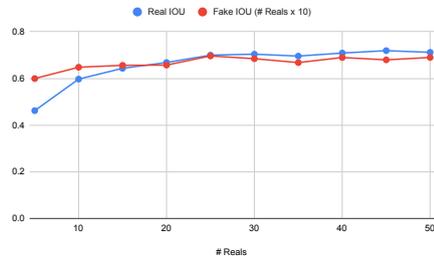
Figure 11: Real versus Fake performance comparison with small training datasets

According to the results presented in Table 1, the UNet++ segmentation network perform better when the real data is used as training data compared to using synthetic data as training data. However, the small performance gap between real and synthetic data as training data implies that the synthetic data generated from the SinGAN-Seg can use as an alternative to sharing segmentation data instead of real datasets, which are restricted to share. The style-transferring step of the SinGAN-Seg pipeline could reduce the performance gap between real and synthetic data as training data for UNet++. The performance gap between real data and the synthetic data as training data for segmentation models is negotiable because the primary purpose of producing the synthetic data is not to improve the performance of segmentation models but to introduce an alternative data sharing which are practically applicable when datasets have privacy concerns to share.

13

Table 2: Real vs Fake comparisons for small datasets. The fake images were generated using style tranfer ration 1 : 1000.

		dice_loss	iou_score	fscore	accuracy	recall	precision
Real	5	0.4662	0.4618	0.5944	0.8751	0.7239	0.6305
Fake	50	**0.3063**	**0.5993**	**0.7048**	**0.9211**	**0.7090**	**0.8133**
Real	10	0.3932	0.5969	0.7079	0.9164	0.7785	0.7516
Fake	100	**0.2565**	**0.6478**	**0.7457**	**0.9259**	**0.7911**	**0.7970**
Real	15	0.2992	0.6431	0.7402	0.9322	0.7388	**0.8602**
Fake	150	**0.2852**	**0.6559**	**0.7624**	**0.9329**	**0.8172**	0.7833
Real	20	0.3070	**0.6680**	**0.7668**	0.9328	**0.7771**	0.8566
Fake	200	**0.2532**	0.6569	0.7544	**0.9342**	0.7317	**0.8827**
Real	25	**0.2166**	**0.6995**	**0.7929**	0.9405	**0.7955**	0.8804
Fake	250	0.2182	0.6961	0.7860	**0.9418**	0.7690	**0.8957**
Real	30	**0.2100**	**0.7037**	**0.7971**	**0.9417**	**0.8005**	0.8758
Fake	300	0.2228	0.6843	0.7797	0.9388	0.7683	**0.8810**
Real	35	**0.2164**	**0.6955**	**0.7889**	**0.9398**	**0.8157**	0.8456
Fake	350	0.2465	0.6677	0.7543	0.9346	0.7385	**0.8933**
Real	40	**0.2065**	**0.7085**	**0.7974**	**0.9417**	0.7881	**0.8947**
Fake	400	0.2194	0.6894	0.7816	0.9305	**0.8276**	0.8219
Real	45	**0.1982**	**0.7188**	**0.8062**	**0.9441**	**0.8120**	**0.8839**
Fake	450	0.2319	0.6794	0.7697	0.9341	0.7859	0.8633
Real	50	**0.2091**	**0.7115**	**0.7948**	**0.9418**	0.7898	**0.8932**
Fake	500	0.2255	0.6896	0.7756	0.9380	**0.7961**	0.8644

4.2 SinGAN-Seg with small datasets

In addition to using the SinGAN-Seg pipeline as a data-sharing technique when the real datasets are restricted to publish, the pipeline can improve the performance of segmentation tasks when a dataset is really small. In this case, the SinGAN-Seg pipeline can generate synthetic data to overcome the problem associated with the small dataset. In other words, the SinGAN-Seg pipeline act as a data augmentation technique. The SinGAN-Seg-based data augmentation acts as an unlimited number of stochastic augmentation techniques due to the randomness of the synthetic data generated from this model. For an example, consider a manual segmentation process such as cell segmentation in any medical laboratory

experiment. This type of task is really hard to perform for experts as well. As a result, the amount of data collected with manually annotated masks are limited. Our SinGAN-Seg pipeline can improve these datasets by generating an unlimited number of random samples from a single manually annotated image. This study showed that these synthetic data generated from a small real dataset can improve the performance of segmentation machine learning models. For example, when the real polyp dataset size is 5 to train our UNnet++ model, the synthetic dataset with 50 samples showed 30% improvement over the IOU score of using the real data samples.

14

5 Conclusions and future work

This paper presented a four-channel SinGAN-Seg model and the corresponding SinGAN-Seg pipeline with a style transfer method to generate realistic synthetic polyp images and the corresponding ground truth masks. This SinGAN-Seg pipeline can be used as an alternative data sharing method when real datasets are restricted to share. Moreover, this pipeline can be used for improving the segmentation performance when we have small segmentation real datasets. The conducted three-folds cross-validation experiments and collected results show that synthetic data can achieve very close performance for segmentation tasks when we use only synthetic images and corresponding masks compared to the segmentation performance if the real data and experts annotated data is used when the real dataset has a considerable amount of data. On the other hand, we show that SinGAN-Seg pipeline can achieve better segmentation performance when training datasets are very small.

In future studies, researchers can combine superresolution GAN model [41] to this pipeline to improve the quality of the output after the style transfer step. When we have high-resolution images, machine learning algorithms show better performance than algorithms trained using low-resolution images [42].

6 acknowledgments

References

[1] F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, and Y. Wang, "Artificial intelligence in healthcare: past, present and future," *Stroke and vascular neurology*, vol. 2, no. 4, 2017.

[2] S. E. Dilsizian and E. L. Siegel, "Artificial intelligence in medicine and cardiac imaging: Harnessing big data and advanced computing to provide personalized medical diagnosis and treatment," *Current Cardiology Reports*, vol. 16, no. 1, p. 441, 2013. [Online]. Available: https://doi.org/10.1007/s11886-013-0441-8

[3] V. L. Patel, E. H. Shortliffe, M. Stefanelli, P. Szolovits, M. R. Berthold, R. Bellazzi, and A. Abu-Hanna, "The coming of age of artificial intelligence in medicine," *Artificial Intelligence in Medicine*, vol. 46, no. 1, pp. 5–17, 2009, artificial Intelligence in Medicine AIME' 07. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0933365708000961

[4] S. Jha and E. J. Topol, "Adapting to artificial intelligence: radiologists and pathologists as information specialists," *Jama*, vol. 316, no. 22, pp. 2353–2354, 2016.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[6] M. A. Hearst, "Support vector machines," *IEEE Intelligent Systems*, vol. 13, no. 4, p. 18–28, Jul. 1998. [Online]. Available: https://doi.org/10.1109/5254.708428

[7] Haifeng Wang and Dejin Hu, "Comparison of svm and ls-svm for regression," in *2005 International Conference on Neural Networks and Brain*, vol. 1, 2005, pp. 279–283.

[8] A. Suárez Sánchez, P. García Nieto, P. Riesgo Fernández, J. del Coz Díaz, and F. Iglesias-Rodríguez, "Application of an svm-based regression model to the air quality study at local scale in the avilés urban area (spain)," *Mathematical and Computer Modelling*, vol. 54, no. 5, pp. 1453–1466, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895717711002196

15

[9] S. Yue, P. Li, and P. Hao, "Svm classification:its contents and challenges," *Applied Mathematics-A Journal of Chinese Universities*, vol. 18, no. 3, pp. 332–342, 2003. [Online]. Available: https://doi.org/10.1007/s11766-003-0059-5

[10] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual review of biomedical engineering*, vol. 2, no. 1, pp. 315–337, 2000.

[11] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, no. 1, p. 29, 2015. [Online]. Available: https://doi.org/10.1186/s12880-015-0068-x

[12] L. K. Lee, S. C. Liew, and W. J. Thong, "A review of image segmentation methodologies in medical image," *Advanced computer and communication engineering technology*, pp. 1069–1080, 2015.

[13] L. A. Torre, F. Bray, R. L. Siegel, J. Ferlay, J. Lortet-Tieulent, and A. Jemal, "Global cancer statistics, 2012," *CA: a cancer journal for clinicians*, vol. 65, no. 2, pp. 87–108, 2015.

[14] V. Thambawita, S. Hicks, P. Halvorsen, and M. A. Riegler, "Pyramid-focus-augmentation: Medical image segmentation with step-wise focus," *arXiv preprint arXiv:2012.07430*, 2020.

[15] D. Jha, S. Ali, N. K. Tomar, H. D. Johansen, D. Johansen, J. Rittscher, M. A. Riegler, and P. Halvorsen, "Real-time polyp detection, localization and segmentation in colonoscopy using deep learning," *Ieee Access*, vol. 9, pp. 40 496–40 510, 2021.

[16] V. Thambawita, S. A. Hicks, P. Halvorsen, and M. A. Riegler, "Divergentnets: Medical image segmentation by network ensemble." in *EndoCV@ ISBI*, 2021.

[17] V. Prasath, "Polyp detection and segmentation from video capsule endoscopy: A review," *Journal of Imaging*, vol. 3, no. 1, p. 1, 2017.

[18] D. Jha, N. K. Tomar, S. Ali, M. A. Riegler, H. D. Johansen, D. Johansen, T. de Lange, and P. Halvorsen, "Nanonet: Real-time polyp segmentation in video capsule endoscopy and colonoscopy," *arXiv preprint arXiv:2104.11138*, 2021.

[19] I. N. Figueiredo, S. Prasath, Y.-H. R. Tsai, and P. N. Figueiredo, "Automatic detection and segmentation of colonic polyps in wireless capsule images," *ICES REPORT*, pp. 10–36, 2010.

[20] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen, D. Johansen, C. Griwodz, H. K. Stensland, E. Garcia-Ceja, P. T. Schmidt, H. L. Hammer, M. A. Riegler, P. Halvorsen, and T. de Lange, "Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, p. 283, 2020.

[21] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach, D. Gil, C. Rodríguez, and F. Vilariño, "Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Computerized Medical Imaging and Graphics*, vol. 43, pp. 99–111, 2015.

[22] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, "Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer," *International journal of computer assisted radiology and surgery*, vol. 9, no. 2, pp. 283–293, 2014.

[23] N. Tajbakhsh, S. R. Gurudu, and J. Liang, "Automated polyp detection in colonoscopy videos using shape and context information," *IEEE transactions on medical imaging*, vol. 35, no. 2, pp. 630–644, 2015.

[24] L. F. Sánchez-Peralta, J. B. Pagador, A. Picón, Á. J. Calderón, F. Polo, N. Andraka, R. Bilbao, B. Glover, C. L. Saratxaga, and F. M. Sánchez-Margallo, "Piccolo white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets," *Applied*

16

Appendix A. Published Articles

Sciences, vol. 10, no. 23, 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/23/8501

[25] "The norwegian data protection authority," accessed: 2021-04-25. [Online]. Available: https://www.datatilsynet.no/en/

[26] "The personal data act," accessed: 2021-04-25. [Online]. Available: https://www.forskningsetikk.no/en/resources/the-research-ethics-library/legal-statutes-and-guidelines/the-personal-data-act/

[27] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)."

[28] P. Edemekong, P. Annamaraju, and M. Haydel, "Health insurance portability and accountability act," *StatPearls*, 2020.

[29] "California consumer privacy act," 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[30] "Act on the protection of personal information," 2003. [Online]. Available: https://www.cas.go.jp/jp/seisaku/hourei/data/APPI.pdf

[31] "Personal information protection commission," 2011. [Online]. Available: http://www.pipc.go.kr/cmt/main/english.do

[32] "The personal data protection bill," 2018. [Online]. Available: https://www.meity.gov.in/writereaddata/files/Personal_Data_Protection_Bill,2018.pdf

[33] F. Renard, S. Guedria, N. D. Palma, and N. Vuillerme, "Variability and reproducibility in deep learning for medical image segmentation," *Scientific Reports*, vol. 10, no. 1, p. 13724, 2020. [Online]. Available: https://doi.org/10.1038/s41598-020-69920-0

[34] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, and M. P. Lungren, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, no. 1, pp. 4–15, 2020.

[35] S. Yu, M. Chen, E. Zhang, J. Wu, H. Yu, Z. Yang, L. Ma, X. Gu, and W. Lu, "Robustness study of noisy annotation in deep learning based medical image segmentation," *Physics in Medicine & Biology*, vol. 65, no. 17, p. 175007, aug 2020. [Online]. Available: https://doi.org/10.1088/1361-6560/ab99e5

[36] T. R. Shaham, T. Dekel, and T. Michaeli, "Singan: Learning a generative model from a single natural image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4570–4580.

[37] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[39] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep learning in medical image analysis and multimodal learning for clinical decision support*. Springer, 2018, pp. 3–11.

[40] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2,

17

370

2020. [Online]. Available: https://www.mdpi.com/2078-1489/11/2/125

[41] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[42] V. L. Thambawita, S. Hicks, I. Strümke, M. A. Riegler, P. Halvorsen, and S. Parasa, "Fr615 impact of image resolution on convolutional neural networks performance in gastrointestinal endoscopy," *Gastroenterology*, vol. 160, no. 6, pp. S–377, 2021.

18

A.26 Paper XXVI - Generative Adversarial Networks For Creating Realistic Artificial Colon Polyp Images

Authors: Vajira Thambawita, Inga Strümke, Steven Hicks, Michael A. Riegler, Pål Halvorsen, Sravanthi Parasa

Abstract: Artificial intelligence is increasingly used to detect and classify colon polyps. However, small datasets are a major obstacle, especially for supervised machine learning. Data collection is challenging, and synthetic data generation, using models such as generative adversarial networks (GANs), may help overcome this hurdle. To determine the clinical utility of synthesized images, we generate images containing colon polyps, and eight endoscopists assess their anatomical correctness.

Published: GIE, DDW Abstract Issue, 2021

Candidate contributions: Vajira contributed to the conception and design of this study. He conducted all the experiments of this research and introduced a novel method to generate synthetic polyp images using a real clean colon image. Vajira evaluated the study critically with experts (doctors) of the domain using a questionnaire. He contributed to drafting the manuscript and revising it.

Thesis objectives: Sub-objective I, Sub-objective II, Sub-objective III, Sub-objective IV

GENERATIVE ADVERSARIAL NETWORKS FOR CREATING REALISTIC ARTIFICIAL COLON POLYP IMAGES

Vajira Thambawita[1,2], Inga Strümke[1], Steven Hicks[1,2], Michael Riegler[1], Pål Halvorsen[1,2], Sravanthi Parasa[3]

[1]Simula Metropolitan Center for Digital Engineering, Oslo, Norway. [2]Oslo Metropolitan University, Oslo, Norway, [3]Department of Gastroenterology, Swedish Medical Group, Seattle, WA, USA

Introduction: Artificial intelligence is increasingly used to detect and classify colon polyps. However, small datasets are a major obstacle, especially for supervised machine learning. Data collection is challenging, and synthetic data generation, using models such as generative adversarial networks (GANs), may help overcome this hurdle. To determine the clinical utility of synthesized images, we generate images containing colon polyps, and eight endoscopists assess their anatomical correctness. **Method**: Using training data from the Kvasir dataset, a large colonoscopy dataset, an image inpainting GAN is used to generate artificial colon polyp images. The GAN is pre-trained with colon images and fine-tuned to generate synthetic polyps using colon images as input. The discriminator of the GAN is used to assess the global and local quality of generated images, in addition to discriminating real from generated. The quality of the generated images is evaluated by 2 expert endoscopists, 3 non-expert endoscopists, and 3 internal medicine residents. The experience of the physicians ranges from 0 to 20 years. Five synthesized and five real images are selected for the evaluation. For each image, the physicians assessed whether the polyp appeared real or generated on a scale from 1-10. **Results**: To measure the agreement among the raters, we calculate Fleiss' kappa for all questions regarding visual appearance across all participants. For all questions, over all, only generated and only real instances, respectively, the Fleiss kappa values are (0.0352, 0.0206, 0.0347) with p-values of (0.00034, 0.176, 0.00909). Similarly, the Fleiss kappa values for the question "Does the polyp appear generated?" are (0.0115, -0.0159, -0.0222). Limiting the included responses to only our two gastroenterologists, the Fleiss' kappa reduces to Cohen's kappa, and the respective values are (-0.235, -0.316, -0.282) with p-values (0.108, 0.193, 0.208). Landis and Koch (1977) provide guidelines for interpreting Fleiss' kappa, and according to these, values in the range 0.01-0.2 indicate only slight agreement between the raters. Moreover, we observe higher reported confidences on generated polyps than real ones. We clearly see that the participants do not find a strong agreement for real or generated, even not the most experienced gastroenterologists. **Conclusion**: We develop and validate a GAN generating high-quality synthetic polyp images. Our evaluation by medical experts indicates only little assessors agreement, even among the most experienced gastroenterologists. We also observe higher reported confidences on generated polyps than real ones. This does not mean that generated polyps are indistinguishable from real ones, but that they share visual and anatomical properties. These promising results show GANs could contribute synthetic data for training and unrestricted sharing.

Figure 1: Generated vs. real polyp images used in the questionnaire.
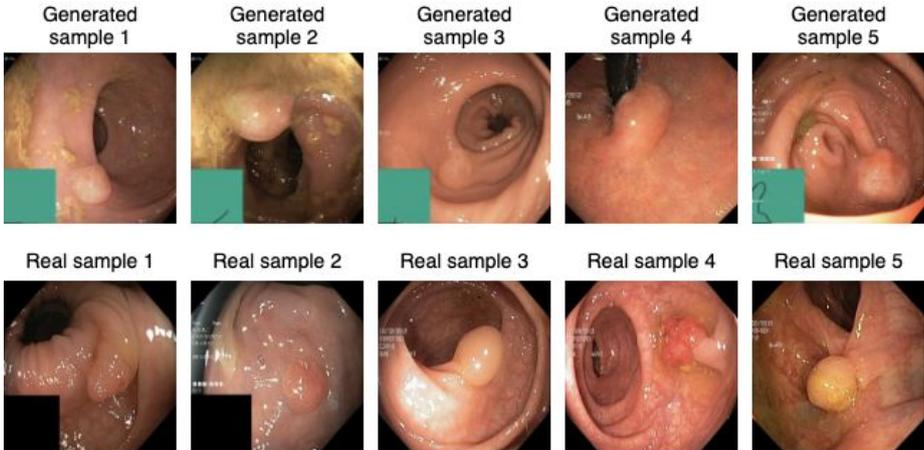
Table 1: Overview of obtained results from all 8 readers (2 experts - EE and 3 non-experts - NE, 3 internal medicine residents - IM).

Reader	TP	FN	FP	TN	Accuracy
EE1	3	4	2	1	0.4
EE2	3	5	2	0	0.3
NE1	2	1	3	4	0.6
NE2	3	2	2	3	0.6
NE3	4	2	1	3	0.7
IM1	4	2	1	3	0.7
IM2	4	3	1	2	0.6
IM3	4	1	1	4	0.8

DATA AUGMENTATION USING GENERATIVE ADVERSARIAL NETWORKS FOR CREATING REALISTIC ARTIFICIAL COLON POLYP IMAGES: VALIDATION STUDY BY ENDOSCOPISTS

V. Thambawita, I. Strümke, S. Hicks, M. Riegler, P. Halvorsen, and S. Parasa

DDW 2021 VIRTUAL
May 21-23

simulamet — Simula Metropolitan Center for Digital Engineering AS
OMET OSO
SWEDISH MEDICAL GROUP

INTRODUCTION

- Artificial intelligence is increasingly used to detect and classify colon polyps. However, small datasets are a major obstacle, especially for supervised machine learning.

- Data collection is challenging, and **synthetic data generation**, using models such as **generative adversarial networks (GANs)**, may help overcome this hurdle.

GOALS:

- To determine the clinical utility of synthesized images containing generated colon polyps.

- To assess synthesized images' anatomical correctness with endoscopists.

METHOD

Using training data from the Kvasir dataset, a large colonoscopy dataset, an **image inpainting GAN** is used to generate artificial colon polyp images.

- The GAN is pre-trained with colon images and fine-tuned to generate synthetic polyps using colon images as input.

- The discriminator of the GAN is used to assess the global and local quality of generated images, in addition to discriminating real from generated. The quality of the generated images is evaluated by **2 expert endoscopists, 3 non-expert endoscopists, and 3 internal medicine residents.**

- Five synthesized and five real images are selected for the evaluation. For each image, the physicians assessed whether the polyp appeared real or generated on a scale from 1-10.

RESULTS

To measure the agreement among the raters, we calculate Fleiss' kappa for all questions. *Kappa values in the range 0.01-0.2 indicate only slight agreement between the raters.*

- For all questions, over all, only generated and only real instances, respectively, the Fleiss kappa values were (0.0352, 0.0206, 0.0347) with p-values of (0.00034, 0.176, 0.00909).

- "**Does the polyp appear generated?**" the kappa values were (0.0115, -0.0159, -0.0222).

- Limiting to only our two gastroenterologists, the kappa values were *(-0.235, -0.316, -0.282) with p-values (0.108, 0.193, 0.208).*

- **We clearly see that the participants do not find a strong agreement for real or generated, even not the most experienced gastroenterologists.**
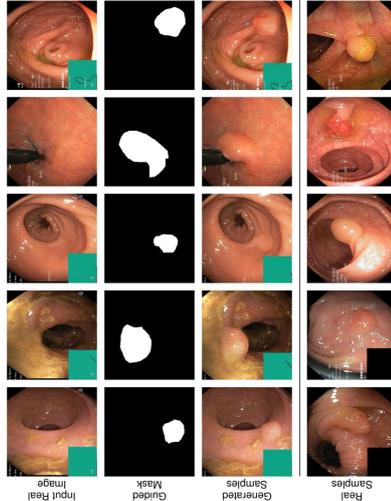
Real and synthetic image samples.

(row labels: Input Real Image / Guided Mask / Generated Samples / Real Samples)

Reader	TP	FN	FP	TN	Accuracy
EE1	3	4	2	1	0.4
EE2	3	5	2	0	0.3
NE1	2	1	3	4	0.6
NE2	3	2	2	3	0.6
NE3	4	2	1	3	0.7
IM1	4	2	1	3	0.7
IM2	4	3	1	2	0.6
IM3	4	1	1	4	0.8

Overview of obtained results from all 8 readers.
(EE - experts, NE - non-experts, IM - internal medicine residents)

CONCLUSIONS

We develop and validate a GAN generating high-quality synthetic polyp images.

Our evaluation by medical experts indicates only little assessors agreement, even among the most experienced gastroenterologists. We also observe higher reported confidences on generated polyps than real ones. This does not mean that generated polyps are indistinguishable from real ones, but that they share visual and anatomical properties. **These promising results show GANs could contribute synthetic data for training and unrestricted sharing.**

ACKNOWLEDGEMENTS

A.27 Paper XXVII - Identification of Spermatozoa by Unsupervised Learning from Video Data

Authors: Michael A. Riegler, Trine B. Haugen, Mette Haug Stensen, Oliwia Witczak, Hugo L. Hammer, Pål Halvorsen, Michael A. Riegler

Abstract: Identification of individual sperm is essential to assess a given sperm sample's motility behaviour. Existing computer-aided systems need training data based on annotations by professionals, which is resource demanding. On the other hand, data analysed by unsupervised machine learning algorithms can improve supervised algorithms that are more stable for clinical applications. Therefore, unsupervised sperm identification can improve computer-aided sperm analysis systems predicting different aspects of sperm samples. Other possible applications are assessing kinematics and counting of spermatozoa. Generative adversarial networks (GANs) have become common AI methods to process data in an unsupervised way. Based on single image frames extracted from videos, a GAN (SinGAN) can be trained to determine and track locations of sperms by translating the real images into localization paintings. The resulting model showed the potential of identifying the presence of sperms without any prior knowledge about data. Visual comparisons of localization paintings to real sperm images show that inverse training of SinGANs can track sperms. Converting colour frames into grayscale frames and using grayscale synthetic sperm-like frames showed the best visual quality of generated localization paintings of sperm frames.

Published: Eshre, 2021

Candidate contributions: Vajira contributed to the design of this concept and he developed all the models and the corresponding experiments for tracking sperms using a modified SinGAN generative model. He evaluated results collected from the experiments and published them for public use (https://vlbthambawita.github.io/singan-sperm/). Vajira contributed to drafting and revising the manuscript.

Thesis objectives: Sub-objective IV

A.27. Paper XXVII - Identification of Spermatozoa by Unsupervised Learning from Video Data

Presentation preference	**Oral presentation or poster**
Abstract title	**Identification of spermatozoa by unsupervised learning from video data**
Biography	**Vajira Thambawita is a Ph.D. student at the HOST department at SimulaMet and the Department of Computer Science at OsloMet. He comes from a computer engineering background and focuses on finding applications of deep generative models in the medical domain. Besides his primary research, he improves the quality of machine learning (ML) applications in medicine by investigating proper analysis and benchmarking ML on medical data.**

V. Thambawita[1], T.B. Haugen[2], M.H. Stensen[3], O. Witczak[2], H.L. Hammer[4], P. Halvorsen[1], M.A. Riegler[1].

[1]Simula Metropolitan Center for Digital Engineering, Department of Holistic Systems, Oslo, Norway.

[2]Faculty of Health Sciences- OsloMet – Oslo Metropolitan University, Department of Life Sciences and Health, Oslo, Norway.

[3]Fertilitetssenteret, Fertilitetssenteret, Oslo, Norway.

[4]Faculty of Technology- Art and Design- OsloMet -Oslo Metropolitan University, Department of Computer Science, Oslo, Norway.

Study question:

Can artificial intelligence (AI) algorithms identify spermatozoa in a semen sample without using training data annotated by professionals?

Summary answer:

Unsupervised AI methods can discriminate the spermatozoon from other cells and debris. These unsupervised methods may have a potential for several applications in reproductive medicine.

What is known already:

Identification of individual sperm is essential to assess a given sperm sample's motility behaviour. Existing computer-aided systems need training data based on annotations by professionals, which is resource demanding. On the other hand, data analysed by unsupervised machine learning algorithms can improve supervised algorithms that are more stable for clinical applications. Therefo unsupervised sperm identification can improve computer-aided sperm analysis systems predicting different aspects of sperm samp Other possible applications are assessing kinematics and counting of spermatozoa.

Study design, size, duration:

Three sperm-like paint images were manipulated using a graphic design tool and used to train our AI system. Two paintings have an ash colour background and randomly distributed white colour circles, and one painting has a predefined pattern of circles. Selected semen sample videos from a public dataset with videos obtained from 85 participants were used to test our AI system.

Participants/materials, setting, methods:

Generative adversarial networks (GANs) have become common AI methods to process data in an unsupervised way. Based on singl image frames extracted from videos, a GAN (SinGAN) can be trained to determine and track locations of sperms by translating the re images into localization paintings. The resulting model showed the potential of identifying the presence of sperms without any prior knowledge about data.

Main results and the role of chance:

Visual comparisons of localization paintings to real sperm images show that inverse training of SinGANs can track sperms. Convert colour frames into grayscale frames and using grayscale synthetic sperm-like frames showed the best visual quality of generated localization paintings of sperm frames. Feeding real sperm video frames to the SinGAN at different scaling factors, which is defining the resolution of the input image, showed different quality levels of generated sperm localization paintings. A sperm frame given to t algorithm with a scaling factor of one leads to random sperm tracking, while the scales two to four result in more accurate localizati maps than scaling levels five to eight. In contrast, scales from six to eight result in an output close to the input frame. The proposed method is robust in terms of the number of spermatozoa, meaning that the detection works well for samples with a low or high sperm count. For visual comparisons, visit our Github page: https://vlbthambawita.github.io/singan-sperm/. The sperm tracking speed of o SinGAN using an NVIDIA 1080 graphic processing unit, is around 17 frames per second, which can be improved by using parallel vid processing capabilities. This shows the capability of using this method for real-time analysis.

Limitations, reasons for caution:

Unsupervised methods are hard to train, and the results need human verification. The proposed method will need quality control an must be standardized. Unsupervised sperm tracking SinGAN may identify blurry bright spots as non-existing sperm heads which m restrict the use of SinGAN sperm tracking for sperm counting.

Wider implications of the findings:

Assessment of semen samples according to the WHO guidelines is subjective and resource-demanding. This unsupervised model might be used to develop new systems for less time-consuming and more accurate evaluation of semen samples. It may also be use for real-time analysis of prepared spermatozoa for use in assisted reproduction technology.

COI	**I have no potential conflict of interest to disclose**
Keywords	**identification of spermatozoa** **artificial intelligence** **unsupervised artificial intelligence** **generative adversarial networks** **sperm localization**

Your abstract will be reviewed and scored and subsequently accepted for presentation or rejected. This process will take some time to complete and the outc will be available by 26 April 2021. At that time you will be notified as to whether your abstract has been accepted or not.

Identification of spermatozoa by unsupervised learning from video data

V. Thambawita, T.B. Haugen, M.H. Stensen, O. Witczak, H.L. Hammer, P. Halvorsen, M.A. Riegler

Introduction:

Identification of individual sperm is essential to assess a given sperm sample's motility behavior. Existing computer-aided systems need training data based on annotations by professionals, which is resource demanding. On the other hand, data analyzed by unsupervised machine learning algorithms can improve supervised algorithms that are more stable for clinical applications. Therefore, unsupervised sperm identification can improve computer-aided sperm analysis systems predicting different aspects of sperm samples. Other possible applications are assessing kinematics and counting of spermatozoa.

Materials:

Three sperm-like paint images were manipulated using a graphic design tool and used to train our AI system. Two paintings have an ash color background and randomly distributed white color circles, and one painting has a predefined pattern of circles. Selected semen sample videos from a public dataset with videos obtained from 85 participants were used to test our AI system.
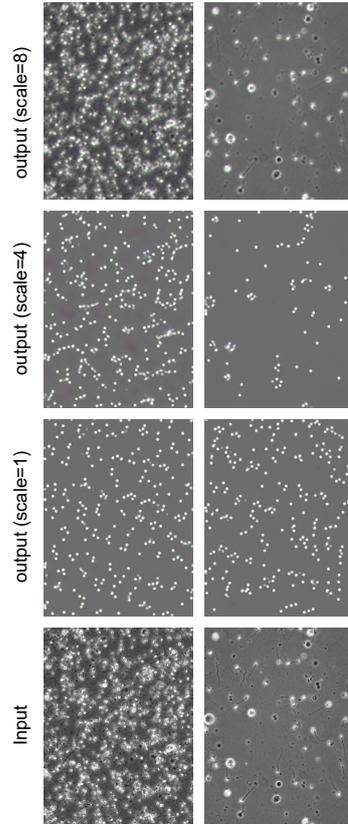
Methods:

Generative adversarial networks (GANs) have become common AI methods to process data in an unsupervised way. Based on single image frames extracted from videos, a GAN (SinGAN) can be trained to determine and track locations of sperms by translating the real images into localization paintings. The resulting model showed the potential of identifying the presence of sperms without any prior knowledge about data.

Sperm like synthetic paintings
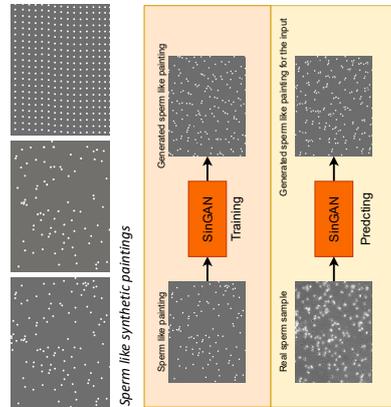
Training and prediction process

Result:

Visual comparisons of localization paintings to real sperm images show that inverse training of SinGANs can track sperms. Converting colour frames into grayscale frames and using grayscale synthetic sperm-like frames showed the best visual quality of generated localization paintings of sperm frames. Feeding real sperm video frames to the SinGAN at different scaling factors, which is defining the resolution of the input image, showed different quality levels of generated sperm localization paintings.

Input output (scale=1) output (scale=4) output (scale=8)

Sample input and the corresponding predicted output from the pre-trained SinGAN. Three different scale levels are presented from the total of eight.

A sperm frame given to the algorithm with a scaling factor of one leads to random sperm tracking, while the scales two to four result in more accurate localization maps. In contrast, scales from six to eight result in an output close to the input frame. The proposed method is robust in terms of the number of spermatozoa, meaning that the detection works well for samples with a low or high sperm count. For visual comparisons, visit our Github page: https://vlbthambawita.github.io/singan-sperm/. The sperm tracking speed of our SinGAN using an NVIDIA 1080 graphic processing unit, is around 17 frames per second, which can be improved by using parallel video processing capabilities. This shows the capability of using this method for real-time analysis.

Limitation:

Unsupervised methods are hard to train, and the results need human verification. The proposed method will need quality control and must be standardized. Unsupervised sperm tracking SinGAN may identify blurry bright spots as non-existing sperm heads which may restrict the use of SinGAN sperm tracking for sperm counting.

Future work:

Assessment of semen samples according to the WHO guidelines is subjective and resource-demanding. This unsupervised model might be used to develop new systems for less time-consuming and more accurate evaluation of semen samples. It may also be used for real-time analysis of prepared spermatozoa for use in assisted reproduction technology.

A.28 Paper XXVIII - DeepSynthBody: the Beginning of the End for Data Deficiency in Medicine

Authors: Vajira Thambawita, Steven A. Hicks, Jonas Isaksen, Mette Haug Stensen, Trine B. Haugen, Jørgen Kanters, Sravanthi Parasa, Thomas de Lange, Håvard D. Johansen, Dag Johansen, Hugo L. Hammer, Pål Halvorsen, Michael A. Riegler

Abstract: Limited access to medical data is a barrier on developing new and efficient machine learning solutions in medicine such as computer-aided diagnosis, risk assessments, predicting optimal treatments and home-based personal healthcare systems. This paper presents DeepSynthBody: a novel framework that overcomes some of the inherent restrictions and limitations of medical data by using deep generative adversarial networks to produce synthetic data with characteristics similar to the real data, so-called DeepSynth (deep synthetic) data. We show that DeepSynthBody can address two key issues commonly associated with medical data, namely privacy concerns (as a result of data protection rules and regulations) and the high costs of annotations. To demonstrate the full pipeline of applying DeepSynthBody concepts and user-friendly functionalities, we also describe a synthetic medical dataset generated and published using our framework. DeepSynthBody opens a new era of machine learning applications in medicine with a synthetic model of the human body.

Published: In proceedings of the International Conference on Applied Artificial Intelligence(ICAPAI), 2021

Candidate contributions: Vajira came with this idea and he contributed to the conception and design. He implemented the whole pipeline and did all the implementations. Vajira has developed this DeepSynthBody as a framework and this is the core of his thesis storyline. He performed several experiments to generate gastrointestinal tract images using GANs as a proof of concept of this framework. Final outcome of this study is available online to end-users of this framework at https://deepsynthbody.org/ which was developed by him. Vajira contributed to drafting the manuscript and revising it.

Thesis objectives: Sub-objective II, Sub-objective IV

DeepSynthBody: the beginning of the end for data deficiency in medicine

Vajira Thambawita[*‡], Steven A. Hicks[*‡], Jonas Isaksen[x], Mette Haug Stensen[xi], Trine B. Haugen[‡],
Jørgen Kanters[x], Sravanthi Parasa[xii], Thomas de Lange[‖††], Håvard D. Johansen[¶], Dag Johansen[¶],
Hugo L. Hammer[‡*], Pål Halvorsen[*‡], and Michael A. Riegler[*]

[*]SimulaMet, Norway [‡]Oslo Metropolitan University, Norway [¶]UIT The Arctic University of Norway
[‖]Bærum Hospital, Norway [††]Sahlgrenska University Hospital-Mölndal Hospital, Sweden
[x]University of Copenhagen, Denmark [xi]Fertilitetssenteret, Norway [xii]Swedish Medical Center, USA
Contact email: vajira@simula.no

Abstract—Limited access to medical data is a barrier on developing new and efficient machine learning solutions in medicine such as computer-aided diagnosis, risk assessments, predicting optimal treatments and home-based personal healthcare systems. This paper presents DeepSynthBody: a novel framework that overcomes some of the inherent restrictions and limitations of medical data by using deep generative adversarial networks to produce synthetic data with characteristics similar to the real data, so-called DeepSynth (deep synthetic) data. We show that DeepSynthBody can address two key issues commonly associated with medical data, namely privacy concerns (as a result of data protection rules and regulations) and the high costs of annotations. To demonstrate the full pipeline of applying DeepSynthBody concepts and user-friendly functionalities, we also describe a synthetic medical dataset generated and published using our framework. DeepSynthBody opens a new era of machine learning applications in medicine with a synthetic model of the human body.

Index Terms—DeepSynthBody, synthetic medical data, deep synthetic human body, synthetic data, GAN, DeepSynth augmentation, privacy issue, medical data privacy, multi-model DeepSynth, DeepSynth explainable AI, explainable DeepSynth

I. INTRODUCTION

Artificial intelligence (AI) has become widespread in medicine because of the success achieved by AI algorithms and rapid hardware development in the recent decade. AI-based medical solutions vary from computer-aided diagnosis [1], [2], risk assessments [3], [4], and predicting optimal treatments [5], [6] in hospitals to home-based personal health care systems [7]–[9].

We identify four main stages in applying machine learning (ML) solutions to medicine as depicted in Figure 1. In the first stage, data is collected from hospitals, outside clinics, health registers and other locations such as medical research institutions. These data have different data modalities like biological signals, images, videos, and unique data formats such as $4D$ data from magnetic resonance imaging (MRI)

Fig. 1. The main four steps for applying deep learning in medicine. XAI is the abbreviation for explainable artificial intelligence.

machines. In the second stage of this process, domain experts annotate or label the data to train ML models. In the third step, one can investigate ML solutions to find the best model which can be generalized into final products and train them using the annotated data coming from the previous stage, or train unsupervised ML models without annotated or labeled data. However, before releasing ML solutions as a final product in medicine, explainable artificial intelligence (XAI) should be applied to explain the decision taken from ML algorithms. This will be an important step in the future to increase trust in the models and lead to better models in general.

In the first stage of the process introduced in Figure 1, producing open access datasets in the medical domain is a time-consuming task [10], [11] or impossible because of protocols that should follow specific rules and regulations such as the general data protection regulation (GDPR) [12] in EU countries. Moreover, rules and regulations for producing open access medical data vary from region to region. For example, Norway should follow the Norwegian data protection authority (NDPA) rules, the health research act [13] and enforce the personal data act in addition to following GDPR. While there is no central level privacy protection guideline in US like GDPR in Europe, there are rules and regulations applicable and in effect through other US privacy laws, such as, Health Insurance Portability and Accountability Act (HIPAA) [14] and California Consumer Privacy Act (CCPA) [15]. In Asian counties, they follow their own set of rules country-wise, such as, Japan's Act on Protection of Personal Information [16], South Korea Personal Information Protection Commission [17] and the Personal Data Protection Bill from India [18]. With these restrictions, researchers publish only the research methods and, as a result, other researchers cannot reproduce or compare those results using the same methods because of limited access to real datasets. Furthermore, universities or other research institutions that use medical domain data for teaching purposes use the same medical domain datasets for years, which probably affects the quality of education. Therefore, data sharing restrictions resulting from privacy protocols are identified as one of the main research problems addressed with the presented framework.

After collecting data, one should collect corresponding ground truth (the second stage as depicted in Figure 1). In

some cases, the ground truth is present, for example the outcomes of treatments in retrospective data, but usually annotation must be done manually. Medical data annotations should be done by experts to ensure the best quality. However, the annotating and labeling process for creating medical domain data is a time-consuming and costly process [19]. This process is identified as another problem to tackle for the larger goal of producing large-scale datasets needed to develop AI-based medical systems.

The third step in Figure 1 is to apply ML methods, which is an often used AI-based solution. As a result of privacy protocols and the aforementioned complex data retrieval and annotation problems, researchers and industries do often not have open access to large datasets annotated by experts. Because of limited data to train supervised ML models, they become less reliable [20] (as a result of poor generalizability) and have fewer functionalities such as limited interpretability [21].

The fourth step in the Figure 1 represents the final stage of producing products of ML to use in clinical settings. Transparency is key, particularly if algorithmic output conflicts with a medical doctors assessment. In this stage, explaining the prediction results (XAI) is an important step because it is the only step in which one can convince doctors and patients to accept decisions made by ML solutions. Explanation by example is currently the preferred XAI method by non-experts [22]. Privacy issues can limit explaining deep learning (DL) solutions by examples [23], when the data used for the examples is restricted.

To address all these issues which are secondary to lack of available medical data, we present a novel framework called DeepSynthBody. This framework can be used to overcome the data accessibility problems in the medical domain by generating realistic synthetic data with reduced privacy issues. The DeepSynthBody concept is inspired by deepfakes [24], [25] which are produced by deep generative models introduced by Ian et al. [26]. By developing the DeepSynthBody framework, we intend to achieve the following objectives:

1) overcome the privacy related limitations for medical data by producing open access deep synthetic data.
2) reduce the time-consuming and resource-consuming process of medical data labeling and annotation.
3) find intra-correlations of human body organs (how one organ affect to other organs) and functions and reproduce them to produce a new model for the human body.

This paper introduces the complete DeepSynthBody framework. In addition, we introduce new research directions based on our framework. To the best of our knowledge, this is the first paper to introduce a complete framework to address the previously discussed challenges in preparing medical data for ML using GAN. We anticipate that DeepSynthBody will be a possible solution for data privacy challenges, mitigate the time and cost to annotate medical data, and aims to find hidden correlations of human body functions using multi-model data to introduce a new model for the human body.

The DeepSynthBody pipeline consists of four steps, namely, (I) collecting real data and analysis, (II) developing generative models, (III) producing deep synthetic data, and (IV) explainable DeepSynth AI and DeepSynth explainable AI. In Section II, we introduce the DeepSynthBody framework via three subsections: DeepSynthBody pipeline, additional features of DeepSynthBody, and technical features behind DeepSynthBody. In Section III, we introduce a case study that was implemented based on the DeepSynthBody concept. At the end, we discuss limitations and future research directions using DeepSynthBody in Section IV followed by a conclusion in Section V.

II. DEEPSYNTHBODY

Our DeepSynthBody framework consists of four major steps: (I) collecting real data and analysis, (II) developing generative models, (III) producing deep synthetic data, and (IV) explainable DeepSynth AI and DeepSynth explainable AI as depicted in the main flow diagram in Figure 2. The top arrow, *Restricted access* represents areas where we consider privacy-related restrictions and guidelines to follow. In contrast, the *Open access* arrow represents the flow after resolving privacy issues with real data by replacing them with deep synthetic data. We discuss more details in the following sections.

A. Pipeline

In this section, we discuss the whole process step by step which is the recommended order one should follow in practice to use the DeepSynthBody framework. In the framework, we can identify mainly two types of users; 1) contributors to develop deep generative models for the DeepSynthBody framework and 2) end users, who are using pre-trained deep generative models from DeepSynthBody to produce synthetic data for their research or applications. In our case study described in Section III, we discuss the development stage from a developer perspective and the data generation from a user perspective.

1) Collecting real data and analysis (I): This step represents the first step in the DeepSynthBody framework (top part of Figure 2) and can be primarily be divided into three sub-steps, namely data categorization, data annotation, and data analysis. After collecting medical data, the data should be categorized into a body category and a data format available in DeepSynthBody. Then, the data must be tagged as either having annotations or not. Finally, a set of baseline metrics should be made by training a ML model to perform a given task. If real datasets are restricted, then all sub-steps can only be done by research institutions with allowed access following the data privacy protocols. If machine learning experts are not accessible to the institutions, then the institutions have to use user friendly GUI-based tools [27], [28] to build ML models. Otherwise, one can follow steps (I) to (IV) in the DeepSynthBody framework without any restrictions with aids from ML experts.

In the first step, we classify almost all the data into 11 organ system categories [29] based on the anatomy of the
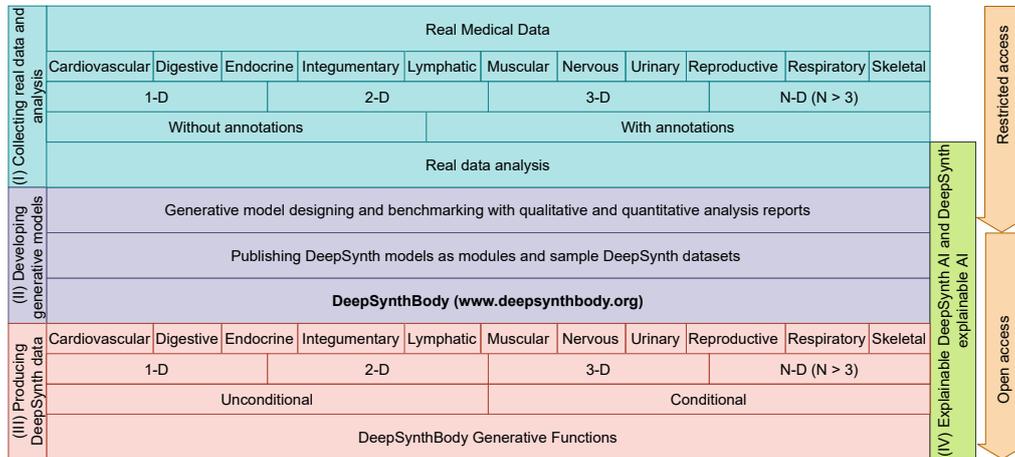
Fig. 2. The overall framework of the DeepSynthBody concept. The main four stages (I) collecting real data and analysis, (II) developing generative models, (III) producing DeepSynth data and (IV) explainable DeepSynth artificial intelligence (AI) and DeepSynth explainable AI are presented using four different colors.

human body. These categories are cardiovascular, digestive, endocrine, integumentary, lymphatic, muscular, nervous, urinary, reproductive, respiratory, and skeletal. This is the main classification which is available to the end users of our DeepSynthBody, who are producing synthetic data from this framework.

The data classified using organ system classification can be classified further using the complexity of the data in terms of its dimensions. The dimensions of real data increase the complexity of generative models, which are implemented in later sections. Taking the dimensionality of the data into account, we classify all biomedical data into one-dimensional (1-D), two-dimensional (2-D), three-dimensional (3-D), and N-dimensional (N-D such that $N > 3$) data to use in development stages in DeepSynthBody framework.

Under the 1-D data, we cover biomedical signals collected from the human body. Well-known signals are Electroencephalogram (EEG), Electrocardiogram (ECG), Electromyogram (EMG), Mechanomyogram (MMG), Electrooculography (EOG), Galvanic skin response (GSR), and Magnetoencephalogram (MEG). However, biomedical signals can be analysed as 2-D, 3-D, or N-D depending on the method. For example, an electrocardiogram (ECG) signal consists of 12-channels and can be processed as 12 streams of 1-D data or a 2-D matrix of $12 \times times$ after combining all 12 channels. Therefore, we consider data dimensions coming through data sources (medical devices) as the dimensionality of data in this DeepSynthBody framework instead of considering the data dimensions used in data processing techniques.

Medical imaging techniques are commonly used to visualize human body organs, functions, and important for making diagnosis and treatment suggestions. In the DeepSynthBody framework, we subcategories these medical imaging data

into three categories, 2-D, 3-D, and N-D, based on the dimensionality of the data obtained. Various technologies produce medical images such as radiography, magnetic resonance imaging, nuclear medicine, ultrasound, elastography, photoacoustic imaging, tomography, functional near-infrared spectroscopy, and magnetic particle imaging under 2-D, 3-D, or N-D. For example, images collected from data sources like video cameras or images extracted from another type of data like MRI, can be categories under 2-D data type. Similarly, video collected from cameras in the medical equipment like endoscope or colonoscope can be identified as a 3-D data type when considering time as the third dimension. However, some data sources produce 3-D data in a spatial domain, e.g., fMRI which produces data in the Neuroimaging Informatics Technology Initiative (NIFTI) format. In this NIFTI format, we can observe 3-D volume spatial data for a specific timestamp. However, we can classify this kind of 3-D data into 4-D (into N-D because $N > 3$) when the source produces a series of 3-D data points. In addition to 4-D data, there are data sources producing 5-D data [30] which are considered under our N-D data category. Under this definition, we identify all real data sources through 1-D, 2-D, 3-D, or N-D classes.

After collecting data following the data dimension classification, we can further categorize data into two categories: (i) without annotations (or labels) and (ii) with annotations. In this step, we also consider if the data was labeled by experts or not. Generally, most of the data coming from medical systems do not have expert annotations or labels to use with ML algorithms. However, if annotated data can be obtained at this step, we can develop advanced deep generative models with controllable input (conditional generative models) [31], which take input parameters such as class labels to produce deep synthetic data in the step (II) which is developing generative

models. While one of our primary objectives is to reduce annotation cost and time required from experts, at least a few data points with annotated ground truth are needed to use conditional generative models.

Analysis of original data produces baseline statistics about the data itself and the ML models trained using this real data. This analysis step is significant for later steps to identify if our deep generative model can generate data samples from a similar distribution compared to the real distribution used to train the GAN models. Therefore, we investigate possible use cases, for example, classification, detection, segmentation, or forecasting, using ML algorithms with real datasets to use those as baseline or benchmarks in the later step (II).

2) Developing generative models (II): This step presents all stages from designing to publishing generative models as a package to end users. Deep generative models used to produce deep synthetic data should be designed using real data sources coming from the first layer (I). The whole process of this step can be divided into three sub-processes, namely designing and evaluating generative adversarial networks (GANs), producing modules (e.g., python packages) and publishing them under www.deepsynthbody.org for the end users who use DeepSynthBody to generate synthetic data.

Recent studies show that deep generative models can be used to generate diverse data formats such as text [32]–[34], signals [35]–[37], images [38]–[40], videos [41]–[43] as well as complex data sources like MRI [44] (this generation process can be considered as 3-D data generation). In the context of DeepSynthBody, we do not consider text GANs because biomedical data sources do not produce a text like data as the main data format at the moment of initiating the DeepSynthBody framework. In contrast to this, all other deep generative models can be trained to generate random deep synthetic data (unconditional) or conditional deep synthetic data [31], [45]. When we compare conditional generative models with unconditional generative models, we can see that conditional generative models have the advantage to have controllable parameters to produce deep synthetic data as needed. Then, the conditional GANs can be used to produce synthetic data with corresponding ground truth.

Not only designing generative models but researching better evaluation methods [46]–[48] for GAN to quantify the quality of generated data is a necessary sub-task under this step. Therefore, proper GAN evaluation methods should be performed along the process of each specific GAN generation such as images, videos, or other medical data formats. With these evaluation methods and real data analysis methods from the top layers, we can perform benchmark experiments and publish benchmark results to the end users as supporting quantitative and qualitative supplement materials to analyze their research studies. Until these benchmark comparisons are available, users can perform their own baseline experiments, which can be time consuming compared to pre-evaluated benchmark results.

3) Producing deep synthetic data (III): End users who use the DeepSynthBody generative models to generate synthetic data interact with our framework through step (III). The step (III) has a flow similar to the flow of the step (I), but they are slightly different. Step (I) uses categorization to classify input data while step (III) uses the same categorization to generate synthetic data. The data annotation layer of step (I) is replaced with two new data generation processes, namely unconditional and conditional. Finally, data generation processes are used in this step (III) instead of real data analysis processes in step (I).

We use the same 11 categories as used in step (I) to generate deep synthetic data from DeepSynthBody to end users. Therefore, this layer is defined as the output layer of the DeepSynthBody pipeline. We further split the 11 categories into four categories based on the data dimensionality (1-D, 2-D, 3-D, and N-D) as discussed in Section II-A1. This layer enables us to decide the data output format when there are multiple data formats for a selected category from the above layer. For example, fMRI data can be generated as images (2-D) or volume data (3-D). In addition, users can also decide that the generation process is either unconditional or conditional. The conditional generation will be allowed if conditional generative models are available for that specific generation task in the layer (II). In the framework, several generative models for a specific generative task can exist (e.g., two different GAN models to generate ECGs, one with conditions and one without). If more than one model exists, the end users can chose a model for their specific application based on benchmark reports or comparisons. Similarly, combinations of multiple GANs can be used as a possible final generative model, which can open for more diversity within the generated data. Technical details about these DeepSynth generative functions are discussed in Section II-C.

4) Explainable DeepSynth AI and DeepSynth XAI (IV): The fourth layer, *called explainable DeepSynth AI and DeepSynth XAI* is introduced to embed explainability and transparency into all other layers. This layer is essential to explain our deep generative models to increase trust and enable deeper failure analysis. In the medical domain, XAI plays a significant role in increasing trust to accept solutions from ML models that generally perform classification, detection, and segmentation. When we use our DeepSynthBody generated synthetic data as a replacement to real medical data, we have to have eXplainable DeepSynth Artificial Intelligence (XSAI), which is introduced as a specific subsection of XAI in this framework. XSAIs main goal is to explain deep generative models to increase understanding of the generative process and quality of the generated data [49], [50].

XSAI discusses the explainability of generative models. In contrast to this, deep synthetic data can be used to support explanations of other ML models and we discuss this under DeepSynth XAI (SXAI). In this context, the main goal is not to explain deep generative models, but to explain other ML models used to classify, detect and segment medical data using DeepSynth-data as examples. We can see that this problem is related to privacy issues as well. For example, when researchers cannot explain their ML models by examples

because these examples could raise privacy problems, they can use DeepSynth examples to explain their models with less concern about patient privacy.

B. Additional features

After establishing the whole pipeline steps from (I) to (IV), this DeepSynthBody concept will open new research areas such as generating deep synthetic data with many modalities such as generating DeepSynth sperm sample videos with corresponding blood reports and synthetic patient data like age and body mass index (BMI). Multi-modal data should be fed into the DeepSynthBody framework in step (I) to achieve this DeepSynth multimodality goal. When the DeepSynthBody is enriched with diverse deep synthetic data generators, the end users get facilitated to generate unlimited data, missing data, and exploring data distributions. In addition, exploring correlations between deep synthetic data that reflect real human body correlations will enable different medical perspectives for diagnosis and treatments and find hidden clues and findings of our human body such as how organs interact with each other [51]. Controllable deep synthetic data generation can generate deep synthetic data using given conditions using input parameters through conditional generative models. This conditional generation can enhance generalizability by improving real datasets, solving data imbalance problems, and introducing new augmented data for ML models that used only real data to train.

C. Technical features

In the initial phase of implementing the DeepSynthBody framework, we use Python version 3 as the main programming language and Python package index (PyPI) as the main hosting place to publish DeepSynthBody functionalities. First, all DeepSynth generative models are published as individual PyPI packages. Therefore, DeepSynth generative models and corresponding benchmark models can be implemented using any ML framework such as Pytorch [52], Tensorflow [53], Microsoft Cognitive Toolkit (CNTK) [54] or other common ML frameworks that support PyPI. The final PyPI package of DeepSynthBody (pip install deepsynthbody) collects all submodules and reorganizes them into the DeepSynthBody framework to give a high-end user experience. An example PyPI package flow is given in Figure 3.

In addition to providing DeepSynth generative models, the DeepSynthBody framework provides functionalities to use pre-generated data published by developers of generative models when the end users do not need to run deep generative models to generate deep synthetic data. On the other hand, this kind of pre-generated data availability is essential when the end users do not have enough computation power to run high-end generative models, requiring more computation power in the inference stage also.

III. Case study

As an example how to follow the framework guidelines we implemented a use-case study which consists of gastrointestinal (GI) tract [55] findings collected from endoscopic images,
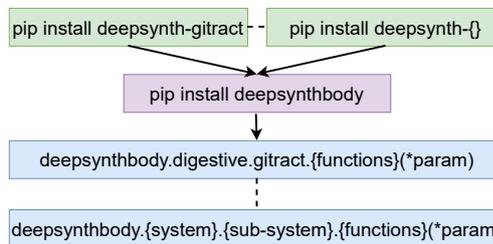

Fig. 3. An example python package flow. The top layer represent sub-modules and the second layer represent the main module of DeepSynthBody.

which represent the 2-D data format of our DeepSynthBody framework. However, we do not cover implementation details of the used generative models because our primary goal is to introduce the framework and not present a reproducible methodology for deep generative models, which is out of the scope of this paper. Generally, every package or every generative model comes with its own research protocols because developing and publishing a DeepSynth model, and a dataset from scratch is a complex research and development task in itself. Therefore, the DeepSynthBody framework provides corresponding research details (corresponding papers explaining details of the methods and implementations) to the end users with the DeepSynthBody functionalities.

HyperKvasir is a GI dataset [55] that is public and open for the research community. We selected this dataset to show that we can reproduce similar deep synthetic data representing the real dataset distribution. To generate DeepSynth-gastroenterology data from this dataset, we used the unlabeled data in the dataset, consisting of around 100,000 images to train a deep generative model of style GAN-v2 [56]. We trained this generative model 10000k steps over more than eight days to get a stable generative model using the Pytorch implementation of Style GAN-v2[1].

The Fréchet inception distance (FID) [48] scores calculated from different checkpoints are tabulated in Table I. These calculations were performed using the FID Pytorch implementation [57] with different feature extraction layers, 64: first max pooling features, 192: second max pooling features, 768: pre-aux classifier features and 2048: final average pooling features, which are introduced in the original implementation [57] . Small FID value represent better quality images than higher FID values. Each check point number represents the check-point at $chk\_point \times 1000$, which is used to calculate FIDs. The best FID values are presented using bold numbers in Table I. We selected our best check point to publish in the DeepSynthBody framework according to the best FID values of 0.1980 and 41.2030 calculated using feature layers 768 and 2048 consecutively. These last two layers selected because they represent high end features compared to basic feature extractions coming from feature layers 64 and 192. We can achieve further improvement to this FID values with more

---

[1]https://github.com/lucidrains/stylegan2-pytorch

TABLE I
FID SCORES CALCULATED FROM DIFFERENT CHECKPOINTS

chk_point	FID_64	FID_192	FID_768	FID_2048
0	39.1090	189.4938	2.6159	342.0751
100	1.7710	8.3480	0.3030	58.9490
200	1.6616	8.0271	0.2977	59.7215
300	1.6575	7.8310	0.2671	52.6597
400	1.2801	6.1183	0.2429	48.5694
500	1.2262	5.8759	0.2372	49.3512
600	1.5974	7.4586	0.2626	52.9441
700	1.3826	6.5063	0.2363	46.2668
800	1.1938	5.9112	0.2312	46.7931
900	**0.6537**	**3.0260**	0.2017	44.3310
1000	0.8736	4.2926	**0.1980**	**41.2039**

training step. However, we stopped our experiment at this level because of time limitation.

After selecting the best checkpoint (alternative option is publishing all interesting checkpoints to select the best at the end user level) to publish with DeepSynthBody framework, we prepared and published a PyPI package for our sub-module (`pip install deepsynth-gitract`). However, this module is specially designed to target the development process of DeepSynthBody. Then, we embedded this sub-module into the DeepSynthBody module (`pip install deepsynthbody`), and all functionalities, research papers, and corresponding usages of these sub-modules are discussed in www.deepsynthbody.org[2]. In this case study, our deepsynth-gitract module is published under `deepsynthbody.digestive.gitract` module which is structured as depicted in Figure 3. In Figure 3, we use word *system* to represent the 11 systems introduced in Figure 2. The keyword *sub-system* is used to represent subcategories under these 11 main systems. In our use case, it is the GI tract (gitract). In Figure 3, *functions* is to represent all generative functions (functionalities) under these sub-modules and **param* represents all input parameters to generative functions.

In this study, we introduce two functions, `generate()` and `generate_interpolation()`. The main functionality of the `generate()` function is to generate DeepSynth GI tract images which can be used in another study. Sample DeepSynth GI tract images generated from this function are presented in Figure 4. Using the `generate_interpolation()` function, we can generate DeepSynth GI tract images between two random noise points for a given interpolation steps. Figure 5 shows DeepSynth GI tract images generated using two different seeds with interpolation step size of 100. In this Figure 5, we visualize the first five consecutive images out of 100.

IV. DISCUSSION

DeepSynthBody assists in getting anonymous, realistic, synthetic medical data to be used for ML that otherwise would be unavailable due to privacy concerns and/or the lack of available medical personal to perform the tedious annotation process. Nevertheless, the state-of-the-art generative models
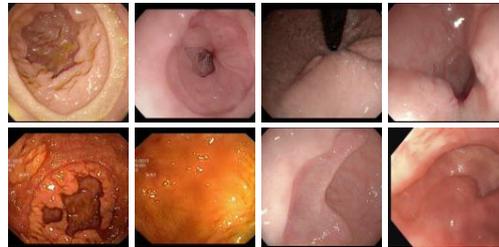
[2]www.deepsynthbody.org

Fig. 4. Style-GAN generated random gastrointestinal-tract findings.

Fig. 5. First five samples generated with 200 interpolation steps for two different random seeds. First and second row represent the two different random seeds.

are incapable [58] to reproduce the same distribution as the real distribution but future research studies can overcome these limitations [59], [60]. On the other hand, limited medical domain data minimize using deep generative models to produce DeepSynth medical data. However, recent advancement [61], [62] in deep generative models to use limited data shows the potentials of applying deep generative models even with smaller datasets.

Finding inter-correlations between DeepSynth generative models such as how organs interact with each other is an advanced functionality in this DeepSynthBody framework. In the initial stage, we cannot handle this functionality due to a lack of real data sources and corresponding deep generative models. Not only that, to accomplish this advanced option, we need more multi-model datasets as well as collected data from the same patients because inter-correlations can be found from the data coming from a single patient. Therefore, this framework's success depends on successful contributions from the research community who are working with medical data and ML, and deep generative model designing.

Initiating this framework collects scattered research in deep generative applications in medicine into a single framework. Then, this DeepSynthBody acts as a repository for synthetic medical data. Similarly, this framework can be used as a data compression mechanism for large scale medical datasets because we keep generative models that can create data if required instead of large datasets. From another perspective, having many DeepSynth generative models under the DeepSynthBody framework can be considered as a new possible way to explain the human body.

Appendix A. Published Articles

V. Conclusion and future work

In this study, we presented a novel framework called DeepSynthBody to solve data deficiency problems caused by privacy issues and time-consuming and costly medical data annotation processes. DeepSynthBody can provide synthetic data as replacements to real data and this can fulfill our first objective. We present that introducing conditional GANs in DeepSynthBody can produce synthetic data and corresponding ground truths to tackle our second objective of reducing cost of medical data annotations. We show that collecting diverse data from different data sources from the same patients can direct our DeepSynthBody framework to find intra-body correlations which is discussed as our third objective. However, in this study, we have provided case studies only for the first objective as a result of limited time and resources.

We can generate reliable and generalizable ML solutions in medicine with the aid of DeepSynthBody. It also might open up the possibility to find advanced human body correlations (intra-body correlations) through conditional generative models developed under this Deepsynthbody concept. Ultimate accomplishments of these findings can produce a novel model to explain the human body. The Explainability section (IV) of the DeepSynthBody framework opens new research areas to explain deep generative models in medicine and use DeepSynthBody data as examples to explain other ML solutions which are used for classification, detection and segmentation in medicine.

Besides our primary objectives, DeepSynthBody can be used as a repository for deep generative models used in medicine and a data compression mechanism to keep big medical datasets in a small storage without any privacy concerns and space to save large amounts of the data. In this context, contributions from the research communities in medicine and ML to develop and improve the sub-modules of DeepSynthBody is the key to the ultimate success of DeepSynthBody framework.

VI. Acknowledgements

References

[1] M. Riegler, K. Pogorelov, P. Halvorsen, T. de Lange, C. Griwodz, P. T. Schmidt, S. L. Eskeland, and D. Johansen, "Eir—efficient computer aided diagnosis framework for gastrointestinal endoscopies," in *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2016, pp. 1–6.

[2] S. M. Weiss, C. A. Kulikowski, S. Amarel, and A. Safir, "A model-based method for computer-aided medical decision-making," *Artificial Intelligence*, vol. 11, no. 1, pp. 145 – 172, 1978, applications to the Sciences and Medicine. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0004370278900152

[3] N. N. Khanna, A. D. Jamthikar, D. Gupta, M. Piga, L. Saba, C. Carcassi, A. A. Giannopoulos, A. Nicolaides, J. R. Laird, H. S. Suri *et al.*, "Rheumatoid arthritis: atherosclerosis imaging and cardiovascular risk assessment using machine and deep learning–based tissue characterization," *Current atherosclerosis reports*, vol. 21, no. 2, p. 7, 2019.

[4] S. Mani, Y. Chen, T. Elasy, W. Clayton, and J. Denny, "Type 2 diabetes risk forecasting from emr data using machine learning," in *Proceedings of the American Medical Informatics Association (AMIA) Annual Symposium*, vol. 2012. American Medical Informatics Association, 2012, p. 606.

[5] C. Huang, R. Mezencev, J. F. McDonald, and F. Vannberg, "Open source machine-learning algorithms for the prediction of optimal cancer drug therapies," *PLoS One*, vol. 12, no. 10, p. e0186906, 2017.

[6] A. Khodayari-Rostamabad, J. P. Reilly, G. M. Hasey, H. de Bruin, and D. J. MacCrimmon, "A machine learning approach using eeg data to predict response to ssri treatment for major depressive disorder," *Clinical Neurophysiology*, vol. 124, no. 10, pp. 1975–1985, 2013.

[7] F. R. Vogenberg, C. I. Barash, and M. Pursel, "Personalized medicine: part 1: evolution and development into theranostics," *Pharmacy and Therapeutics*, vol. 35, no. 10, p. 560, 2010.

[8] E. Grönvall and N. Verdezoto, "Beyond self-monitoring: Understanding non-functional aspects of home-based healthcare technology," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, ser. UbiComp '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 587–596. [Online]. Available: https://doi.org/10.1145/2493432.2493495

[9] S. Ghorbani and W. Du, "Personal health service framework," *Procedia Computer Science*, vol. 21, pp. 343 – 350, 2013, the 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH). [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050913008387

[10] M. Ienca, A. Ferretti, S. Hurst, M. Puhan, C. Lovis, and E. Vayena, "Considerations for ethics review of big data health research: A scoping review," *PloS one*, vol. 13, no. 10, p. e0204937, 2018.

[11] B. M. Knoppers and A. M. Thorogood, "Ethics and big data in health," *Current Opinion in Systems Biology*, vol. 4, pp. 53–57, 2017.

[12] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)."

[13] "The health research act," https://www.forskningsetikk.no/en/resources/the-research-ethics-library/legal-statutes-and-guidelines/the-health-research-act/, accessed: 2021-02-25.

[14] P. Edemekong, P. Annamaraju, and M. Haydel, "Health insurance portability and accountability act," *StatPearls*, 2020.

[15] "California consumer privacy act," 2018. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[16] "Act on the protection of personal information," 2003. [Online]. Available: https://www.cas.go.jp/jp/seisaku/hourei/data/APPI.pdf

[17] "Personal information protection commission," 2011. [Online]. Available: http://www.pipc.go.kr/cmt/main/english.do

[18] "The personal data protection bill," 2018. [Online]. Available: https://www.meity.gov.in/writereaddata/files/Personal_Data_Protection_Bill,2018.pdf

[19] M. J. Willemink, W. A. Koszek, C. Hardell, J. Wu, D. Fleischmann, H. Harvey, L. R. Folio, R. M. Summers, D. L. Rubin, and M. P. Lungren, "Preparing medical imaging data for machine learning," *Radiology*, vol. 295, no. 1, pp. 4–15, 2020.

[20] V. Thambawita, D. Jha, H. L. Hammer, H. D. Johansen, D. Johansen, P. Halvorsen, and M. A. Riegler, "An extensive study on cross-dataset bias and evaluation metrics interpretation for machine learning applied to gastrointestinal tract abnormality classification," *ACM Trans. Comput. Healthcare*, vol. 1, no. 3, Jun. 2020.

[21] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King, "Key challenges for delivering clinical impact with artificial intelligence," *BMC medicine*, vol. 17, no. 1, pp. 1–9, 2019.

[22] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, "How can i explain this to you? an empirical study of deep neural network explanation methods," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[23] J. van der Waa, E. Nieuwburg, A. Cremers, and M. Neerincx, "Evaluating xai: A comparison of rule-based and example-based explanations," *Artificial Intelligence*, vol. 291, p. 103404, 2021. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370220301533

[24] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and J. Ortega-Garcia, "Deepfakes and beyond: A survey of face manipulation and fake detection," *Information Fusion*, vol. 64, pp. 131–148, 2020.

[25] P. Korshunov and S. Marcel, "Deepfakes: a new threat to face recognition? assessment and detection," *arXiv preprint arXiv:1812.08685*, 2018.

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.

[27] "The nvidia deep learning gpu training system (digits)," 2021, accessed: 2021-03-26. [Online]. Available: https://developer.nvidia.com/digits

[28] V. Thambawita, H. L. Hammer, M. Riegler, and P. Halvorsen, "Ganex: A complete pipeline of training, inference and benchmarking gan experiments," in *Proceedings of the International Conference on Content-Based Multimedia Indexing (CBMI)*, 2019, pp. 1–4.

[29] "Organ system definition," https://biologydictionary.net/organ-system/, accessed: 2021-01-25.

[30] Yan Zhang, P. J. Passmore, and R. H. Bayford, "Visualization and post-processing of 5d brain images," in *Proceedings of the IEEE Engineering in Medicine and Biology (EMBC)*, 2005, pp. 1083–1086.

[31] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[32] T. Iqbal and S. Qureshi, "The survey: Text generation models in deep learning," *Journal of King Saud University - Computer and Information Sciences*, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157820303360

[33] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing, "Toward controlled generation of text," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 1587–1596.

[34] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin, "Adversarial feature matching for text generation," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 4006–4015.

[35] S. Harada, H. Hayashi, and S. Uchida, "Biosignal generation and latent variable analysis with recurrent generative adversarial networks," *IEEE Access*, vol. 7, pp. 144 292–144 302, 2019.

[36] C. Donahue, J. McAuley, and M. Puckette, "Adversarial audio synthesis," *arXiv preprint arXiv:1802.04208*, 2018.

[37] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-series generative adversarial networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 5508–5518.

[38] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019. [Online]. Available: https://openreview.net/forum?id=B1xsqj09Fm

[39] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4401–4410.

[40] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[41] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1526–1535.

[42] M. Saito and S. Saito, "Tganv2: Efficient training of large models for video generation with multiple subsampling layers," *CoRR*, vol. abs/1811.09245, 2018. [Online]. Available: http://arxiv.org/abs/1811.09245

[43] J. Wu, Z. Huang, D. Acharya, W. Li, J. Thoma, D. P. Paudel, and L. V. Gool, "Sliced wasserstein generative models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *Proceedings of the International Workshop on Simulation and Synthesis in Medical Imaging (SASHIMI)*. Springer, 2018, pp. 1–11.

[45] S. Ravuri and O. Vinyals, "Classification accuracy score for conditional generative models," *arXiv preprint arXiv:1905.10887*, 2019.

[46] A. Borji, "Pros and cons of gan evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41 – 65, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314218304272

[47] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *arXiv preprint arXiv:1606.03498*, 2016.

[48] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, 2017, pp. 6629–6640.

[49] V. Nagisetty, L. Graves, J. Scott, and V. Ganesh, "xai-gan: Enhancing generative adversarial networks via explainable ai systems," *arXiv e-prints*, pp. arXiv–2002, 2020.

[50] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "Gan dissection: Visualizing and understanding generative adversarial networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[51] R. P. Bartsch, K. K. L. Liu, A. Bashan, and P. C. Ivanov, "Network physiology: How organ systems dynamically interact," *PLOS ONE*, vol. 10, no. 11, pp. 1–36, 11 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0142143

[52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[54] F. Seide and A. Agarwal, "Cntk: Microsoft's open-source deep-learning toolkit," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 2135–2135.

[55] H. Borgli, V. Thambawita, P. H. Smedsrud, S. Hicks, D. Jha, S. L. Eskeland, K. R. Randel, K. Pogorelov, M. Lux, D. T. D. Nguyen *et al.*, "Hyperkvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, vol. 7, no. 1, pp. 1–14, 2020.

[56] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *CoRR*, vol. abs/1912.04958, 2019.

[57] M. Seitzer, "pytorch-fid: FID Score for PyTorch," https://github.com/mseitzer/pytorch-fid, August 2020, version 0.1.1.

[58] D. Bau, J.-Y. Zhu, J. Wulff, W. Peebles, H. Strobelt, B. Zhou, and A. Torralba, "Seeing what a gan cannot generate," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4502–4511.

[59] S. Arora, A. Risteski, and Y. Zhang, "Do gans learn the distribution? some theory and empirics," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[60] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (gans)," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2017, pp. 224–232.

[61] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *arXiv preprint arXiv:2006.06676*, 2020.

[62] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient gan training," *arXiv preprint arXiv:2006.10738*, 2020.