UNIVERSITY
OF OSLO

Master's thesis

Exploring the Potential of Diffusion Models in Generating Synthetic Polyps

Alexander Klougman Pishva

Robotics and Intelligent Systems
60 ECTS study points

Department of Informatics
Faculty of Mathematics and Natural Sciences

Spring 2023

Alexander Klougman Pishva

Exploring the Potential of Diffusion Models in Generating Synthetic Polyps

Supervisors:
Steven A. Hicks
Vajira Thambawita
Pål Halvorsen
Jim Tørresen

Abstract

Colorectal cancer in the form of polyps is life-threatening and early detection is central to a person's survival rate. Colonoscopy is a common method carried out by medical professionals to detect these polyps in the lower gastrointestinal (GI)-tract, however, only relying on humans is unfavorable as humans make mistakes and typically have a miss rate between 14% to 30% when it comes to detecting polyps. Supplementary tools such as computer aided diagnosis (CAD) systems are therefore investigated to aid medical professionals. CAD systems have been shown to be capable of increasing detection efficiency, and accuracy, and aiding in the early detection of colorectal cancer. The absence of labeled data, however, is frequently a problem when developing CAD systems.

Machine learning (ML) models, which are at the heart of CAD systems, require a large quantity of data to be trained effectively. However, legal limitations and the high cost of conducting exams make it difficult to obtain medical data. Currently, annotating data also requires a highly qualified medical expert, which complicates matters. The issue of having few positive cases (polyp is present) in the medical field is being addressed through ongoing research into the generation of synthetic medical data.

This thesis explores the use of generative diffusion models to generate synthetic polyp images to address this sparse domain. Generated polyps from the diffusion models are presented to domain experts to assess their realism as a qualitative measure. We subsequently propose the RePolyp framework to generate synthetic polyps that can be used for segmentation tasks to increase dataset size.

In the end, we demonstrate that generative models namely diffusion models can increase segmentation models' performance. The segmentation models trained with synthetic polyps were significantly improved for one out of three datasets and inconclusive for the two others as they are non-significant improvements.

Acknowledgments

I would like to thank everyone at Simula especially my two main supervisors Steven A. Hicks and Vajira Thambawita. I also want to thank my co-supervisor Pål Halvorsen and internal supervisor at the University Jim Tørresen. Finally, I'd want to thank my family for all of their help and support over the course of my work.

The research presented in this thesis has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3) servers, which is financially supported by the Research Council of Norway under contract 270053.

Contents

Acronyms

AI artificial intelligence. 15, 23, 73

CAD computer aided diagnosis. i, 1, 2, 9, 14, 15, 30, 31, 33

CNN convolutional neural network. 6, 9, 15, 17, 20, 21, 23

DDIM Denoising Diffusion Implicit Models. 28

DDPM Denoising Diffusion Probabilistic Models. xiii, 27, 28, 38, 41, 47, 48, 60, 62, 65, 66, 73, 75

DL deep learning. 9, 17, 19, 23, 29, 31, 37, 40, 45, 73

DNN Deep Neural Network. 26, 42

DSC Dice similarity coefficient. xii, 43, 68, 70

FID Fréchet inception distance. xii, xiii, 41, 42, 44–46, 48–52, 54, 57, 60, 62, 63, 65, 66, 71, 74

GAN generative adversarial network. xi, 2, 6, 25, 26, 28–30, 35, 37, 40, 41, 44, 71, 73, 75

GI gastrointestinal. i, xi, xiii, 2, 5, 6, 9–14, 30–32, 35, 37, 50, 52, 59, 73, 74

IoU Intersection over Union. xii, 42, 43, 70

IS inception score. 41

mIoU mean Intersection over Union. xii, 43, 68, 70

ML machine learning. i, 1, 3, 9, 14–16, 23, 29, 30, 60

MLP multilayer perceptron. xi, 6, 15, 17, 18, 20, 23

RAM Random-access memory. 3, 31

SGD stochastic gradient descent. 19, 41

VAE variational autoencoder. 2, 25, 28, 29

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Colorectal cancer is the second leading cause of cancer-related deaths for both men and women in the world with more than 935,000 deaths and 1.9 million new colorectal cancer (including anus) cases estimated to occur in 2020 [1], corresponding to about one in ten cancer cases (10.0 %) and deaths (9.4 %). Data shows that the risk of colon cancer is increasing with age with the majority of colorectal cancers occurring in people older than 50 [2]. Colorectal cancer is however highly treatable when diagnosed at a localized stage [3] with a 5-year relative survival rate of 90%. About 37% of patients are diagnosed at this early stage [4].

For colorectal cancer, colonoscopy is considered the gold standard as colonoscopies give a detailed look at the rectum and the entire large intestine. A colonoscopy is a small flexible tube with a camera attached to the end that is inserted through the rectum. The camera will then give a doctor who specializes in the digestive system a continuous video feed to look for abnormalities in the colon. Polyps in the colon are abnormalities that can be of varying sizes with either precancerous or cancerous polyps. If a polyp is detected during a colonoscopy it is usually removed if possible. Polyps removed during a colonoscopy are then examined by a pathologist to determine if the polyp contains cancerous or precancerous cells. Polyp detection during colonoscopies is prone to human error with miss rates between 14% to 30% [5].

Computer aided diagnosis (CAD) systems are used for diagnostic aid to provide doctors with better medical decision-making [6]. Machine learning (ML) models do however usually need a lot of data to be generalizable. Datasets in the medical domain are often small and far between. This thesis will try to address this issue by generating synthetic data to solve the lack of data that ML solutions require inspired by frameworks such as DeepSynthBody [7].

1.2 Problem Statement

Based on the motivation in the previous section, our goal is to improve performance in the medical domain where data is sparse. Section 1.1 points out how a significant portion of polyps are missed during colonoscopy due to human error. CAD systems are therefore introduced which are tools that use machine learning in efforts to reduce polyp miss rates. However, CAD systems need large amounts and quality data to be robust and reliable. Therefore, we want to generate synthetic polyps to artificially increase the amount of data available and images to look realistic.

The research question we are aiming to answer with this thesis is as follows:

> *Can synthetic polyp images look realistic and be used to improve the performance of segmentation models?*

The research question can be broken down into two parts, one concerning improvement for systems that use synthetic images and the other the realism of generated images when interpreted by humans. To answer the research question we divide our efforts and experiments into three objectives. The overall ultimate goal is to improve generated synthetic polyp images' realism and quality. We, therefore, assess quality and realism with quantitative assessments involving objective metrics. In addition, is domain experts asked to give subjective feedback on the generated polyp realism as a qualitative assessment.

- **Objective 1** Generate synthetic images from the GI-tract by training diffusion models on the data collected in the thesis. The generated samples should ideally be of the same quality and diversity as the data they were trained on. The generative models should be able to generate a complete image or use inpainting.
- **Objective 2** The second objective is training segmentation models either on real data or a mix of real and synthetic data. Investigation of the performance of segmentation models when trained on real or mixed data.
- **Objective 3** The third objective presents generated images to domain experts to assess realism. The results are a qualitative assessment that will give an indication of whether or not the synthetic images are indistinguishable from real images.

1.3 Scope and Limitations

Diffusion models are up-and-coming generative models that have gained huge attention in the last few years. They are however slow when it comes to sampling speed compared to their competing architecture such as GANs and variational autoencoders (VAEs). This is rooted in the trilemma problem in generative models. Due to the time consumption of diffusion models when it comes to sampling speed were also the amount of configuration limited. Ideally, should k-fold Cross-Validation have been used, but was not performed

due to time limitations. In the last months have also diffusion models have been seen suggest to also leak more training data than their counterparts. This issue is monitored and talked about further in Section 3.8.

As a case study for our specified research problem and as a representative area of application for the techniques we develop, we will use the medical situation of locating and detecting polyps (Objective 2). Despite the fact that our thesis is restricted to one particular medical case, we admit that the techniques examined throughout this thesis will typically be useful for a number of disciplines where machine learning can be used.

The data we train on is either from HyperKvasir or Kvasir-SEG for polyps with segmentation masks. Both of these datasets are collected by Simula and may have some biases. The amount of polyp samples in Kvasir-SEG is also limited which in turn reduces the generalizability of our generative models. To evaluate the performance of adding synthetic data to the segmentation models are subsequently only Kvasir-SEG, CVC-ClinicDB, and ETIS-LaribDB were used for validation.

Due to both hardware and the complexity of the models trained is image sizes restricted to size 128×128. On the hardware side, Random-access memory (RAM) and memory are the main limitations restricting us from directly generating larger images. However, we can train secondary models that can increase image size known as super-resolution, a common practice for diffusion models.

1.4 Ethical Considerations

The use of ML in the medical field presents numerous ethical considerations, including issues related but not limited to privacy, bias, and accountability. When it comes to using generative models, a number of ethical considerations must be taken into account. One key concern is the potential for these models to be used in a malicious manner, particularly within a clinical context. For instance, synthetic data generated by diffusion models could be passed off as real data, which would be deeply problematic [8]. Furthermore, as synthetic data is not covered by common data protection rules, it may be used to spread information outside the confines of an organization. This can cause problems with anonymity at the personal level and possibly make it possible for general trends to be abused. In order to unfairly determine insurance premiums, for instance, false data may be utilized [9].

The ability to properly judge the quality of synthetic data is another ethical concern. Even though there are metrics in place to assess the realism of synthetic data, are presently no quantitative metrics for comparing the anonymity and realism of synthetic data to actual data. Synthetic data must be distinct enough to prevent original data points from being recognized while still being realistic enough to be used in place of actual data to lower the risk of information leakage.

1.5 Research Methods

This thesis applied the Association for Computing Machinery's (ACM) "Computing as a Discipline" [10] methodology which presents computing in three main categories.

1.5.1 Theory

The theory is rooted within mathematical science and describes a theoretically coherent development of a theory. They are described as four steps, (I) characterize objects of study (definition), (II) hypothesize possible relations among them (theorem), (III) determine whether the relationships are true (proof), and (IV) Interpret results.

This thesis goes through the elementary theory behind machine learning, more specifically deep learning and generative models. We identify problems such as generalizability and synthetic data quality.

1.5.2 Abstraction

The abstraction of this thesis is rooted in an experimental scientific method and relates to the development and investigation of the hypothesis. The four stages of investigation are as follows: (I) form a hypothesis, (II) construct a model and make a prediction, (III) design an experiment and collect data, (IV) analyze results.

The experiments conducted during this thesis fall under this category. We looked into the causes of these results based on the outcomes of our tests. We then came up with a theory as to why it behaved in this way and modified subsequent experiments in light of this theory. The validity of our hypothesis was then tested by running models that had been modified in accordance with it, and the results either supported or refuted our initial notion.

1.5.3 Design

The design part is rooted in engineering and relates to the construction of a system to solve a given problem. It is described as four steps and is as follows: (I) state requirements, (II) state specifications, (III) design and implement the system, (IV) test the system.

By using the generative models and framework for generating synthetic images, our work complies with the steps of this category. The generated images were then utilized as a component of this thesis to conduct tests that demonstrated the value of the system being able to create synthetic images.

1.6 Main Contributions

The goal of this thesis is to address the problems presented in Section 1.2. We list the three primary contributions that this work made in order to accomplish each goal. In addition to the work presented in this thesis is a paper "RePolyp: A Framework for Generating Realistic Colon Polyps with Corresponding Segmentation Masks using Diffusion Models" shown in Appendix A accepted to CBMS 2023 conference. The paper compares the training of the polyp segmentation model (U-Net) with only real data versus a mix of real and synthetic. The results are validated on three different polyp datasets with two of the validation dataset having a different origin than the training data. The code is available in provided Github repository [1] with some documentation.

Objective 1

Generate synthetic images from the GI-tract by training diffusion models on the data collected in the thesis. The generated samples should ideally be of the same quality and diversity as the data they were trained on. The generative models should be able to generate a complete image or use inpainting.

This objective covers the training of the diffusion models and their generated samples. The generated synthetic samples are evaluated based on a quantitative metric. The models' polyp feature knowledge is also shown through interpolation. This will be the main contribution of this thesis in hopes of increasing data available in sparse data domains such as the medical sector.

Objective 2

The second objective is training segmentation models either on real data or a mix of real and synthetic data. Investigation of the performance of segmentation models when trained on real or mixed data.

This objective answers half of the summarized question in the problem statement on whether or not synthetic data can improve segmentation models. The synthetically generated samples with segmentation masks are used in combination with real data to train segmentation models based on the U-Net architecture. We compared U-Net models only trained on real data and U-Net models trained on a mix of real and fake data and observed an increase with adding synthetic data.

Objective 3

The third objective presents generated images to domain experts to assess realism. The results are a qualitative assessment that will give an indication of whether or not the synthetic data is indistinguishable from the real.

The third objective uses polyp images without segmentation masks that are evaluated by domain experts. The domain experts are presented with real polyp images and synthetic polyp images. This objective addresses the issues of the realism of generated

[1] https://github.com/alexakp/Master-Thesis

samples.

1.7 Thesis Outline

The thesis is organized into five chapters. The first two chapters are introductory giving the reader the necessary knowledge and history of what our work relates to. Chapters three and four explain the ideas used and the work done in this thesis. The fifth chapter concludes the thesis. The structure of all chapters except Chapter 1 is presented below as follows:

Chapter 2 - Background

The background chapter introduces the necessary information for image generation in the medical domain. The background is essentially split into two parts, the medical background and the technical background with details pertaining to machine learning. The medical background begins by introducing the GI-tract. We then dive deeper within the GI-tract learning of how it is composed, its function, and common diseases we can find. We then take a quick look at two common ways of medically examining the GI-tract and how technology can be used to aid professionals in the medical field. We then introduce the machine learning background which is used as a foundation for the work in this thesis. We then take a look at the history of neural networks and how they are built up going from MLP to convolutional neural network (CNN). We then elaborate further explaining terms like deep learning, regularization, and finally generative models. The background should give a good understanding of both GANs and diffusion models as they are vital to understanding the work of this thesis.

Chapter 3 - Methodology

The methodology chapter introduces possible approaches to generating synthetic images in the medical domain with inpainting. Models used either for image generation or segmentation are also explained with their relevant metrics to evaluate performance. The datasets used in this thesis are also presented and discussed here.

Chapter 4 - Polyp Generation

The Polyp Generation chapter presents all results from pre-training and fine-tuning of diffusion models. In addition, is interpolation in latent space shown between two source images. Both quantitative and qualitative data are presented. The results are made up of generative models trained on five different types of data.

Chapter 5 - Polyp Segmentation with Synthetic Data

The Polyp Segmentation with Synthetic Data chapter presents the use of synthetic images to train segmentation models. It compares segmentation models only trained on real data versus models trained on a mix of real and synthetic data. The results are made up of images from the previous chapter that are inpainted. A total of three inpainting models are presented and six U-Net models are used for the segmentation of

polyps.

Chapter 6 - Conclusion and Future Work

The final chapter summarizes the work done and contributions in this thesis. Lastly, is future work presented that introduces possible areas to be researched.

Chapter 2

Background

This chapter presents the technical background needed to understand how deep learning (DL) techniques are applied to medical data. The medical background gives a key understanding of the GI-tract, particularly polyps. The three main categories within ML are then introduced and then gradually work our way to more advanced topics and models such as neural networks in detail. Furthermore, we take a thorough look at CNNs and state-of-the-art generative models in computer vision and look at some drawbacks of these advanced models.

2.1 Medical Background

This section will start by discussing the importance of the GI system in the human body. Studying the various GI-tract disorders is the focus of our effort for this thesis. Next are some landmarks and findings presented that can be found in the GI system. They are used to pinpoint specific findings and that can also be used to identify a number of disorders. Furthermore, is two common GI screening techniques currently employed to find these disorders presented. We will also discuss how medical data is essential for CAD systems that can be used to improve patient health.

2.1.1 The Gastrointestinal Tract

The gastrointestinal (GI)-tract, also called the digestive tract or alimentary canal, is the pathway of the digestive system from the mouth to the anus. The GI is divided into two main parts, the upper GI-tract, and the lower GI-tract. The upper GI-tract consists of the mouth, pharynx, esophagus, stomach, and duodenum. While the lower GI consists of most of the small intestine and all of the large intestine. The upper and lower GI-tract can be seen in Figure 2.1. The GI-tract is an essential part of the human body that plays a crucial role in the digestion and absorption of nutrients. The GI-tract is also susceptible to a variety of medical conditions, such as polyps, ulcers, inflammatory bowel disease, and cancer. We will take a closer look at some of what we can find in the

GI-tract in the following sections.

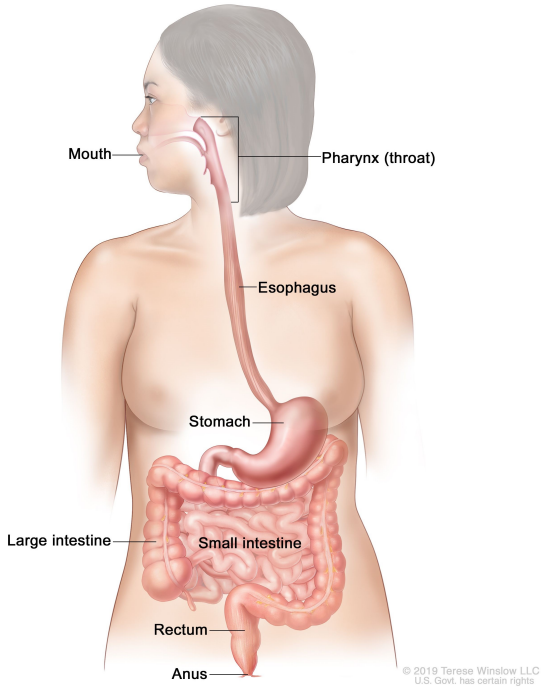


Figure 2.1: Illustration of the GI-tract [11].© (2023) Terese Winslow LLC, U.S. Govt. has certain rights.

2.1.1.1 Pathological Findings

Infections, inflammatory illnesses, autoimmune disorders, genetic abnormalities, and tumors are just a few of the factors that can lead to pathological symptoms in the GI-tract. However, abnormalities can appear in any area of the GI-tract, from the mouth to the anus. The colon is the most prevalent site for GI pathology. Medical or surgical intervention, continuous monitoring, and surveillance for the emergence of complications or cancer are just a few of the substantial consequences that pathological findings in the GI-tract might have for patient management. Gastroenterologists, pathologists, surgeons, radiologists, and other medical specialists must all work together to effectively manage GI pathology.

2.1.1.1.1 Polyps

Polyps are an abnormal outgrowth of tissue that can be found in several locations in the GI-tract, but most commonly in the colon region. Polyps can grow into two different shapes sessile or pedunculated polyps [12]. Sessile polyps are harder to detect during screening as they lie flat against the surface of the colon's lining. Polyps that are not easy to remove and polyps, in general, are often elevated before removal [13]. Pedunculated polyps hang from a stalk attached to the colon wall. We further divide polyps into neoplastic and non-neoplastic. Neoplastic polyps cover adenomatous polyps and serrated polyps with adenomatous polyps being the most common type of polyps. Non-neoplastic

polyps cover inflammatory polyps, hyperplastic polyps, and hamartomatous polyps where all of which have a low chance of becoming cancerous. Besides the type of polyps are also sizes important. Around 1% of polyps less than 1 cm in diameter are cancerous, 10% of polyps between 1-2 cm in size, and 50% of polyps larger than 2 cm [14]. An example of a polyp can be seen in Figure 2.2.

Figure 2.2: Example of a polyp in the lower GI-tract from Kvasir-SEG [15].

2.1.1.1.2 Ulcerative Colitis

Chronic inflammatory bowel illness called ulcerative colitis primarily affects the colon and the rectum. Exmaple of Ulcerative Colitis is shown in Figure 2.3.

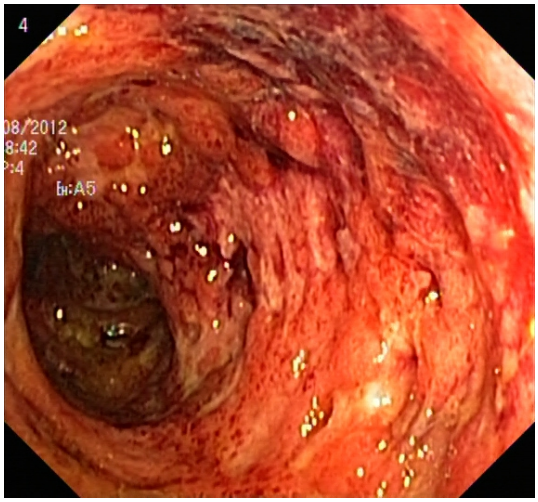


Figure 2.3: Example of ulcerative colitis in the GI-tract with grading 3 from HyperKvasir [16]. HyperKvasir grades ulcerative colitis from 0 to 3, with 3 being the most severe.

Inflammation, ulceration, and bleeding in the colon and rectum are the defining characteristics of ulcerative colitis in the GI-tract. The inner lining of the colon and rectum may look bloated, inflamed, and frail during an endoscopy, and it may also appear red or pink because of increased blood flow to the area. It is possible to have sores or ulcers that are open and bleed or create mucus. The colon and rectum

are frequently continuously affected by inflammation and ulcerations in a diffuse and continuous manner. Depending on the severity of the condition and the particular patient, the level of inflammation and ulceration may vary. In moderate cases, the rectum or a small piece of the colon may only experience inflammation and ulceration, whereas in severe cases, the entire colon may be affected. In some instances, the inflammation may impact deeper tissue layers in addition to the colon's lining, which can result in consequences like strictures or perforations.

2.1.1.2 Anatomical Landmarks

We can better comprehend the organization and operation of this complicated system by using Anatomical Landmarks in the GI-tract as essential reference points. The GI-tract contains several significant anatomical landmarks, including the esophagus, stomach, small intestine, large intestine, rectum, and anus. These landmarks are crucial for waste removal as well as for the digestion and absorption of meals. They are essential for the body's immunological defense since the GI-tract has a large quantity of lymphatic tissue. The diagnosis and treatment of certain GI illnesses, including inflammatory bowel disease, infections, and malignancies, depend on having a thorough understanding of the anatomy and function of these landmarks.

2.1.1.2.1 Ileum

The Ileum is the last section in the small intestine to connect to the large intestine shown in Figure 2.4.

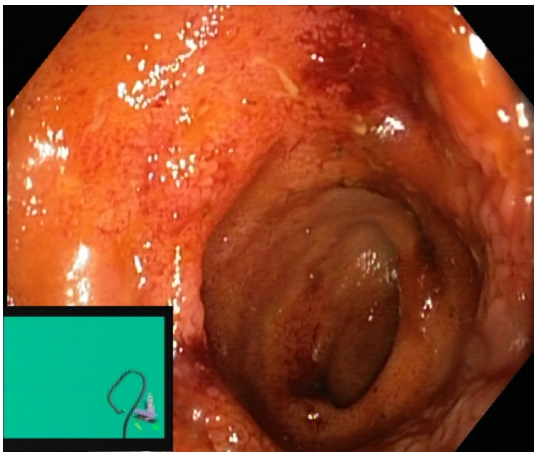


Figure 2.4: Example of the ileum in the lower GI-tract from HyperKvasir [16].

Its primary function is to take in nutrients from food. It has Peyer's patches, which are lymphatic tissue, which aids in infection defense. Villi and microvilli, which resemble little fingers and aid in nutrition absorption, cover the lining of the ileum. The large intestines' big ileocecal valve divides the ileum from it and regulates the passage of food that has been digested. Crohn's disease, infections, and tumors are a few conditions that can damage the ileum. Endoscopy and imaging studies may be

used in the diagnosis. Depending on the exact problem, treatment options could involve either medicine, surgery, or a combination of the two.

2.1.1.2.2 Cecum

The intersection of the ileum and colon is where the cecum, the first section of the large intestine, is situated shown in Figure 2.5.

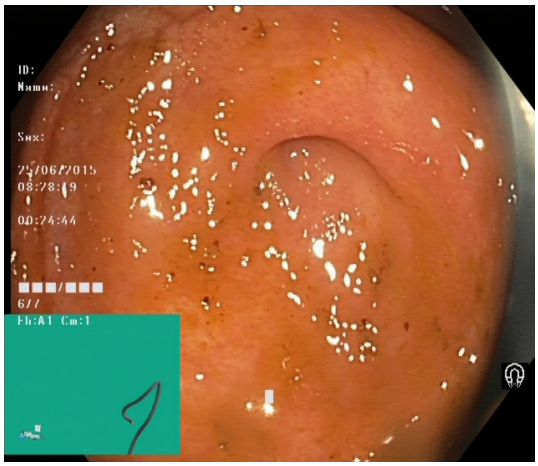


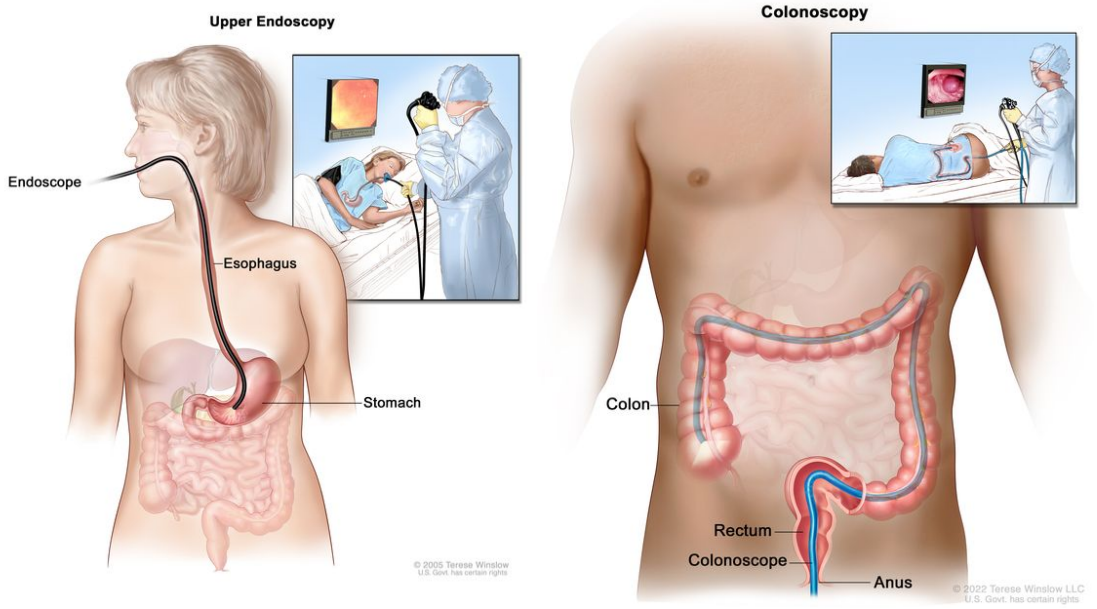
Figure 2.5: Example of the cecum in the lower GI-tract from HyperKvasir [16].

Bacteria in the cecum aid in the breakdown of fiber and complex carbohydrates that cannot be broken down in the small intestine. The cecum in humans is relatively tiny and plays a small part in digestion. It is however still prone to numerous illnesses like inflammation, infections, and tumors. Imaging tests like a CT scan or an MRI may be used to diagnose cecal disorders, and a biopsy may also be required during an endoscopic examination to provide a firm diagnosis. Depending on the exact problem and its severity, treatment options for cecal disorders may include drugs, surgery, or a combination of the two.

2.1.2 Gastrointestinal Endoscopy

GI endoscopy is a medical procedure where the GI is examined for abnormalities such as disease, infection, and other conditions. The instrument used for the procedure is called an endoscope, which is essentially a thin, long flexible tube attached to a small camera at the end for looking at tissues inside the body. The endoscope is inserted either through the patient's mouth (gastroscopy) or anus (colonoscopy). The procedure for gastroscopy and colonoscopy is illustrated in Figure 2.6.

During the procedure, the endoscope is carefully guided through the GI-tract by a trained healthcare provider, who can view the images from the camera on a screen in real-time. The endoscope also allows for the collection of tissue samples or biopsies, which can then be analyzed in a laboratory for further diagnosis.



(a) Gastroscopy

(b) Colonoscopy

Figure 2.6: Illustration of endoscopy procedures, either as gastroscopy [17] or colonoscopy [18] procedures. © (2023) Terese Winslow LLC, U.S. Govt. has certain rights.

Endoscopy is considered a safe and effective procedure that can help diagnose a wide range of GI conditions, including ulcers, polyps, tumors, and inflammatory bowel disease. It can also be used to monitor the progress of treatment and assess the effectiveness of the medication. Despite its many benefits, endoscopy does carry some risks, such as but not limited to bleeding, infection, and perforation of the GI-tract. It is therefore important to discuss the procedure beforehand with a healthcare provider to discuss associated risks.

2.1.3 Computer Aided Diagnosis

Computer aided diagnosis (CAD) is an innovative technology that is transforming the medical field by making it possible to diagnose different diseases more effectively and accurately. CAD systems analyze medical images and help healthcare professionals identify and diagnose various illnesses by using state-of-the-art ML algorithms and computer vision techniques. By examining mammograms and CT scans, respectively, CAD has been widely employed in the area of radiology to aid in the detection of early symptoms of cancer, including breast cancer[19] and lung cancer [20]. Additionally, CAD systems have been created to assist in the diagnosis of additional conditions like osteoporosis, Alzheimer's disease, and cardiovascular disease.

One of the main benefits of CAD is that it can aid medical professionals in providing quicker and more accurate diagnoses, which can improve patient treatment. This can be done by automating some diagnostic steps, like image processing and report preparation can CAD also lessen the burden on medical professionals. It is crucial to remember that CAD does not replace human skill [6], and healthcare professionals should always rely on

their clinical judgment when establishing diagnoses and selecting treatments. Overall, does CAD have the potential to completely change how we identify and treat different diseases, and its application is anticipated to increase over the next few years as more sophisticated tools and methods are created.

2.2 Machine Learning

Machine learning (ML) is a sub-field of artificial intelligence (AI) that uses statistical modeling and algorithms to enable computer systems to "learn" based on patterns and inference without explicit instructions [21]. ML can be used in many different ways and for multiple purposes. ML models usually fall into three major groups: supervised learning, unsupervised learning, and reinforcement learning.

2.2.1 Supervised Learning

Supervised learning is a group of algorithms that learn from labeled data. The training is carried out through an iterative process by predicting a sample and comparing them to the ground truth. The model then updates its weights based on how incorrect the prediction was. This process is repeated until the model either stops improving or when it reaches a set amount of iterations. Common applications for supervised learning are image classification, segmentation, speech recognition, and language translation. Some of the most well-known supervised learning algorithms are Support Vector Machines (SVMs)[22], MLP, and CNN.

2.2.2 Unsupervised Learning

Unsupervised learning is a group of algorithms that learn from unlabeled data. The goal is to find some pattern in the provided data and learn a task related to this pattern. Traditionally, cluster analysis has been the most common within unsupervised learning. These algorithms analyze the unlabeled data and try to find some common patterns between the data points and divide similar data into groups that are called clusters. Unsupervised learning can greatly reduce the amount of time needed to label data, which is beneficial in fields where we lack experts or labeling is costly. Some of the most well-known cluster analysis algorithms in unsupervised learning are k-means clustering [23] and hierarchical clustering.

2.2.3 Reinforcement Learning

Reinforcement learning algorithms use an agent where the goal is to maximize their reward in a given environment. The agent is either rewarded for reaching a goal or penalized if it does not. In other words, reinforcement learning is inspired by behaviorism, where the agent "learns" from the action it takes and given states. Common

applications for reinforcement learning are robot motion control, swarm intelligence [24], and playing Atari games [25] and Go [26]. Popular algorithms within reinforcement learning are State-Action-Reward-State-Action (SARSA) [27], Q-learning [28], and a Deep Q-Network (DQN) [29].

2.3 Neural Networks

Artificial Neural Networks or Neural networks are computational models inspired by the biological networks in the human brain. Historically, the development of these networks has been heavily inspired by biology, but has since diverged and taken more principles from engineering to achieve better results for solving ML tasks. In this section, we take a closer look at different types of neural networks, their layout, and how they learn. It will be especially important to take note of models typically used for image generation and image classification, which will be our main challenge in this thesis.

2.3.1 The Perceptron

The perceptron is an algorithm for supervised learning of binary classifiers introduced in 1958 by Frank Rosenblatt [30]. It is a linear classifier that takes a set of elements and then decides which of our two classes our element belongs to. The single-layer perceptron is the simplest feedforward neural network that is a linear classifier. The network does only consist of input and output nodes and does therefore not have hidden layers. Equation 2.1 shows us the function the perceptron uses to determine which class our element belongs to.

$$f(x) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

The function uses some weights w and input x plus some bias that maps to either 0 or 1. In short, is the perceptron built up of three parts; inputs $\mathbf{x}$, the weights $\mathbf{w}$, and the unit step activation function, shown in Figure 2.7.

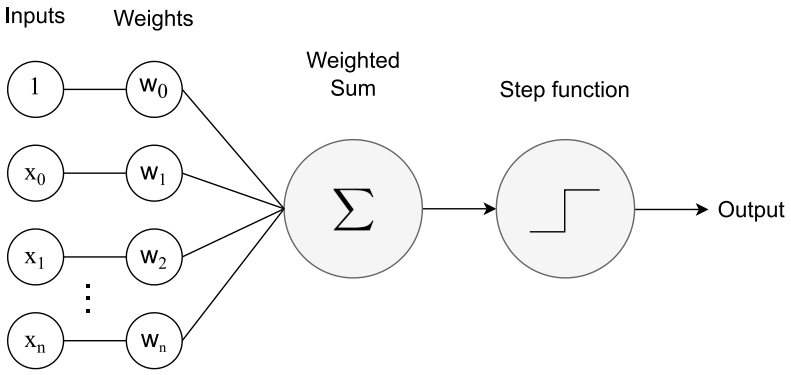


Figure 2.7: Schematic of the original perceptron.

2.3.2 Multilayer Perceptron

The MLP was first described by McCulloch and Pitts in 1943 [31]. Together they created a computational model for neural networks based on algorithms called threshold logic. In contrast to the single-layer perceptron, which can only learn linear classification, the MLP can learn functions that are also non-linearly separable. The hidden layers in the MLP allow for non-linear transformations of the input data, enabling the network to learn more complex decision boundaries. MLPs can therefore handle the XOR problem, which is not linearly separable. This is possible since the MLP is comprised of at least three layers, one input layer, one or more hidden layers, and one output layer, shown in Figure 2.8.

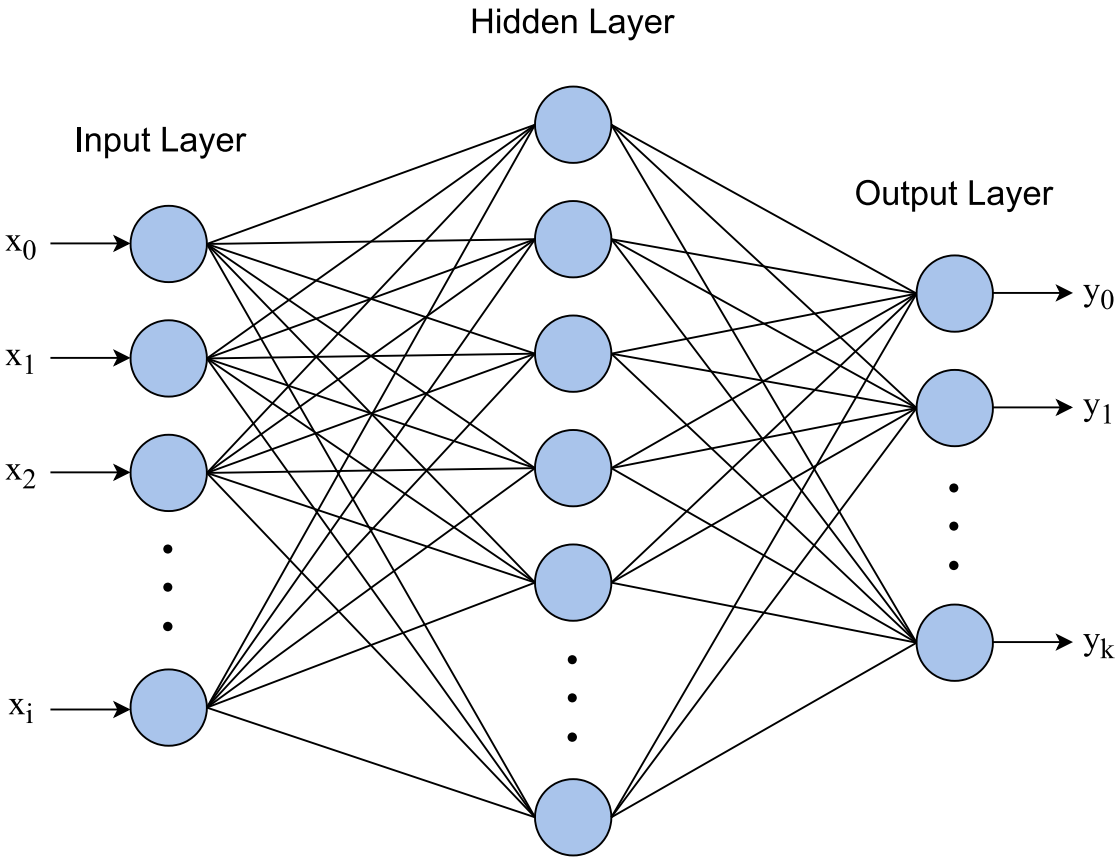


Figure 2.8: Illustration of a MLP network with one input layer, one hidden layer, and one output layer.

Over the years, various modifications and improvements have been made to the MLP, including the addition of regularization techniques to prevent overfitting, the use of different activation functions such as the rectified linear unit (ReLU), and the introduction of DL architectures such as CNN and recurrent neural networks (RNNs).

2.3.3 The Life Cycle of a Neural Network

The previous section described the architecture of MLP and how it can learn non-linear functions. This section on the other hand will focus on how a neural network is trained and learns through updating weights between layer connections. We will take a closer look at some activation functions, how loss is calculated, backpropagation, and optimizer functions.

2.3.3.1 Activation Functions

To learn non-linear solutions is some activation function needed so that our network does not collapse into a two-layer input-output network. The activation function is applied to our outputs to determine if they should fire or not. Various activation functions are used for different networks to solve different problems. We will introduce some of the most common activation functions and discuss their strength and weaknesses when applied to a neural network.

The sigmoid function is presented in Equation 2.2. It usually returns a value in the range of 0 to 1. This property makes it useful for probability problems.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

The Rectified Linear Unit(ReLU) is an activation function that is positive with input larger than zero. The ReLU is shown in Equation 2.3 with values larger than zero is the same as the input and zero otherwise. A problem with the standard ReLU is that all negative values become zero and can therefore cause problems when a lot of neurons only output zero. This effect is called dying ReLU and is something that the Leaky ReLU tries to mitigate.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The Leaky Rectified Linear Unit (Leaky ReLU) was introduced in 2013 by Maas, Hannun, and Ng [32]. The Leaky ReLU is the same as the standard ReLU expects for values smaller than zero, where it has a small negative slope. To achieve this negative slope is a small value, traditionally 0.01 multiplied with the input x shown in Equation 2.4 for values smaller than zero. The Leaky ReLU resolves the issue with vanishing gradients in this way since all negative values map onto some small negative values rather than zero.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases} \tag{2.4}$$

2.3.3.2 Cross Entropy

Cross entropy is the most common algorithm for calculating loss. We can see our loss in Equation 2.5 where y_i is the true output of class i and $\hat{y}_i$ is the predicted output of class i, summed over all classes N. This means that the loss will always be non-negative and closer to zero the better the network performs. A loss of zero would thus reflect the network making a perfect prediction. Cross entropy is frequently used for multi-class classification.

$$Loss = -\sum_{i=1}^{N} y_i \cdot \log \hat{y}_i \tag{2.5}$$

2.3.3.3 Optimizers

The goal for the optimizer is to minimize the loss in the network, this is accomplished through tuning our weights in the network. The algorithm used to update the network is called backpropagation which calculates the gradient of each node in the network and then uses the chain rule to update the weights. This is performed with a forward and backward pass, with the optimizer looking at the predicted output compared to the ground truth. Several optimizers can be used to train a neural network, and we will briefly introduce one of them in the next section.

One of the most used optimizers is stochastic gradient descent (SGD) [33]. The algorithm replaces the actual gradient with an estimation of the gradient for the entire dataset. By doing this, SGD can run iteration faster in high-dimensional optimization in exchange for a lower convergence rate which can cause us to overshoot the local minimum. It is however possible to use a technique known as momentum to reduce the effect of overshooting in SGD. This is accomplished by gradually decreasing the learning rate of the optimizer as the number of epochs increases. In Equation 2.6 can we see the formula for SGD.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)}) \tag{2.6}$$

The AdamW [34] optimizer is a variation of the Adam [35] optimizer, which is a commonly used SGD optimization algorithm. It was introduced in 2018 by Loshchilov and Hutter as a modification of the original Adam algorithm to prevent overfitting and improve generalization performance. The W in AdamW stands for weight decay, which is a regularization technique used to prevent overfitting by adding a penalty term to the loss function that discourages large weights. The AdamW optimizer essentially incorporates weight decay directly into the optimization process, making it more effective than using weight decay as a separate hyperparameter. It has been shown to outperform other popular optimizers such as SGD with momentum and Adam on a variety of DL tasks.

2.3.4 Convolutional Neural Networks

Similar to MLPs is CNNs also feed-forward networks that learn through backpropagation by updating their weights. MLPs are not particularly good for image classification as well as fully connected, meaning that our parameters and depth model grow rapidly when increasing the size of the input. CNN on the other hand, is only connected to the local region of the input [36]. This makes it so that CNNs can understand spatial relations between pixels of images and therefore much better suited for complicated images. Keeping the number of parameters low while being able to express complex models is the biggest reason for convolutional networks' success. The two concepts used to achieve this improvement compared to the traditional network is convolutional and pooling operations.

2.3.4.1 Convolutional Layers

The convolutional layer is the most vital part of a CNN. It is here that the majority of computation is performed, requiring input data, a filter, and a feature map. To do the convolution operations is a filter with kernel size and spatial size used on input to produce an output. Every filter slides across the input in the forward pass using the dot product between the weighted filter and input. The result of this convolution operation is what we refer to as the feature map. An example of this operation is illustrated in Figure 2.9 with a filter of size of $3 \times 3 \times 1$ slid across a $4 \times 4 \times 1$ input image producing a $2 \times 2 \times 1$ feature map.

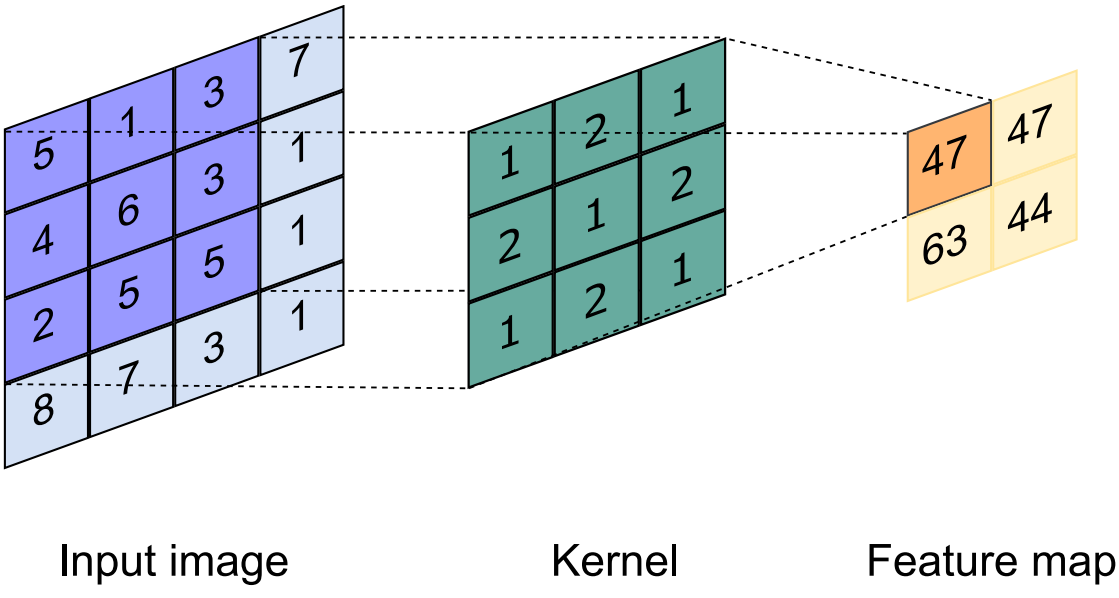


Figure 2.9: Diagram showing an example of convolution operation with input image $4 \times 4 \times 1$ using a kernel of size $3 \times 3 \times 1$ and stride of 1.

Instead of using the dot product, the convolution layer uses the convolution operation. Thus, the output vector is a result of convolution between the input and a filter kernel which is then passed to an activation function. The total output of the convolutional layer can thus be interpreted as a 3D tensor giving us all feature maps. A slice of this

3D tensor is then a 2D tensor which is a single feature map.

2.3.4.2 Pooling Layers

Pooling layers also called downsampling are commonly found in CNNs. Its purpose is to reduce the spatial size of the data given, in doing so is the number of parameters and computations needed by the network reduced. Typically, the purpose is to either reduce processing or memory costs. The two most common types of pooling are max and average pooling. Max pooling is shown in Figure 2.10 and average pooling is shown in Figure 2.11.

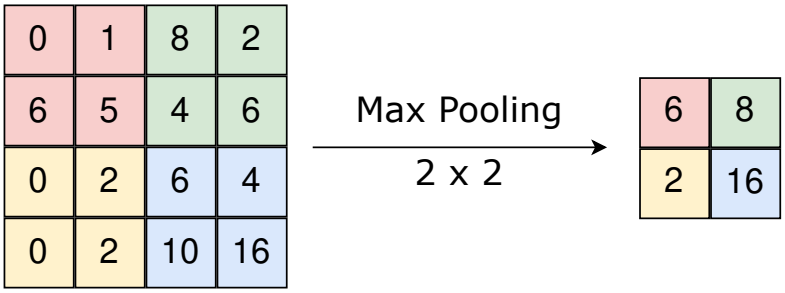


Figure 2.10: Max pooling performed on a 4×4 matrix with stride and pool size of 2.

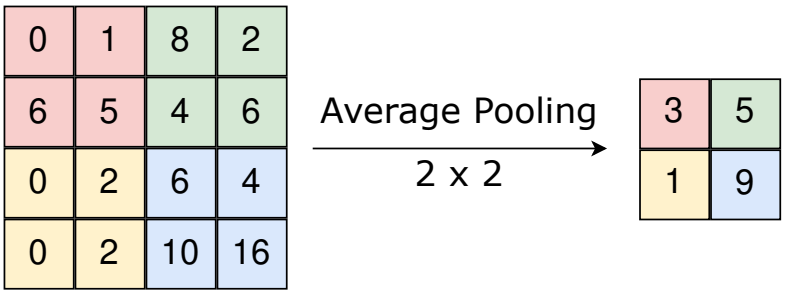


Figure 2.11: Average pooling performed on a 4×4 matrix with stride and pool size of 2.

Pooling works similarly to the convolution operation with a sliding window over the represented data. The parameters used for pooling are generally a filter size of 2 and a stride of 2. There are no weights tied to the pooling layer and the layer thus only routes gradients back without changing them during backpropagation.

2.3.4.3 U-Net — Network Architecture

U-Net is a CNN that was developed for biomedical image segmentation. It was developed at the University of Freiburg and its paper was published in 2015 [37]. The U-Net architecture originally stems from a fully convolutional network. The U-Net architecture consists of two parts a downsampling part and an upsampling part which results in u-shaped architecture illustrated in Figure 2.12.

The first part of the U-Net architecture is the downsampling also known as the contracting path similar to an encoder. This part takes in an input image and transforms

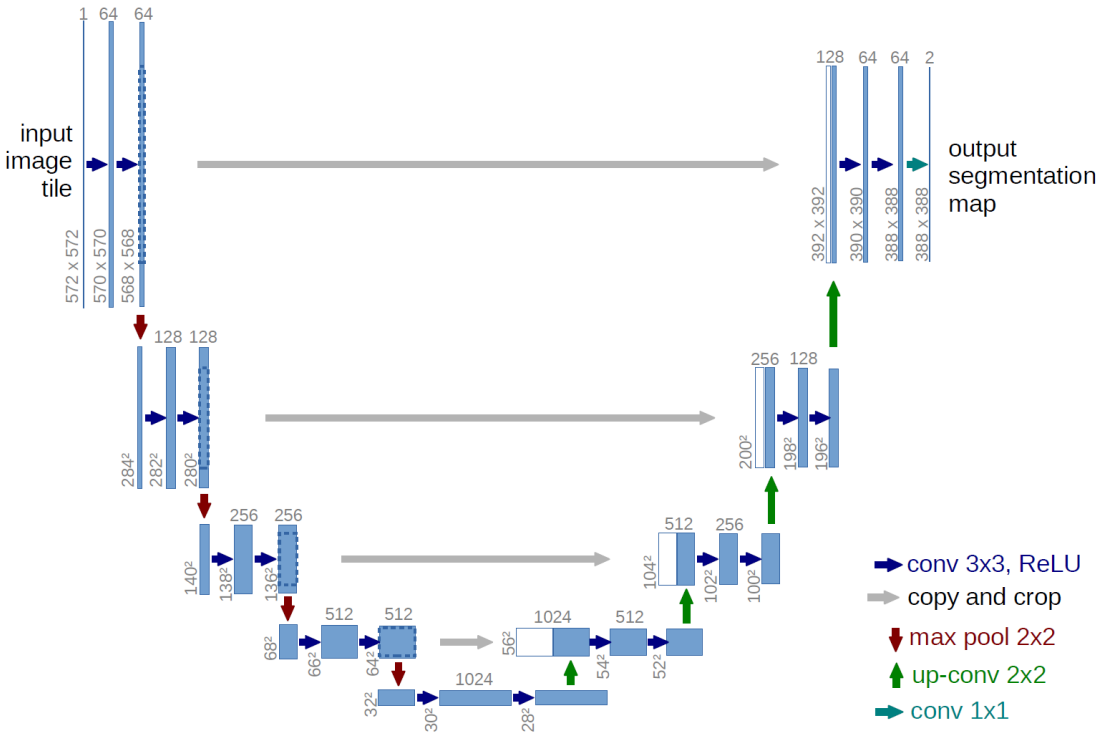


Figure 2.12: Original U-Net architecture published in 2015 [37]. Today are typically modified versions of this architecture used.

it into high-level features in the feature space. The exact location of the segmented objects learned in this part is not known, and this is where the upsampling part gives the spatial details needed.

The second part is the upsampling also known as the expansive path similar to a decoder. The upsampling part takes in the result of the downsampling part and fills in where the features are situated in finer detail with the help of skip connections. Skip connections for U-Net are illustrated in Figure 2.12 as grey arrows. Skip connections in an encoder-decoded architecture such as U-Net can recover fine-grained details in the prediction.

Even though the initial paper that introduced the U-Net architecture was to solve the task of biological image segmentation it is currently used for other tasks with several variations of the original architecture. The downsampling part of the network utilizes convolution and max pooling. The upsampling part consists of transposed convolution also known as upsampling, convolution, and skip connections from the downsampling part. The input from a skip connection corresponds to the output from the corresponding downsampling layer. Skip connections are concatenated with the output from the transposed convolution.

2.3.5 Deep Learning

Deep learning (DL) is a sub-field within ML that mainly uses neural network architectures. DL uses the term "deep" meaning it is comprised of three or more layers. In earlier sections, has MLPs and CNNs were introduced, which use three or more layers and are therefore DL algorithms. DL is not only restricted to supervised learning but can also be used for unsupervised learning and reinforcement learning. DL has achieved significant success in fields like computer vision and natural language processing, but challenges still exist such as the requirement for large amounts of labeled data and specialized hardware. Recent research focuses on developing more efficient algorithms. Despite DL limitations has it transformed the field of ML and AI with potential applications continuing to grow. Ongoing research and development in DL will likely lead to even more breakthroughs in the years to come.

2.3.6 Regularization

Regularization is a technique [38] used in ML to prevent overfitting and improve the generalization of models. Overfitting occurs when a model fits too closely to the training data while simultaneously performing poorly on new unseen data. The goal of regularization is to encourage the network to learn a simpler, more general solution to the problem rather than memorizing the training data. The penalty term applied by regularization adds a cost to the model for having high-magnitude coefficients or weights, which are associated with more complex models. By adding this penalty, regularization helps to prevent overfitting and improves the generalization ability of the model.

There are many different types of regularization techniques, including L1 regularization and L2 regularization. The selection of which regularization technique to use will depend on the specifics of the problem and the model being trained. Regularization is a crucial technique that is often used in conjunction such as, such as data augmentation and dropout, and more to improve the generalization ability of models.

2.3.6.1 Dropout

Dropout was first introduced in 2014 by Srivastava et al. [39] to prevent overfitting in neural networks. The main idea behind dropout is to ensure that no single neuron has too much influence over a model's prediction. Dropout works by randomly cutting a certain amount of connections in a layer during training. The number of cut of connections can vary, but at the time of writing are dropout usually in the interval of 10% to 50% as values higher than that can result in a significant reduction in model performance. An example of dropout in a single hidden layer is illustrated in Figure 2.13.

Using dropout makes the network so that it cannot just rely on a few weights during training, as there is a possibility that the connection from the weight will be cut off. This simplicity makes the network's strong weights more evenly distributed and therefore more robust.

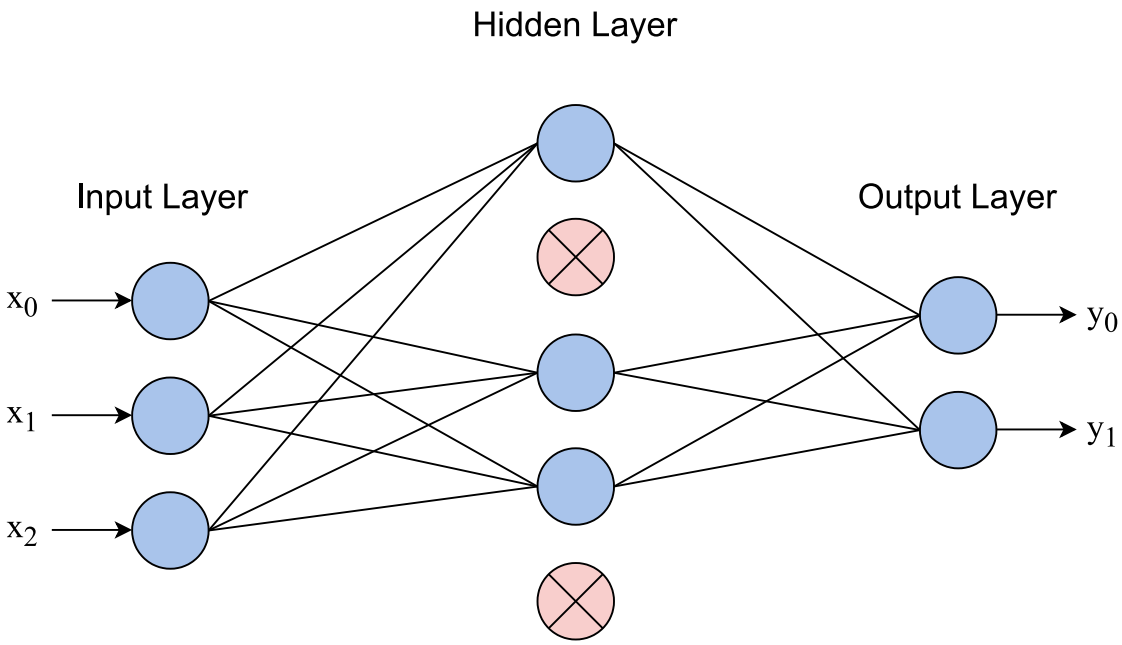


Figure 2.13: Example of how a network with dropout $p = 0.4$ can look like. Dropped neurons from the top to bottom are neurons 2 and 5 in the hidden layer.

2.3.6.2 Data Augmentation

Data augmentation is a commonly used regularization technique used to reduce overfitting and thus improve the generalization of a model. Data augmentation artificially increases the size of the training dataset and exposes the model to different variations of the same data. In data augmentation, existing data is transformed by applying various operations, such as but not limited to rotation, flipping, cropping, scaling, and adding noise. These operations simulate different perspectives and variations of the same data, making the model more robust to different types of input. By training on a larger and more diverse dataset is the model less likely to memorize the training data and instead learns to recognize the underlying patterns and features of the data.

Data augmentation is especially useful in computer vision and image classification tasks [40], where it can be applied to images to increase the size of the training dataset. However, it can also be applied to other types of data, such as text and audio, by applying operations that simulate variations in the data.

While data augmentation can be an effective regularization technique it is important to choose appropriate augmentations that are representative of the data and the given problem. Furthermore, data augmentation should not be used instead of obtaining additional labeled data or using alternative regularization methods. Instead, it is often used in combination with other regularization techniques to achieve optimal performance.

2.4 Generative Models

Generative models fall under the unsupervised learning category. Generative models have gained a lot of popularity over the last years, especially GANs and Generative Pre-trained Transformer (GPT). Common for these models is that they learn the true data distribution of the training data in order to generate new data with a similar distribution which is categorized as synthetic data. The models must also have a stochastic element to make individual samples differ from each other, generative models are therefore probabilistic rather than deterministic. Common applications for generative models include text-to-image translation, music generation, video generation, and image generation. Image generation has been especially used to generate human faces, also known as deepfakes.

2.4.1 Autoencoders

Autoencoders were first introduced in the 1980s by Hinton et al. [41]. An autoencoder is an unsupervised artificial neural network that learns to recreate data as close as possible to the original input. The architecture of the vanilla autoencoder is comprised of three main parts: An encoder network, a bottleneck, and a decoder network. The encoder network reduces our input dimensions and compresses our data into latent space. The bottleneck contains the compressed representation of the input data with the lowest possible dimensions. The decoder network reconstructs the data from the latent space to be as close as possible to the original data, the whole process can be seen in Figure 2.14.

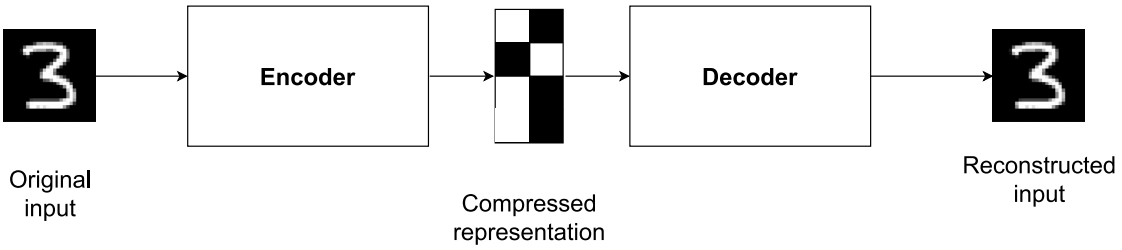


Figure 2.14: Basic structure of autoencoder with its three main parts; The encoder, latent space, and decoder.

Traditionally, autoencoders have been used for learning important features and dimensionality reduction. Types of autoencoders are vast and have recently become more used for learning generative models. In the next section, variational autoencoders are described, which is a popular type of autoencoder used for generating new data.

2.4.2 Variational Autoencoders

The VAE was introduced in 2013 by Kingma and Welling's publication "Auto-Encoding Variational Bayes" [42]. The variational autoencoder differs in the fact that it maps the input to a multivariate Gaussian distribution in the latent space and not a single point. The input is encoded into a multivariate normal distribution with mean value

μ and the variance σ^2 into the latent space. A point is then sampled from the distribution as a representation in latent space, which is then decoded to reconstruct the input. Variational autoencoder uses a statistical approach to approximate complex distributions. The encoder outputs a latent representation of parameters in the latent space for every input. The network then forces the latent distribution to be normally distributed. A common issue with a variational autoencoder is that it often produces blurry images due to its normal distribution assumption.

2.4.3 Generative Adversarial Networks

The basic GAN was introduced in 2014 by Ian Goodfellow et al. [43]. It contains two Deep Neural Networks (DNNs), one called the generator G and the second called discriminator D, where they compete in a zero-sum game, where one agent's gain is another agent's loss. GANs can learn to generate realistic images for a given training set of images if it manages to deduce a distribution from the training set.

The generator's main job is to generate synthetic data. This is achieved by taking a random noise vector as input, which can be sampled from statistical distributions, most commonly a Gaussian distribution. The discriminator's task is to distinguish between generated data and real data. This is brought about in hopes of generating data that is realistic enough to be labeled as real data by the discriminator. The basic architecture of the model can be seen in Figure 2.15. Ideally, the network should generate data that "fools" the discriminator about half of the time or when the generator and discriminator stop improving.

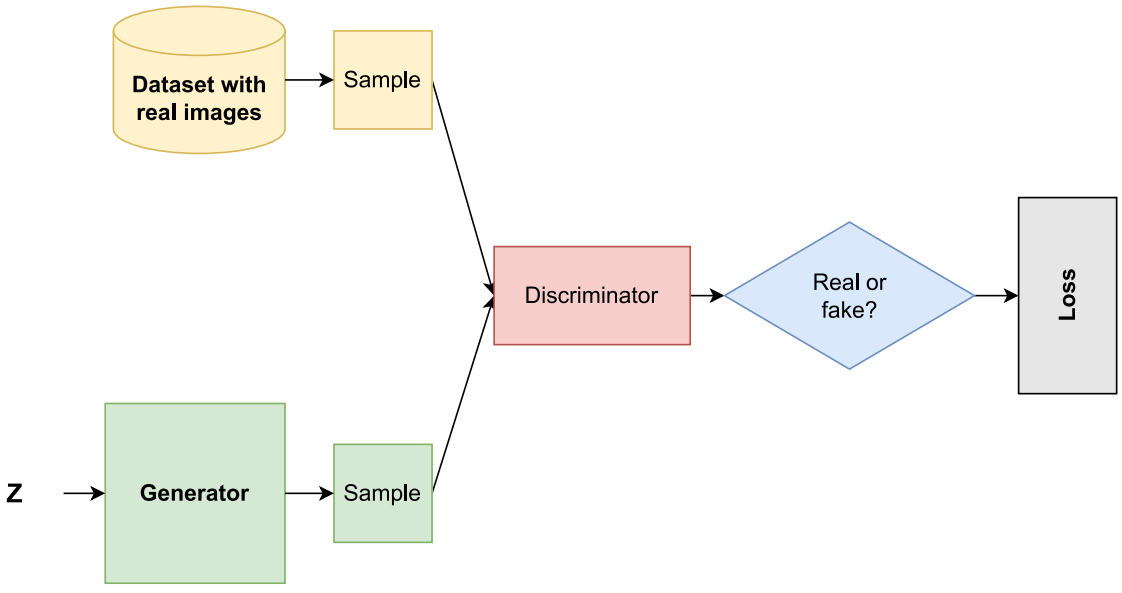


Figure 2.15: General structure of a basic GAN where **z** denotes a random latent input.

The two networks are trained simultaneously by learning each other's mistakes in hopes of generating realistic data. The generator will only be able to generate realistic images from the data distribution given by the training data.

2.4.4 Diffusion models

Diffusion models also known as diffusion probabilistic models are models that fall under the class latent variable models. Diffusion models were introduced in 2015 with motivation from non-equilibrium thermodynamics [44]. Diffusion models learn the latent structure of a dataset by modeling the way in which data is diffused through the latent space.

Diffusion models can be applied in a variety of tasks such as image denoising, inpainting, super-resolution, and image generation. Diffusion models consist of two processes the forward process which gradually adds Gaussian noise to the image, and the backward process which gradually removes Gaussian noise. These two processes are illustrated in Figure 2.16.

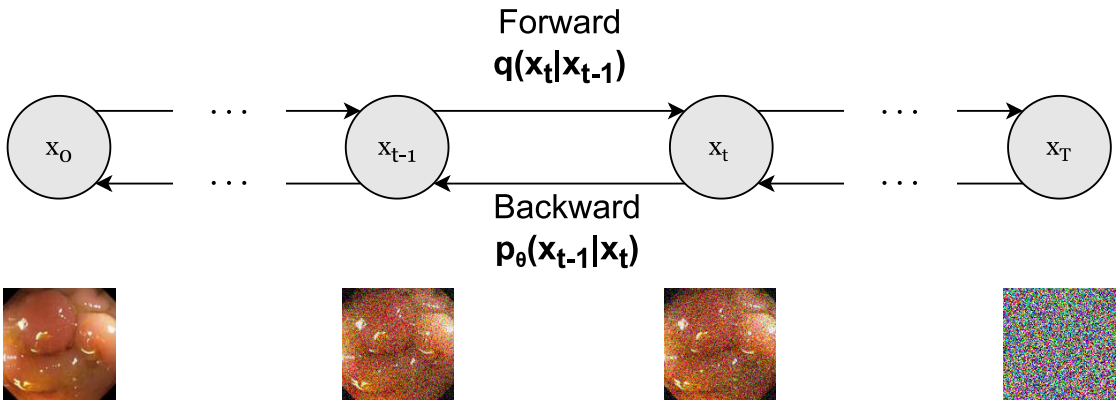


Figure 2.16: Forward and backward process of diffusion models, with t=200 and T=1000.

2.4.4.1 Denoising Diffusion Probabilistic Models

DDPM is a class of latent variable models inspired by considerations from non-equilibrium thermodynamics. DDPM have achieved high-quality image generation without adversarial training, by simulating a Markov chain for many steps to produce a sample. This makes DDPM slow for image generation, the regular amount of timesteps is 1000 (*T=1000*) as utilized in the original DDPM [45].

The reverse process $p_\theta(x_{0:T})$ is defined as a Markov chain with learned Gaussian transitions starting at $p(x_T) = \mathcal{N}(x_T; 0, I)$, the process is further explained in equation 2.7.

$$\begin{aligned} p_\theta(x_{0:T}) &:= p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \\ p_\theta(x_{t-1}|x_t) &:= \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \end{aligned} \tag{2.7}$$

The forward process or the diffusion process, on the other hand, approximates posterior $q(x_{1:T}|x_0)$ to a fixed Markov chain that gradually adds Gaussian noise to

the data with a variance scheduler $\beta_1, \ldots, \beta_T$ until data is nearly an isotropic Gaussian distribution, this process is further described in equation 2.8.

$$\begin{aligned} q(x_{1:T}|x_0) &:= \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ q(x_t|x_{t-1}) &:= \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I) \end{aligned} \tag{2.8}$$

Both the training and sampling processes are described in Table 2.1 as pseudo-code. The key takeaway is that we either add Gaussian noise during training or remove Gaussian noise during sampling.

Algorithm 1 Training	**Algorithm 2** Sampling
1: **repeat**	1: $x_T \sim \boldsymbol{\mathcal{N}(0, I)}$
2: $\boldsymbol{x_0} \sim q(\boldsymbol{x_0})$	2: **for** $t = T, \ldots, 1$ **do**
3: $t \sim$ Uniform($\{1, \ldots, T\}$)	3: $\mathbf{z} \sim \boldsymbol{\mathcal{N}(0, I)}$ if t >1, else $\mathbf{z = 0}$
4: $\boldsymbol{\epsilon} \sim \boldsymbol{\mathcal{N}(0, I)}$	4: $\boldsymbol{x_{t-1}} = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x_t} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x_t}, t)\right) + \sigma_t \mathbf{z}$
5: Take gradient descent step on $\nabla_\theta \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\boldsymbol{x_0} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t)\|^2$	5: **end for**
6: **until** converged	6: **return** $\boldsymbol{x_0}$

Table 2.1: Pseudo code for the training and sampling procedure for DDPMs from the original paper [45].

2.4.4.2 Denoising Diffusion Implicit Models

To accelerate sampling for diffusion models was Denoising Diffusion Implicit Models (DDIM) [46] introduced, as a more efficient class of iterative implicit probabilistic models with the same training procedure as DDPM. This is achieved through generalizing DDPM via a class of non-Markovian diffusion processes. These non-Markovian processes can correspond to generative processes that are deterministic. Sampling speeds for DDIM are anywhere from 10x to 50x faster than DDPM in terms of wall-clock time, the increase in sampling speed is brought forth by trading off computation for sample quality.

2.4.5 Trilemma

The trade-off between three important generating model sample quality, sample variety, and computing efficiency is known as the generative trilemma. The trilemma is a fundamental problem faced by all generative models, including GANs, VAEs, and Diffusion Models.

GANs as introduced in Section 2.4.3 are a class of generative models that learn to generate synthetic data by training two neural networks against each other. While GANs can produce high-quality samples with sharp details, they can suffer from mode collapse,

where the generator learns to produce only a limited set of samples without exploring the full space of possible samples. This can result in poor sample diversity.

VAEs as introduced in Section 2.4.2 on the other hand, learn to model the underlying distribution of the data and generate samples by sampling from this distribution. VAEs tend to produce more diverse samples than GANs, but they can suffer from blurry samples and limited sample quality due to the inherent trade-off between sample diversity and sample quality.

Diffusion models as introduced in Section 2.4.4 use a reverse diffusion process to generate high-quality samples and have been shown to produce the highest-quality samples among the three types of models. However, diffusion models are computationally expensive and require a large number of iterations to generate each sample, making them less efficient than GANs and VAEs.

Therefore, the generative trilemma involves a trade-off between sample quality, sample diversity, and computational efficiency, and each type of generative model has its strengths and weaknesses in addressing this trilemma. It ultimately depends on the specific application and the desired outcome which model is the most suitable for the task at hand. The trilemma is shown in Figure 2.17 with the trade-offs for the different generative models.

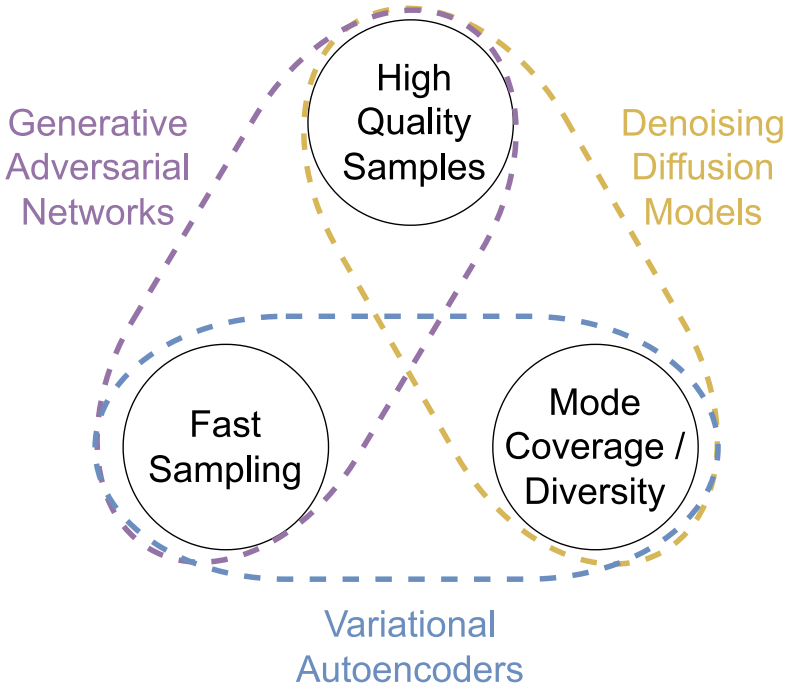


Figure 2.17: Generative models trilemma.

2.5 Summary

This chapter presented ML methods, such as DL neural networks and generative models, that are revolutionizing medical image analysis. These methods have made it possible

to develop cutting-edge approaches to medical image analysis, such as CAD systems that can assist clinicians in disease diagnosis, by automatically finding and measuring abnormalities in medical images. It is important that ML models are generalizable and may therefore need regularization to perform well on unseen data. We have talked about the importance of the GI-tract for food digestion and absorption. The esophagus, stomach, small intestine, and large intestine are only a few of the organs and tissues that make up the GI-tract. The GI-tract can be impacted by a number of illnesses, including colorectal cancer, irritable bowel syndrome, and inflammatory bowel disease. Endoscopy is a frequent medical technique used to identify and treat conditions of the GI-tract. A little, flexible tube with a camera on the end is put into the patient's digestive tract during an endoscopy. Doctors can see anomalies like polyps, ulcers, and tumors while viewing real-time images of the GI system and can even collect tissue samples for additional examination. Early detection of these anomalies is crucial for an effective treatment that can save lives.

For a variety of applications, including image synthesis has ML gained popularity. GAN and diffusion models are two of the many ML models that are used for image synthesis that has drawn a lot of interest.GANs consists of two networks a discriminator network used to examine created images and a generator network that learns to produce synthetic images. On the other hand, diffusion models employ a Markov Chain Monte Carlo strategy to iteratively improve an initial image into a final generated image. There are benefits and drawbacks to both GANs and diffusion models. While GANs are known for producing realistic and aesthetically pleasing images can they be difficult to train and are prone to mode collapse, which happens when the generator only outputs a small number of options while disregarding the larger field of possibilities. While diffusion models on the other hand can be computationally expensive to train and need a lot of memory to retain the intermediate noise samples during training, they are typically more stable and do not experience mode collapse. Both GANs and diffusion models are effective tools for creating images despite these drawbacks, and continuing research tries to address these issues and enhance their effectiveness.

Chapter 3

Methodology

The previous chapter introduced abnormalities in the GI-tract and how CAD systems can help professionals to detect these abnormalities. However, for CAD systems to be effective, a large amount of data is required. This chapter focuses on how we can generate polyp images to increase polyp samples that can be used for CAD systems. Four datasets are introduced, three being polyp datasets and one unlabeled dataset. We present two approaches to generate polyp images, one without segmentation masks and one with segmentation masks. Both approaches rely on the same underlying generative models. Lastly, we present how we train polyp segmentation models using real data or a mix of synthetic and real data.

3.1 System Specifications

The models trained in this thesis have been programmed in Python [1] and implemented using PyTorch [47], which is an open-source DL framework widely utilized in research. PyTorch was for instance used for rescaling images to size 128×128 and center-cropping before training. During training, PyTorch was used for giving images a fifty-fifty chance of flipping horizontally as well as being used to implement all models. To monitor training progression was WandB [2] utilized. This package makes it possible for us to track how long our models have trained and their losses in real time. OpenCV [3] is an open source computer vision and machine learning software library and it uses in our thesis is talked about in the next chapter in regards to image correlation. The system specifications used in the experiments can be seen in Table 3.1. The main limitation of our experiment is the hardware pertained to RAM. In all experiments was only a single GPU used at a time.

[1] https://www.python.org/downloads/release/python-385/
[2] https://docs.wandb.ai/quickstart
[3] https://opencv.org/

Hardware	
CPU	Intel Xeon Platinum 8168 @ 2.70GHz
GPU	Nvidea V100 Tensor Core
RAM	32 GB
Software	
Python	3.8.5
Pytorch	1.12.1
OpenCV	4.6.0
WandB	0.13.4

Table 3.1: System specification for both hardware and software. For more in-depth software dependencies visit the Github repository.

3.2 Data Material

In this thesis have we used 4 datasets, HyperKvasir [16], Kvasir-SEG [15], CVC-ClinicDB [48], and ETIS-Larib Polyp DB [49]. An overview of the datasets used can be seen in Table 3.2.

Dataset	Focus	Resolution	# Images
HyperKvasir [16]	Multiple	Variable	111 079
Kvasir-SEG [15]	Polyps	Variable	1 000
CVC-ClinicDB [48]	Polyps	388×288	612
ETIS-Larib Polyp DB [49]	Polyps	1255×966	196

Table 3.2: Overview of GI datasets used for training and validation.

The Kvasir datasets have mainly been used for training and some validation, and the CVC-ClinicDB and ETIS-LaribDB datasets have only been used for validation. The HyperKvasir dataset was chosen as the main dataset as it has a vast and diverse amount of data and is therefore preferred for training. All collected datasets are only to be used for research and educational purposes under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

3.2.1 HyperKvasir

The HyperKvasir [16] dataset is the largest open-source dataset from the GI-tract. The dataset can be split into four main parts; Labeled images, unlabeled images, segmented images, and annotated videos. The labeled part is comprised of 10,662 labeled images with 23 different classes. The unlabeled part is comprised of 99,417 unlabeled images. Lastly, there are also 373 videos containing different findings and landmarks, which correspond to 11.62 hours of videos and 1,059,519 video frames.

3.2.2 Kvasir-SEG

The Kvasir-SEG [15] dataset contains 1000 polyp images, their corresponding pixel-wise segmented ground truth, and a bounding box for the corresponding images stored in a JSON file. The resolution of the images in Kvasir-SEG varies from 332x487 to 1920x1072 pixels. The dataset was collected using endoscopic equipment and verified by experienced gastroenterologists from Vestre Viken Health Trust (VV) in Norway based on the original Kvasir dataset [50]. The VV is compromised of 4 hospitals and provides health care to 470 000 residents in 26 municipalities.

3.2.3 CVC-ClinicDB

CVC-ClinicDB [48] is a public dataset of frames taken from colonoscopy videos. Many examples of polyps can be seen in these frames. Each polyp frame has also been labeled with a corresponding ground truth. The ground truth is a mask that corresponds to the area in the image where a polyp is present. The 612 colonoscopy frames that make up the CVC-ClinicDB dataset were recorded with high-definition endoscopes. The frames, which include hyperplastic, adenomatous, and serrated polyps, were taken at the Hospital Clínic de Barcelona and other partnering hospitals.

3.2.4 ETIS-Larib Polyp DB

The ETIS-Larib Polyp DB [49] is a publicly accessible collection of colon polyps that have been segmented with the goal of creating and testing CAD systems for polyp detection. 16 colonoscopy videos make up the ETIS-Larib Polyp DB, from which 196 pictures of polyps were taken. The images are in JPEG format, and an XML file including annotations of the polyps written by knowledgeable gastroenterologists is included with each image.

3.2.5 Discussion of the Different Datasets

The CVC-ClinicDB dataset, for instance, includes endoscopic images of polyp tumors that were gleaned from colonoscopy videos. Similarly to this, the ETIS-LaribDB dataset includes retinal images derived from retinal video sequences. In both instances, the images from the same sequences are highly related to one another in terms of visual appearance, imaging process, and outside elements like lighting and camera position. Kvasir-SEG, on the other hand, is much more diverse when it comes to types of polyps and is therefore used for training. Examples of images and their segmentation masks can be found in Figure 3.1.

Kvasir-SEG has some biases even though it was chosen for training. It was shown to have some closely related polyps as shown in Figure 3.2 which can cause biases.

It is also worth mentioning that many images in Kvasir-SEG have artifacts, like

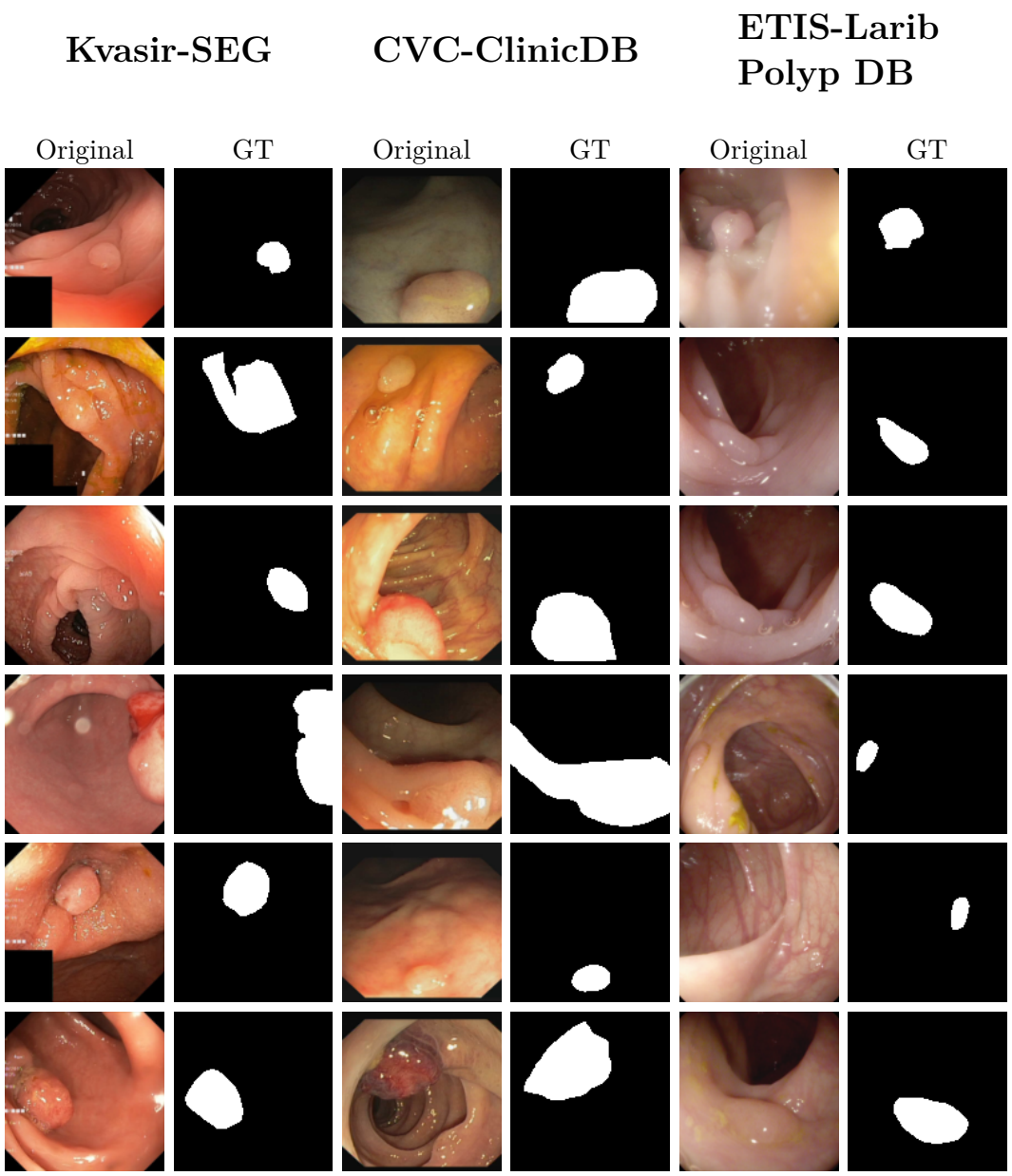


Figure 3.1: Examples of original images with their GT - Ground Truth for polyps from the Kvasir-SEG [15], CVC-ClinicDB [48], and ETIS-Larib Polyp DB [49] datasets.

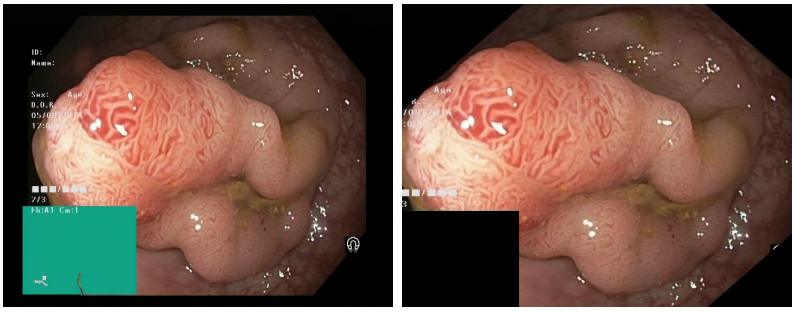


Figure 3.2: Two different images from Kvasir-SEG [15], where the right image can be seen as an augmented version of the left image. It appears that multiple parts of the image are cropped.

green boxes in the lower left corner or overlayed text, which are addressed in [51] by preprocessing the images using GANs. While using closely related images from video sequences can help with some medical image analysis tasks, such as object tracking, can it also pose some difficulties for others, such as image segmentation or classification tasks. The key issue is the risk of overfitting to unique properties of individual images within the sequence which might reduce a model's generalizability to other data.

3.3 Approaches

This section presents three approaches to generating polyp images. The approaches depend on different data and applications with their set of advantages and limitations. The most appropriate approach depends on the specific application and available resources at hand. By presenting these different approaches, readers will gain a better understanding of the complexity of generating synthetic data and the range of possible solutions.

3.3.1 Polyp Generation

The approach can be explained in the following steps:

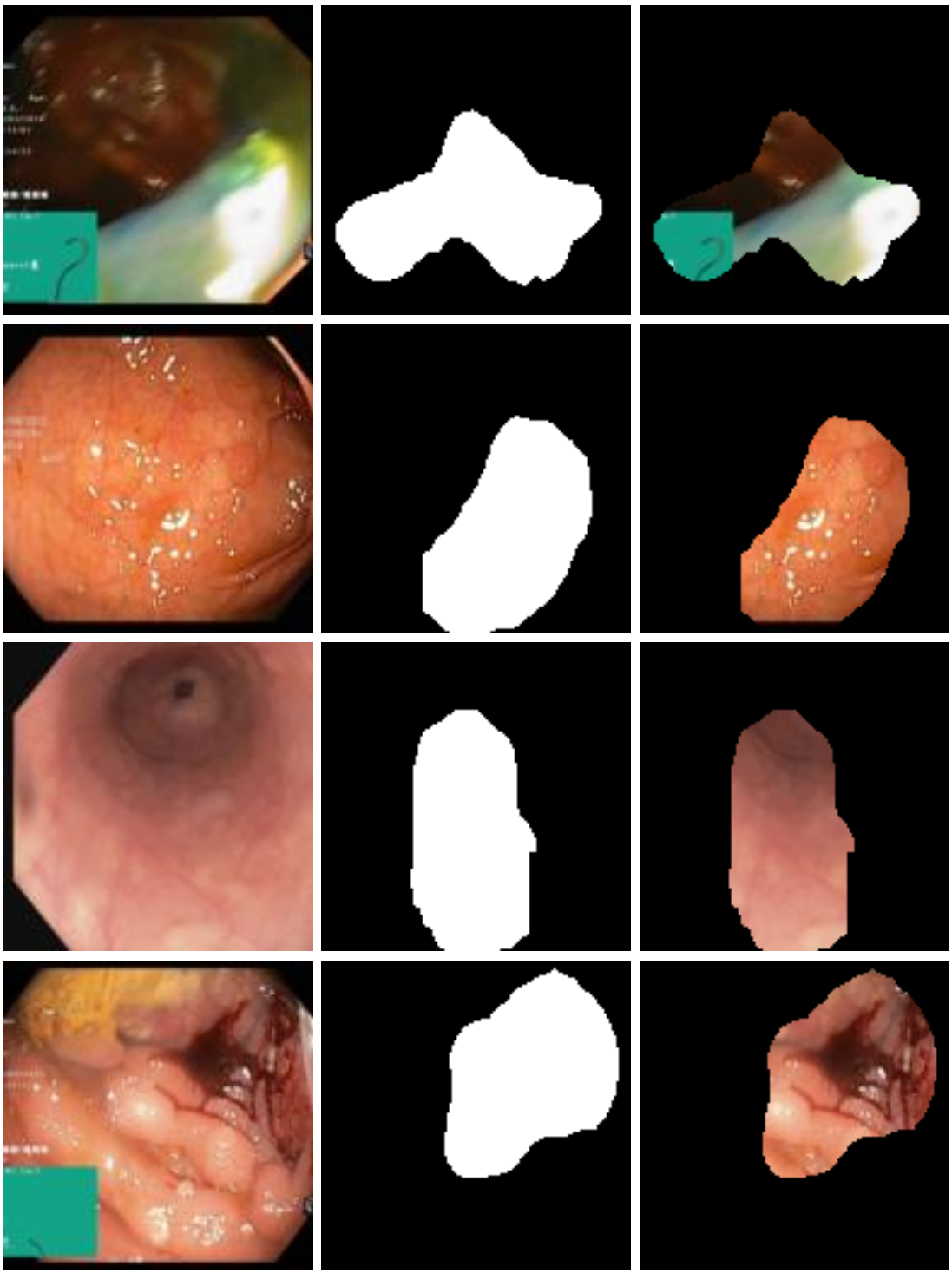
1. Diffusion model trained on a large number of unlabeled images. The training is performed so that the diffusion model has a general idea of what the GI-tract looks like. This results in a pre-trained model that can generate images similar to those found in the GI-tract. The pre-trained model weights are saved to be tuned in the next step.
2. Diffusion model from the previous step loaded. It is then fine-tuned on polyp images. The model will now produce polyp images.

Application that the generated polyps can be used in is classification tasks, for example, classifying whether an image is from the polyp class or clean colon class.

3.3.2 Polyp Generation for Segmentation

The approach can be explained in the following steps:

1. Use generated masks using a FastGAN-based [52] model. This model is trained on a large number of masked images, resulting in our first pre-trained model. Masking an image is a process of only revealing parts of an image. The pre-trained models' weights are saved to be tuned in the next step. Unlabeled images from HyperKvasir, masks generated by the FastGAN, and corresponding masked unlabeled images can be seen in Figure 3.3.



(a) Complete image (b) Mask (c) Masked image

Figure 3.3: Column (a) shows original images in unlabeled part of HyperKvasir, (b) masks generated with a FastGAN, and (c) masked image achieved by comparing (a) and (b).

2. The diffusion model trained in the previous step on real cropped-out polyps. The cropped-out polyps are obtained by removing everything in the images except the ground truth region. This is our fine-tuning toward generating realistic cropped-out polyps.
3. New diffusion model trained on a large number of images. The training is performed so that the diffusion model has a general idea of what the GI-tract looks like. This results in a pre-trained model that can generate images similar to those you can find in the GI-tract. The pre-trained model weights are saved to be tuned in the next step. Same procedure as step 1 in 3.3.1.
4. Diffusion model from the previous step loaded. It is then fine-tuned on clean colon images. The model can now produce colon images that do not contain polyps. The fine-tuned model weights are saved to be used in the next step. Same procedure as step 2 in 3.3.1, but tuned to generate a clean colon and not polyps.
5. In our final step, do first create a corresponding segmentation mask for our polyps from step 2 by thresholding. We then use our ground truth and corresponding segmentation mask with our diffusion model from step 4. This is performed to generate clean probabilistic background created for our cropped-out polyps while simultaneously having the polyp images segmentation mask.

Application that the generated polyps can be used in segmentation and classification tasks, for example, to artificially increase dataset size and diversity when training a segmentation model to improve performance.

3.4 The Art of Inpainting

Image inpainting is a method for restoring missing or damaged portions of an image. Inpainting algorithms based on DL have proven the ability to produce realistic and visually attractive images. Ensuring that the resulting images are aesthetically and conceptually cohesive is one of the primary issues in image inpainting. This calls for the method to remain consistent with the underlying structure and content of the image in addition to filling in missing pixels with colors and textures that fit the surrounding regions.

The handling of various sorts of missing information brought on by occlusions or data loss presents another challenge in picture inpainting. A variety of inpainting methods that combine various forms of information, such as context information from nearby pixels, semantic information from object recognition, and texture information from image synthesis, have been offered as solutions to this problem.

3.5 Diffusion Based Frameworks

In this section, we present models that train or depend on pre-trained diffusion models. Diffusion models can achieve better sample quality and diversity than GANs, and we

try to look at some specific implementations of how they can achieve this. In addition, we perform an investigation of how pre-trained DDPMs can be used for inpainting objectives. To inpaint is a model that has a good understanding of what to inpaint in specific domains needed. We, therefore, look at a way to adjust our DDPMs to understand new more specific data.

3.5.1 Guided Diffusion

The guided-diffusion codebase is based on improved diffusion[4] with some modifications and the same authors. The corresponding article for the guided-diffusion codebase is named "Diffusion Models Beat GANs on Image Synthesis" [53]. The presented diffusion models achieved image sample quality superior to that of state-of-the-art generative models at the time. Several diffusion models were trained and evaluated on datasets such as ImageNet [54] and 3 classes on the LSUN dataset [55] with classes: bedroom, horse, and cat. Guided diffusion supports both training unconditional and conditional diffusion models and has many different pre-trained models available.

The use of a multi-scale method for image generation is a significant architectural improvement in guided diffusion. Starting with a low-resolution image and gradually adding information at higher resolutions, the model creates images at various scales. With this method, the model is able to capture an image's overall structure as well as its minute details, producing synthesized images of superior quality. The introduction of a conditioning mechanism, which enables the model to produce images that adhere to specific restrictions, is another architectural advancement. The diffusion process is guided by the conditioning mechanism to produce images that adhere to the stated restrictions. The conditioning mechanism can take many different forms, such as a textual description or a set of input vectors. This approach is useful when creating images with certain characteristics, such as a particular design or color scheme.

Lastly, are also a number of additional architectural advancements, such as the use of skip connections to enhance information flow through the model and the use of normalization techniques to enhance the stability and convergence of the training process implemented. The visualization of a conditional diffusion model can be seen in Figure 3.4.

3.5.2 RePaint

RePaint is an inference scheme [57] introduced by et al. Lugmayr in 2022 for free-form inpainting tasks. RePaint relies on pre-trained unconditional DDPM for inpainting generation. By conditioning information in the known parts of the image and filling masked regions with probabilistic pixels, RePaint alters reverse diffusion to generate missing regions. The RePaint procedure can be observed in Figure 3.5.

The scheme does not modify the trained DDPM meaning inpainting will be of high quality and diverse. RePaint introduces a resampling procedure to avoid semantically

[4]https://github.com/openai/improved-diffusion

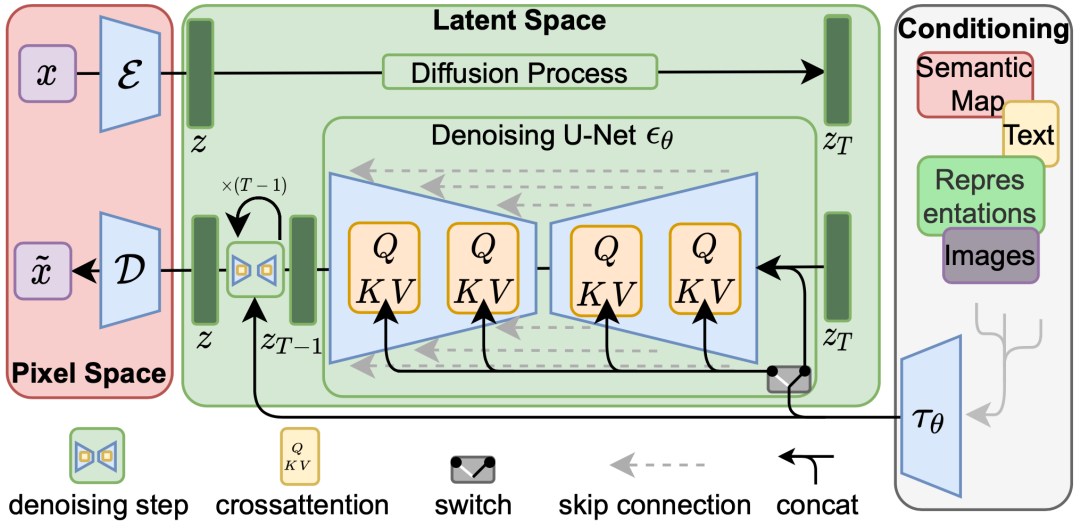


Figure 3.4: The architecture of a latent diffusion model that supports conditional generation [56].

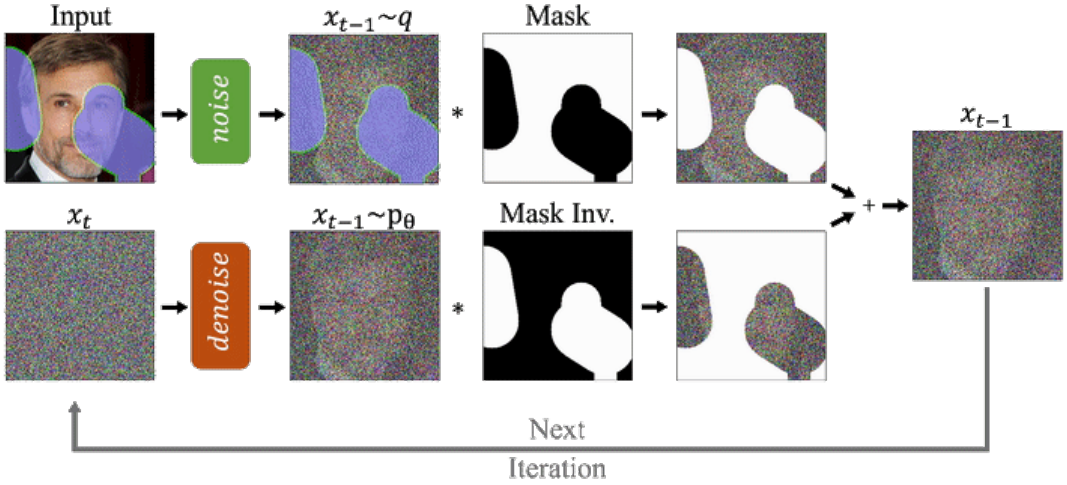


Figure 3.5: Illustration of the process in RePaint [57].

incorrect inpainting by harmonizing the inpainting. It also uses a jump length that seems to decrease blurriness when inpainting. The best-achieved results presented in RePaint are with $T = 250$ timesteps, $r = 10$ resampling steps, and $j = 10$ jump lengths. RePaint outperformed state-of-the-art Autoregressive, and GAN approaches on five out of six mask distributions with the sixth being inconclusive. Figure 3.6 shows examples of inpainting using different input images using RePaint.

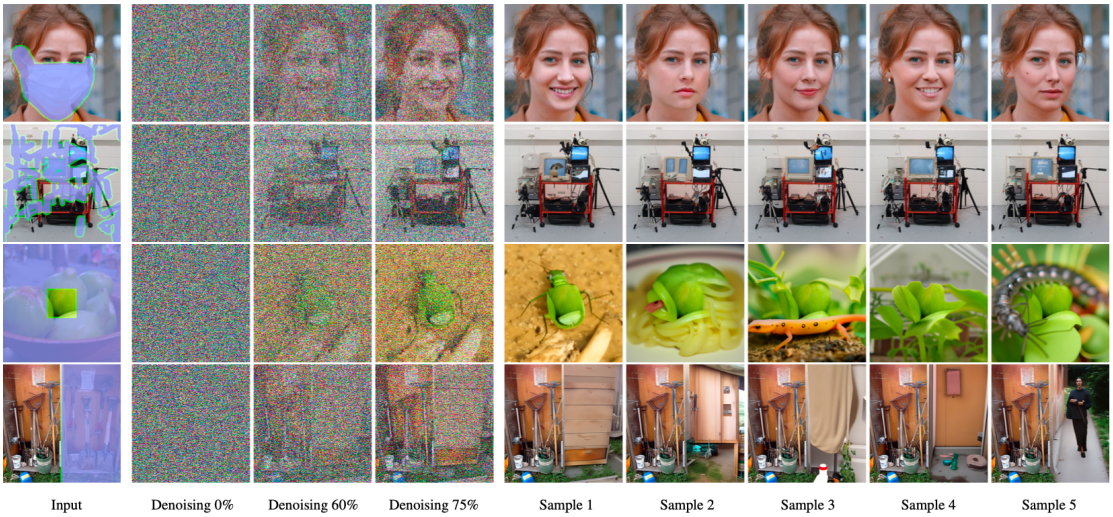


Figure 3.6: The process is conditioned on the masked input seen in input images. Diffusion showed for 0 timesteps 60% of all timesteps and 75% timesteps. 5 possible samples are generated as the process is stochastic [57].

Every reverse diffusion step, as introduced in the Background Section is fundamentally stochastic since it includes new noise from a Gaussian Distribution. The model is also allowed to paint anything that semantically corresponds with the inpainted region because it does not directly guide the inpainted area with any loss. In particular, illustrates images in row 3 in Figure 3.6 the diversity and flexibility of RePaint.

3.5.3 Transfer Learning Models

Transfer learning is a machine learning technique that involves taking a pre-trained model and adapting it to a new task or dataset. The pre-trained model has already learned to recognize relevant features in a large dataset and can be used as a starting point for training a new model on a smaller dataset with similar features. Transfer learning has become increasingly popular in DL due to the availability of large, pre-trained models. These models have been trained on large datasets such as ImageNet and can be used as feature extractors to identify relevant patterns in new datasets.

Diffusion models can also benefit from transfer learning. Transfer learning in diffusion models is applying a previously learned diffusion model to a new dataset or task. The pre-trained diffusion model may be used as a feature extractor, where the higher layers are changed with new layers that have been trained on the new dataset while the bottom layers of the model are frozen. The pre-trained model can with that take advantage of the learned feature representations to find relevant patterns in the new dataset and provide

high-quality samples. However, we do not freeze bottom layers in our experiments but rather fine-tune all layers as DDPMs uses U-Net which has been shown to perform better when fine-tuning all layers [58] rather than freezing lower layers.

3.6 Regularizing Diffusion Models

To prevent overfitting in our diffusion models is regularization introduced. In general, is regularization not something that needs much attention in diffusion models. However we use very little data sometimes in this thesis and regularization of diffusion models is, therefore, something that needs to be used to combat overfitting. Regularization techniques used are but not limited to dropout [59], AdamW optimizer [60], and horizontally flipping images [61].

Dropout is as mentioned in Section 2.3.6.1 a common regulation technique that has the chance to turn off a certain amount of neurons. This helps to spread out strong weights more evenly. Training our DDPMs is dropout amounts of 0, 0.1, or 0.3 used corresponding to a probability 0%, 10 %, or 30 % respectively.

To train the diffusion models is the AdamW optimizer utilized. As was previously indicated in Section 2.3.3.3, AdamW directly introduces weight decay during the optimization process. The weight decay is the part that can be seen as regularization. It is therefore preferred our both Adam and SGD with momentum optimizers.

To artificially increase the dataset size are images horizontally flipped, and the chance for an image to be flipped during training is 50%. Flipping images horizontally is a very common regularization technique, but may introduce some problems when it comes to generative models as will be displayed in the next Chapter.

3.7 Metrics

In this section, metrics for accurately evaluating the quality of generated images and segmentation models presented. The metric used for evaluating generated images measures the distance between real and fake data distribution. Metrics for measuring segmentation models use predicted masks versus ground truth. All metrics used are quantifiable and should give an indication of how well our models perform for our tasks.

3.7.1 Fréchet Inception Distance - FID

The FID was introduced in 2017 by Heusel et al. [62]. FID is a metric used to assess the quality of images in generative models such as GANs. It is commonly seen as an extension of inception score (IS) [63], which only evaluates the distribution of generated images, whereas FID compares the distribution of generated images with the distribution of a

set of real images. This feature space is typically derived from a DNN that has already been trained, such as the Inception-v3 model.

The mean and covariance of the network's activation on a collection of real images and a set of artificially generated images are first computed to get the FID score. The mean and covariance are then used to compute the distance between the two distributions. A lower FID score means that the generated images are of greater quality and are more closely distributed to the real images. To visualize how distortion affects the FID is an example given in Figure 3.7. From this example can we see that swirled images affect the FID the least and salt and pepper the most. The augmented images and how they affect the FID give us an indication of the limitations of the FID score.

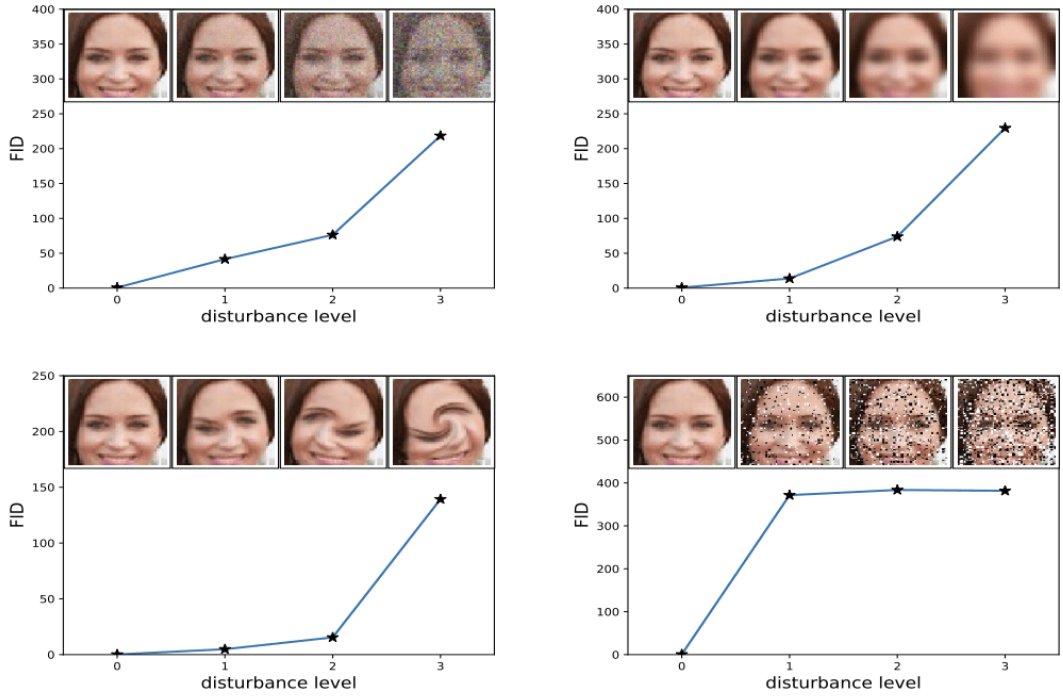


Figure 3.7: Example of how increased distortion correlates with FID score, example taken from [62]. **Upper left** Gaussian noise, **upper right** Gaussian blur, **lower left** swirled images, **lower right** salt and pepper

3.7.2 Intersection over Union - IoU

The IoU measures the overlap between two sets of pixels or bounding boxes and is commonly used to evaluate the accuracy of object detection or segmentation models. The area of overlap between two sets is divided by the area of their union to determine the IoU score shown in Equation 3.1. An IoU score of high means good accuracy, whereas a score of low means inaccuracy. IoU values range from 0 to 1, where a value of 0 indicates no overlap between the two sets, and a value of 1 indicates a perfect overlap. The IoU score can often be thought of as the following: IoU < 0.4 poor, IoU > 0.7 good, and IoU > 0.95 excellent [5]. In many computer vision applications, such as object

[5]https://hasty.ai/docs/mp-wiki/metrics/iou-intersection-over-union

detection, picture segmentation, and scene comprehension, is a widely used metric.

$$IoU = \frac{TP}{TP + FP + FN} \tag{3.1}$$

We evaluated both the IoU and mIoU, which are two distinct metrics for computing intersection over union, in our experiments. Before calculating the IoU score, the mIoU involves averaging the intersection and union values throughout the whole dataset. In contrast, the IoU determines the intersection and union values for every image separately before averaging them.

3.7.3 Dice Similarity Coefficient - DSC

The DSC is a commonly used metric for evaluating the performance of image segmentation models. The DSC scales from 0 to 1, with 1 denoting complete overlap and 0 denoting no overlap, and it assesses the overlap between the anticipated segmentation and the ground truth segmentation shown in Equation 3.2. When there is an imbalance between the amount of foreground and background pixels, which occurs frequently in medical image segmentation, the DSC is especially helpful for evaluating models. The DSC is usually combined with other measures in real-world applications. The DSC is regarded as a standard metric for assessing segmentation models since it has found broad applicability in many fields, such as computer vision, remote sensing, and medical imaging.

$$DSC = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.2}$$

3.7.4 Precision

Precision is a metric used to evaluate the performance of classification models, including image segmentation models. It is defined as the ratio of the true positive (TP) predictions to the total number of positive predictions shown in Equation 3.3. In image segmentation, precision measures how accurate the model is in identifying pixels or regions that belong to the object of interest. A higher precision indicates that the model is better at correctly identifying object pixels and minimizing false positives.

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

3.7.5 Recall

Recall is another metric used to evaluate the performance of classification and segmentation models. It is defined as the ratio of the true positive predictions to the total

number of actual positive cases shown in Equation 3.4. In image segmentation, recall measures how well the model is able to identify all the pixels or regions that belong to the object of interest, including those that may have been missed or incorrectly classified as background. A higher recall indicates that the model is better at identifying all object pixels and minimizing false negatives.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

3.8 Monitoring Diffusion Data Leakage

Image diffusion models are great at generating high-quality synthetic images. However, they are prone to memorization of training data as shown in "Extracting Training Data from Diffusion Models" [64]. In short, the paper highlights that high-performance diffusion models are double as likely to leak data from the training data compared to GANs. This can be a great problem if the data we are training on are subject to privacy concerns. It is also worth noting that leakage increases when FID score decreases in the paper.

To monitor data leakage are generated images compared to training data using the L_2 distance as they do in [64]. The formula for the L_2 distance between two points can be seen in Equation 3.5. In addition to comparing generated RGB images using the L_2 distance, we also turn generated images into grayscale images for comparison against training images. This monitoring can be time-consuming, for example, if we have 1000 synthetic images and 800 training images is a total of $1000 \times 800 \times 2 = 1,600,000$ comparison if we both do RGB and grayscale. Data leakages are however something that should be addressed as it gives an understanding of the generalization capabilities of the generative models.

$$d(p, q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{3.5}$$

3.9 Segmentation with U-Net

In its most basic form, segmentation divides an image's pixels into two distinct subgroups or classes, usually foreground, and background. A common preprocessing step for many computer vision tasks, like object detection and tracking, is this binary segmentation.

To separate pixels into three or more meaningful segments, however, is frequently necessary and calls for advanced segmentation algorithms. For instance, segmentation is used in medical imaging to recognize and isolate particular tissues or organs in an image, such as the heart, liver, and kidneys. Similarly to this, segmentation is used in

autonomous driving to recognize and monitor various items on the road, including cars, pedestrians, and traffic signs.

Image segmentation can be done in a variety of ways, from conventional methods based on low-level picture attributes like color, texture, and borders to more contemporary DL-based methods that automatically learn high-level features from the data. On several benchmark datasets, DL-based segmentation techniques in particular have displayed outstanding performance, especially on large-scale and complex images.

In our experiment is segmentation used to classify pixels as either polyp or background/"clean". We make the distinction between a polyp image where polyps are presented in the polyp and a cropped-out polyp where the actual polyp region is. Therefore is the cropped-out polyp region of key interest when segmenting.

Evaluating the generated synthetic polyp images with segmentation masks uses both the FID and train U-Net with either only real images or a mix of real and synthetic images. The segmentation experiment is tested on three different datasets, thus we train a total of six U-Net models. The U-Net in this thesis used ResNet-18 [65] as an encoder with a total of $14,328,209$ parameters [66]. In Figure 3.8 can we see how U-Net segments a polyp image the downsampling used in the Figure is VGG-16 [67] instead of ResNet-18.

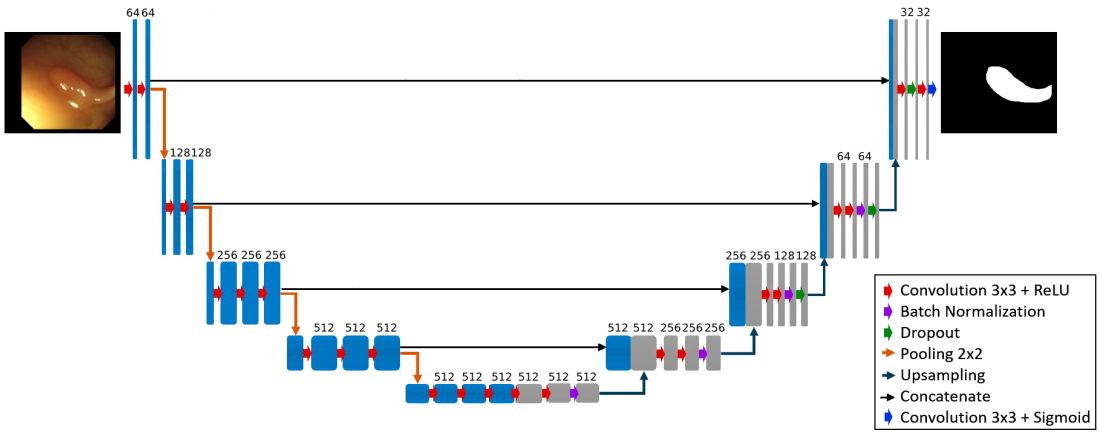


Figure 3.8: Polyp segmentation using the encoder-decoder U-Net architecture with VGG-16 as a pre-trained encoder [68].

The training of the segmentation models used the Adam optimizer. Hyperparameters used during training of the U-Net segmentation models can be seen in Table 3.3. Since our segmentation model only solves binary segmentation is also the sigmoid activation function used in the last layer.

3.10 Summary

In this chapter is two approaches to generating polyp images presented. The approaches rely on pre-training on large amounts of unlabeled data and then fine-tuning. We

Parameter	Value
Batch size	4
Epochs	20
Optimizer	Adam
Learning rate	0.0001

Table 3.3: Hyperparameters and optimizer used for training U-Net segmentation model.

presented the four datasets we will use throughout the thesis where mainly two datasets are used for training and two for validation. Since the thesis revolves around diffusion models are state-of-the-art papers presented. It is explained how the different papers achieve these results from a configuration of models viewpoint. Lastly, is the metric to evaluate the quality of generated images FID presented. The chapter further explains regularization and how we use it in diffusion models, with the main regularization approach being dropout. Other metrics are also introduced to evaluate the effect of segmentation models which will be talked about more in the next chapter. By the end of this chapter, readers will have a solid understanding of the various techniques and metrics used for generating polyp images.

Chapter 4

Polyp Generation

The previous chapter introduced methods used in this and the next chapter. This chapter focuses on simple ways to generate synthetic images using DDPM. Every checkpoint in this chapter generated 1000 samples that took 1 hour and 42 minutes on average. This means that if we have 10 checkpoints the total time spent generating samples is a total of 17 hours. Images in this chapter are all randomly selected to give an indication of generated image quality by being transparent and avoiding "cherry picking". Generated datasets with total iterations, trained for, checkpoints, and noise scheduler is shown in Table 4.1.

Dataset Name	Iterations	Checkpoints	Noise scheduler
I - Unlabeled data	500K	50K	[Linear, Cosine]
II - Masked unlabeled data	500K	50K	Linear
III - Polyp images	[20K, 40K]	2K	Cosine
IV - Cropped Polyp images	30K	2K	Cosine
V - Clean images	20K	2K	Linear

Table 4.1: Overview of generated datasets with total training iterations, model checkpoints, and noise scheduler used.

4.1 Model setup

In order to train the diffusion models described in Chapter 3 much first hyperparameter be selected. The optimizer used in all experiments is the AdamW with a learning rate of 0.0001. The most important hyperparameters at the optimizer used can be found in Table 4.2.

During training is all images center-cropped and resized to size 128×128. The dropout rate varies as some models test multiple different amounts, but always either 0, 0.1, or 0.3. The noise scheduler is also an important hyperparameter either linear or cosine.

Parameter	Value
Diffusion steps (T)	1000
Attention resolution	32, 16, 8
Number channels	128
Batch size	32
Dropout	Varies
Optimizer	AdamW
Learning rate	0.0001

Table 4.2: Hyperparameters and optimizer used for training diffusion model.

4.2 Pre-Training on the GI-tract

The models were trained for 500 000 iterations with a batch size of 32, the total time elapsed during training is approximately 3 days and 11 hours. The models trained in this Section use horizontal flipping as a regularization technique. A total of 95 000 images from HyperKvasir were used for training and 5000 for validation giving a training and validation split of 95/5. All model weights were saved for every 50 000 iteration and generated 1000 synthetic images to evaluate FID score.

4.2.1 Complete Images

To generate complete images was two DDPMs trained, one with a linear noise scheduler and the other with a cosine noise scheduler. The dropout rate was held constant for both models with a value of 0. The difference between a linear and cosine noise scheduler in diffusion models is how fast they add/remove noise. The difference between linear and cosine noise schedulers in latent space can be seen in Figure 4.1.

Figure 4.1: Samples in latent space from linear (top) and cosine (bottom) scheduler with values t from 0 to T=1000. We can observe that the linear scheduler adds noise much faster than the cosine scheduler [69].

The resulting FID score can be found in Table 4.3.

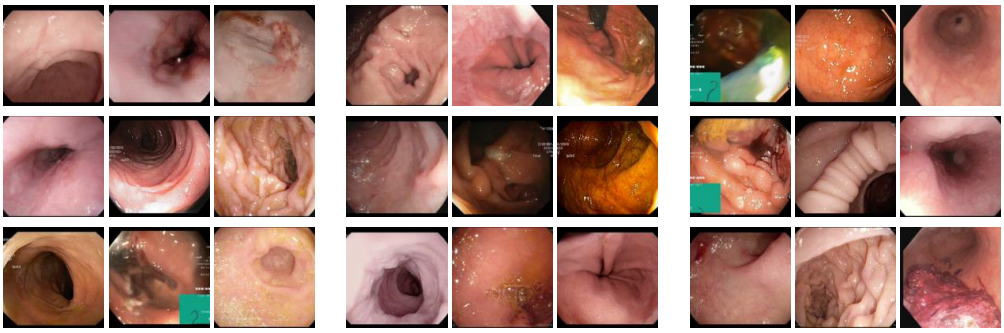
Looking at the FID score comparison between linear and cosine can we see that the linear scheduler seems to achieve better results faster than the cosine. This is the opposite effect observed in [69] on CIFAR-10 [70] where the cosine scheduler achieved better results faster. This effect might be caused by the cosine scheduler having less regularization implying that the unlabeled data in HyperKvasir is less diverse than CIFAR-10. Both schedulers achieve similar results after 300K iterations based on FID

	FID	
Iterations	Linear	Cosine
50K	74.76	92.77
100K	28.85	34.53
150K	28.04	30.51
200K	27.17	28.05
250K	***26.18***	27.45
300K	26.81	26.88
350K	26.46	25.99
400K	26.27	***25.86***
450K	26.17	25.91
500K	**25.83**	**25.66**

Table 4.3: Comparison of generated unlabeled images trained on HyperKvasir 128×128 model with linear and cosine noise scheduler. The best early stopping FID score is highlighted in ***italic-bold*** and the overall best FID score is highlighted in **bold**.

scores.

In Figure 4.2 can we see a comparison between generated images between the two different schedulers trained for 500K iterations and unlabeled images from HyperKvasir.



(a) Linear (b) Cosine (c) HyperKvasir

Figure 4.2: Comparison between linear scheduler, cosine scheduler, and images from HyperKvasir (rescaled and center-cropped).

All images were as previously mentioned randomly selected to give an indication of generated image quality. The comparison shows that there is no clear distinction between generated samples using a linear and cosine scheduler. It is hard to deduce whether or not overfitting has occurred during unlabeled generation. The reason for that is that we would then need to compare synthetically generated images to images from the training data. This process is computationally heavy as we train on 95 000 images. Such comparison was not performed on all synthetic unlabeled images but for a majority and proved no strong signs of overfitting.

4.2.2 Masked Images

Masked images from the unlabeled dataset were shown in Section 3.3.2. The masks used were generated by a FastGAN earlier explained to train on unlabeled images that have shapes similar to that of polyps. The masked images are by nature a more sparse representation than a complete image as we have fewer non-zero pixel values.

To generate masked images was a model trained using a cosine noise scheduler and 0.1 dropout. The resulting FID score can be seen in Table 4.4. The best-performing model is trained for 300K iterations and results in a FID score of 18.80 on unlabeled images in HyperKvasir.

	FID
Iterations	Cosine with 0.1 dropout
50K	57.74
100K	20.66
150K	19.43
200K	19.29
250K	19.03
300K	**18.80**
350K	18.88
400K	19.06
450K	18.99
500K	19.14

Table 4.4: FID score for generated masked images on the unlabeled dataset. The best FID score is highlighted in **bold**.

Generated masked images from the best model can be seen in Figure 4.3 and should look similar to masked unlabeled images shown in Figure 3.3.

The generation of masked images appears to be good as we can see parts of green boxes generated in the bottom left corner of the top middle image. In the bottom left image can we see text generated on the right-hand side which can be attributed to that text is usually on the left-hand side of images, but we train using horizontal flipping. Color seems to be consistent and images look like they can come from the GI-tract judging by the pixels that are non-zero.

4.3 Fine-Tuning with a Motive

The models were either tuned on polyps from the Kvasir-SEG dataset or "clean" colon images from the labeled part of HyperKvasir. The models were tuned for between 20K to 40K iterations with a batch size of 32, the total time elapsed during tuning is approximately between 3 hours and 30 minutes to 7 hours and 13 minutes depending on the number of iterations. The training and validation split was 80/20 for all tuning models. Regularization techniques are similar to those used in Section 4.2 which uses

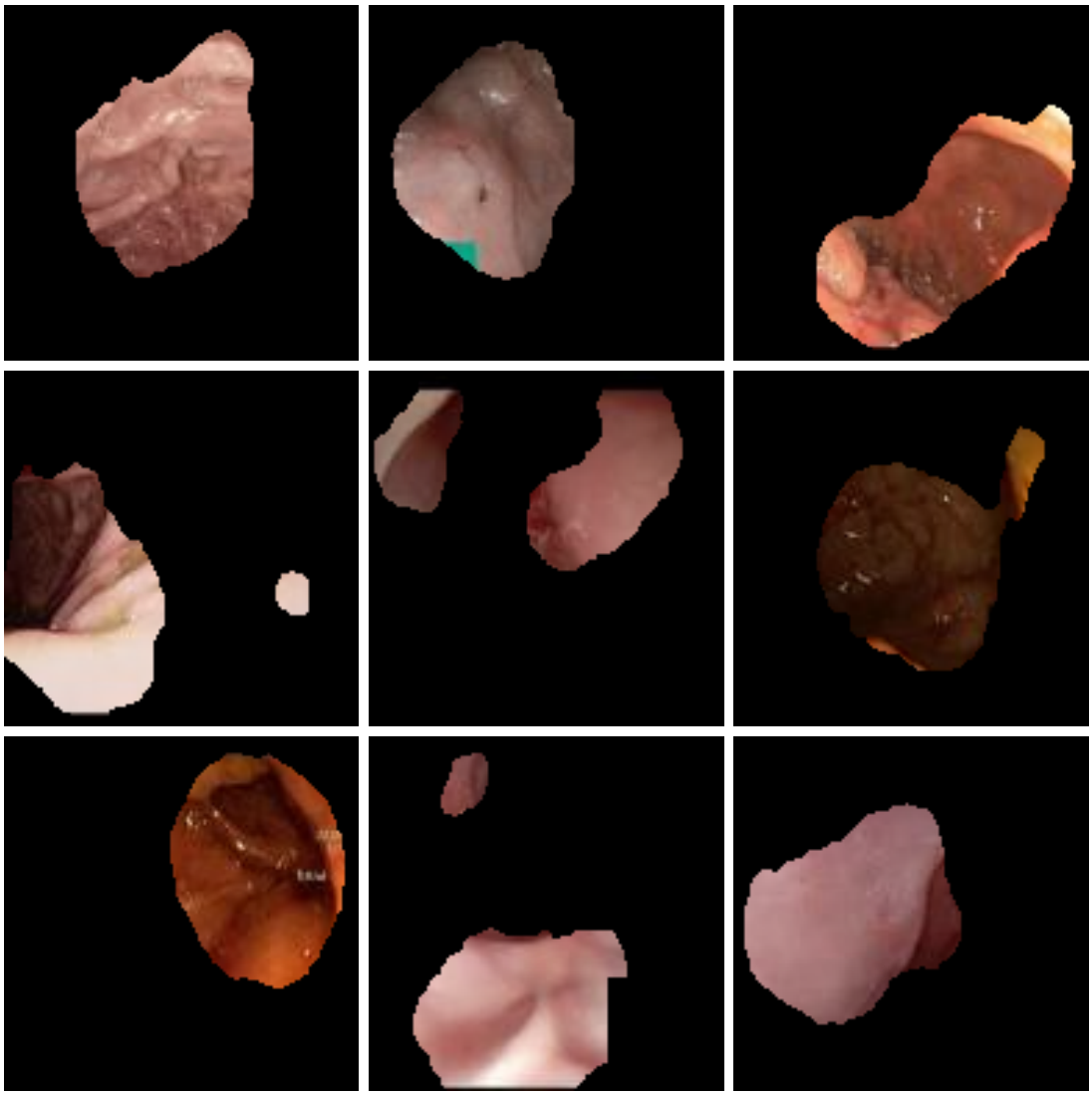


Figure 4.3: Generated masked images samples from 300K iterations with cosine noise scheduler and 0.1 dropout model .

horizontal flipping. All model weights were saved for every 2000 iterations and generated 1000 synthetic images to evaluate FID score. "clean" images used in this thesis are ulcerative colitis, ileum, and cecum. This is important to keep in mind as ulcerative colitis is classified as a bowel disease as described in 2.1.1.1.2.

4.3.1 Polyp Images

To generate synthetic polyp images was the best-performing pre-trained model (500K iterations cosine noise) fine-tuned. The images used for fine-tuning are polyp images from the Kvasir-SEG data, only using the original images and not ground truth. Two models were fine-tuned with 0 and 0.3 in dropout respectively. The resulting FID score can be seen in Table 4.5 with the model having 0.3 dropout tuned for 26K iterations achieving the best result based on FID score.

	FID	
Iterations	Dropout 0	Dropout 0.3
2K	149.20	150.05
4K	136.79	149.62
6K	127.53	133.73
8K	**116.94**	110.56
10K	118.25	100.22
12K	118.47	93.53
14K	122.05	91.49
16K	126.69	87.00
18K	129.66	86.10
20K	135.88	84.12
22K		82.90
24K		82.83
26K		**80.55**
28K		81.72
30K		83.13
32K		84.30
34K		82.81
36K		84.85
38K		86.66
40K		86.04

Table 4.5: FID score for generated images tuned towards polyp generation with different amounts of dropout. The best FID scores are highlighted in **bold**.

Comparing the two models are similar effects observed for iterations after the best FID score. The generated images after the best models would then lack fine features, such as blood vessels which are essential when it comes to polyps. These overfitting artifacts are similar to that from [71] which is reflected by increasing FID.

Samples from the best model can be seen in Figure 4.4. From this can we see that for the most part are highly realistic polyps generated, however sometimes it is unclear if polyps are generated or not. The generated polyps should therefore be checked by humans beforehand if they are going to be used further for example training models or educating medical professionals. For the generation to be considered successful should the image contain one or more polyps.

The generation of polyps is not always accurate and some errors can occur. This can be seen in image 2 from the top left where it does not seem that the model has generated a polyp. Image 3 is also questionable as it looks like an image from the esophagus which is considered a part of the upper GI-tract while polyps in Kvasir-SEG are from the lower GI-tract.

Figure 4.4: Generated polyps image samples from 26K iterations fine-tuned model with 0.3 dropout.

4.3.2 Cropped-out Polyps

To generate synthetic cropped-out polyp images was the best-performing masked pre-trained model (300K iterations) fine-tuned. The images used for fine-tuning are cropped-out polyp images from the Kvasir-SEG dataset. The cropped-out polyps are obtained by removing everything in the images except the ground truth region and therefore rely on both original and ground truth images. To generate cropped polyps were three models trained with different amounts of dropout 0, 0.1, and 0.3. The scores for the models can be seen in Table 4.6 with the model tuned for 18K iteration achieving the best result.

	FID		
Iterations	Dropout 0	Dropout 0.1	Dropout 0.3
2K	138.24	135.42	135.22
4K	117.14	115.59	115.23
6K	96.78	98.30	95.75
8K	93.10	91.47	82.27
10K	92.72	87.27	77.22
12K	**92.70**	86.24	74.00
14K	95.91	**84.64**	71.25
16K	97.95	86.56	71.12
18K	100.66	85.25	**69.51**
20K	100.35	88.02	72.16
22K	100.77	87.39	71.21
24K	99.75	86.41	73.54
26K	99.63	86.34	72.53
28K	97.21	85.72	72.21
30K	99.24	85.44	71.56

Table 4.6: FID score for generated images tuned towards cropped-out polyp generation with different amounts of dropout. The best FID scores are highlighted in **bold**.

Samples from the best model can be seen in Figure 4.5. From this can we see that it is not clear whether or not the cropped-out polyps are realistic. Some features might not be present as the generated cropped-out polyps are highly dependent on polyp features in Kvasir-SEG. The generation of cropped-out polyp images proved to be a challenging task. The reason for the problem is that cropped-out polyps have even fewer pixels available to learn from than experiments in previous sections. The data deficiency stems from that the actual polyps commonly only take up 5-70% in a polyp image.

Simultaneously were other drawbacks detected where the generator did not generate anything by setting all pixel values to zero corresponding to black pixels. This effect peaked around 6-8K iteration and can be seen in Figure 4.6 for the three different models. The reason for this effect is unclear but indicates that the models think it is better to not generate anything than to try to generate something between unlabeled data and polyps as it is in a transition between the pre-trained model going to fine-tuned. These completely empty images are not useful and were therefore discarded for further use.

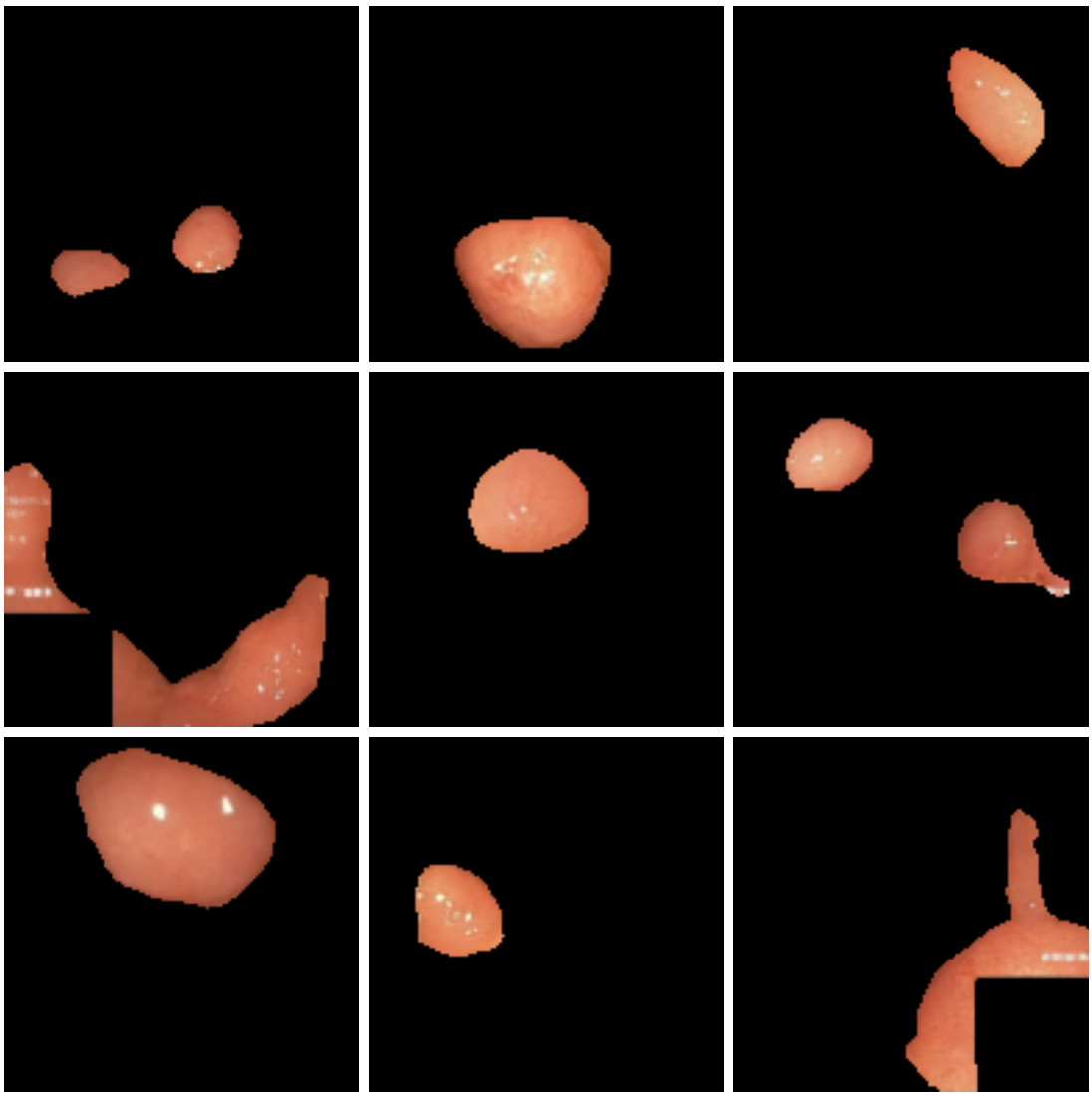


Figure 4.5: Generated cropped-out polyps from our best model 18K iterations 0.3 dropout.

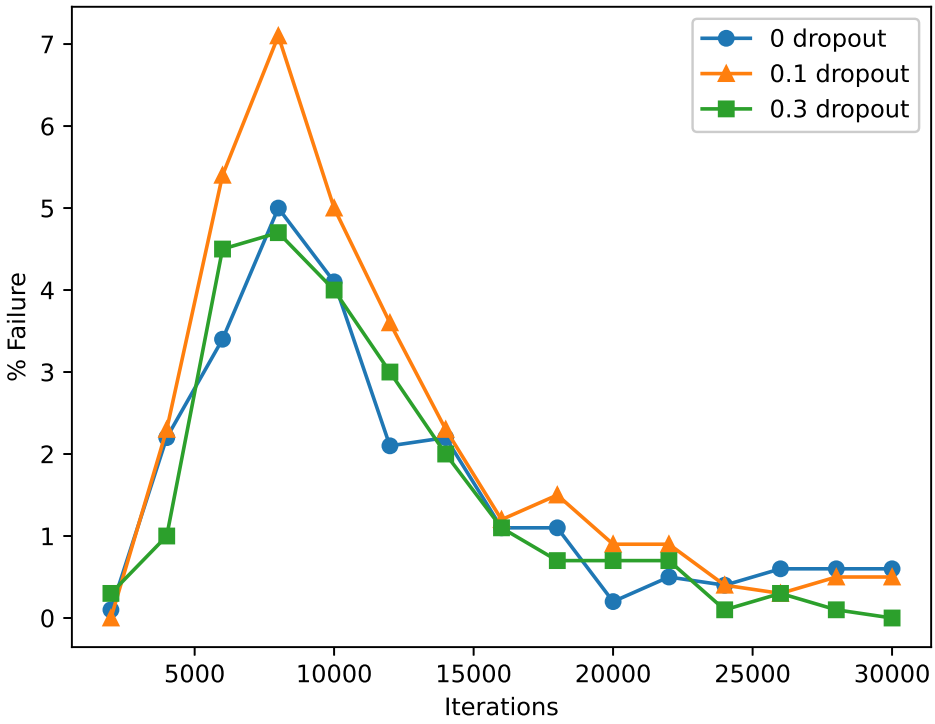


Figure 4.6: Amount of black images created total with 0, 0.1, or 0.3 dropout models based on iterations tuned. The total number of images generated per iteration for each model is 1000.

4.3.3 Clean Colon

To generate synthetic "clean" images was the best-performing pre-trained model (500K iterations cosine noise) fine-tuned.

The images used for fine-tuning are "clean" images from the labeled part of the HyperKvasir dataset with classes ulcerative colitis, ileum, and cecum. The amount of ulcerative colitis images in the labeled part in HyperKvasir is 851. The amount of ileum images in the labeled part in HyperKvasir is 9. The amount of cecum colitis images in the labeled part in HyperKvasir is 1009. Resulting in a total of $1,869$ "clean" images. The split used for "clean" images was 80% (1495 images) for training and 20% for validation (373 images).

Recalling from the Background is ulcerative colitis actually an inflammatory disease in the bowel region that comes in various degrees. Nevertheless was all degrees of ulcerative colitis images used as "clean" images as they do not contain any polyps.

Only one model was fine-tuned with 0.3 in dropout on these "clean" images. The resulting FID score can be seen in Table 4.7 with the model tuned for 20K iterations achieving the best result based on FID score. It is worth noting that the model could be trained longer as the FID was still decreasing. The model was however not trained further as the decrease in FID started to slow down. This effect was followed by a rise in FID (overfitting effects) in earlier testing. Further training would therefore have a minimal effect on the FID from our previous experience.

	FID
Iterations	Dropout 0.3
2K	94.63
4K	86.01
6K	70.90
8K	61.33
10K	56.84
12K	53.37
14K	52.01
16K	52.84
18K	51.88
20K	**51.77**

Table 4.7: FID score for generated images tuned towards clean generation. The best FID score is highlighted in **bold**.

Samples from the best model can be seen in Figure 4.7. From this can we see that the generated images are highly realistic. The images are able to generate fine features such as blood veins and different green boxes either in the bottom left or right corner as we used flipping during training.

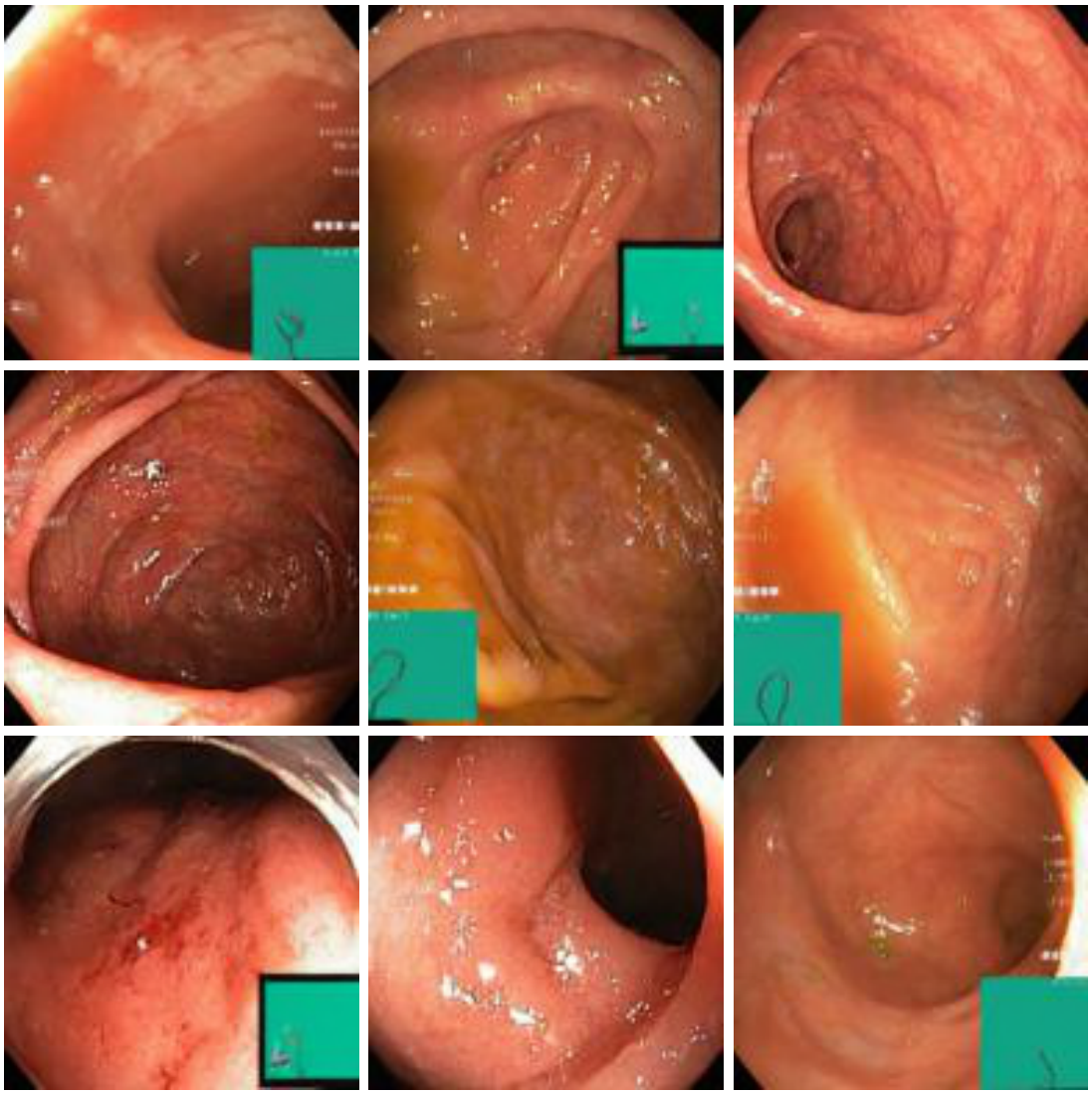


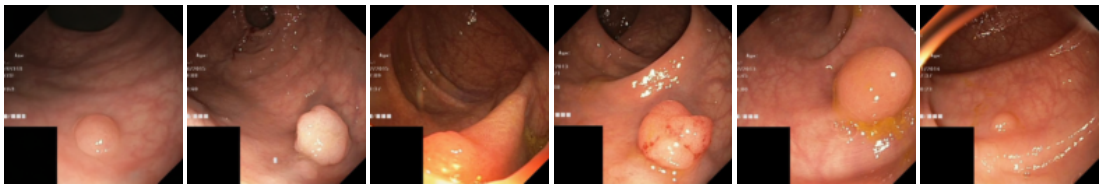
Figure 4.7: Generated clean colon images from our best model 20K iterations 0.3 dropout.

4.4 Image Correlation

The motivation for this section revolves around both privacy risks and generalization. Medical images can be sensitive private data, therefore, is an understanding of the risks of generative models needed when training on such data. Similarly is an understanding of how and why diffusion models memorize training data needed to understand their generalization capabilities. In particular, if it goes unnoticed, remembering specific examples can be problematic. We, therefore, show a generated polyp image with its five closest images from the training dataset.

Evaluating the closeness of images to a source image can use one use a variety of different metrics. In our experiments, both the L_2 distance and correlation coefficient from OpenCV were used. The L_2 distance did not prove to be effective with our generated samples in detecting similar images. We therefore instead used the TM_CCOEFF from OpenCV which stands for Template Matching Correlation

Coefficient. The range of TM_CCOEFF is normalized and ranges from 0 to 1 where 0 is no matching and 1 is a total match. In Figure 4.8 can see the source image which is a generated polyp image from our best-performing model in Section 4.3.1.

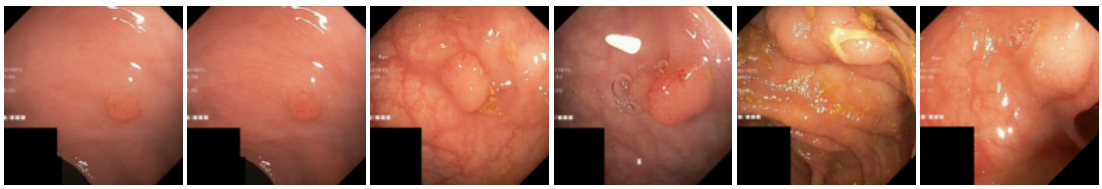


Source	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
	0.909	0.883	0.882	0.877	0.876

Figure 4.8: Synthetic generated polyp as the source image. 5 samples from the training dataset closest related to the source image with their TM_CCOEFF scores.

In addition, 5 images from the training dataset that is closest related to the source image based on their TM_CCOEFF. The source image was hand-picked as it was more correlated to images from the training dataset than other generated samples. This was performed to show how a generated image can look similar to those in the training data. In our example can we especially find common features between our source image and sample 1 for example with respect to the color of the GI-tract, text on the left-hand side, the continuation of the tract in the upper middle part of the image, and the black boxes in the bottom left corner.

To underline how a memorized image looks like, we generate images using a model fine-tuned for 50K iterations. In Figure 4.9 can we see how a source image from overfitted model an overfitted model is almost the same as sample 1. There are some minor differences between the source image and sample 1, but we can consider that the diffusion model has memorized sample 1 from the training data.



Source	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5
	0.991	0.910	0.896	0.888	0.872

Figure 4.9: Synthetic generated polyp from an overfitted model as the source image. 5 samples from the training dataset closest related to the source image with their TM_CCOEFF scores.

4.5 Interpolation

From earlier sections can we recall how we can either add noise to an image $x_0, x_0 \sim q(x_0)$ or denoise it by $\bar{x}_0 \sim p(x_0|\bar{x}_t)$. We can use this to first add noise to two source (src) images then interpolate between them and finally denoising the images as presented in [45]. We do this in intervals of 125 from $t = 0$ to $t = 1000$ with t denoting timesteps/steps. Interpolation in $t = 0$ is the same as interpolation in the pixel domain.

We do interpolation using our best polyp DDPM from Section 4.3.1.

Figure 4.10 shows interpolation between two different polyp images. From the interpolation, we see how an equal combination of the two polyps looks in the $\lambda = 0.5$ column. Particularly interpolation between 250 and 750 timesteps shows interesting results. The polyp in source image 1 can be seen on the right side with heavy red indicating blood vessels suggesting it might be adenomatous. The continuation of the tract is also visible in the middle of the image. The polyp in source image 2 is large being approximately in the middle of the image and being more yellow suggesting it might be non-adenomatous. Going from source images 1 to 2 an enclosing of the tract, as well as a reduction in blood vessels. When the interpolated has an equal contribution ($\lambda = 0.5$) from the source images is generally the polyp on the right part of the image. This might be since source image 1 has the polyp on the right side while source 2 does not have the polyp in a particular position.

In the interpolation between a polyp image and a clean colon image in Figure 4.11 can we see how the model "sees" a decrease in polyp features. Polyp features are lost more with higher t and $\lambda > 0.7$. Source image 1 has a polyp in the bottom left of the image with the continuation of the tract being in the middle of the image. It is hard to determine the key features of the polyp as it seems to be heavily exposed to some light. Source image 2 shows a clean colon where the continuation of the tract is a little more to the bottom right. The color of the tract in source image 1 also appears to be more brown while more pink in source image 2. The interpolation between the two images uses the model only trained on polyps and therefore favors polyp features. To get a more balanced feature visualization can one train a model on both polyp and clean images. The interpolation could then show polyps for $\lambda < 0.5$ and clean colon for $\lambda > 0.5$ in the case when $\lambda = 0.5$ is possible very subtle polyps assuming that the model is trained on an equal amount of polyp and clean colon images. The chance of the interpolated image containing a polyp is however more like the smaller λ is.

Interpolation test features go from finer to coarser with increasing timesteps. Going from source 1 to source 2 is a gradual change in both the polyp and background. In both examples can we see that we lose all information from our source images with $t = 1000$. On the hand when $t = 0$ it is interpolation in pixel space and therefore also not relevant to understand learned features. Additional interpolation is shown in Appendix B.

4.6 Questionnaire

Generated polyps have previously in Section 4.3.1 been evaluated based on their FID which is a form of quantitative measure. This section will focus on a qualitative approach to judging the generated polyps. This involved asking domain experts by giving their subjective feedback on generated images. To be more precise are the domain experts computer science researchers that focus on using ML in the medical domain. The questionnaire in its entirety is shown in Appendix C.

The participants were given 10 polyps images where 5 of our real polyp images and 5 fake polyp images. The real polyps are real polyps randomly selected from the Kvasir-

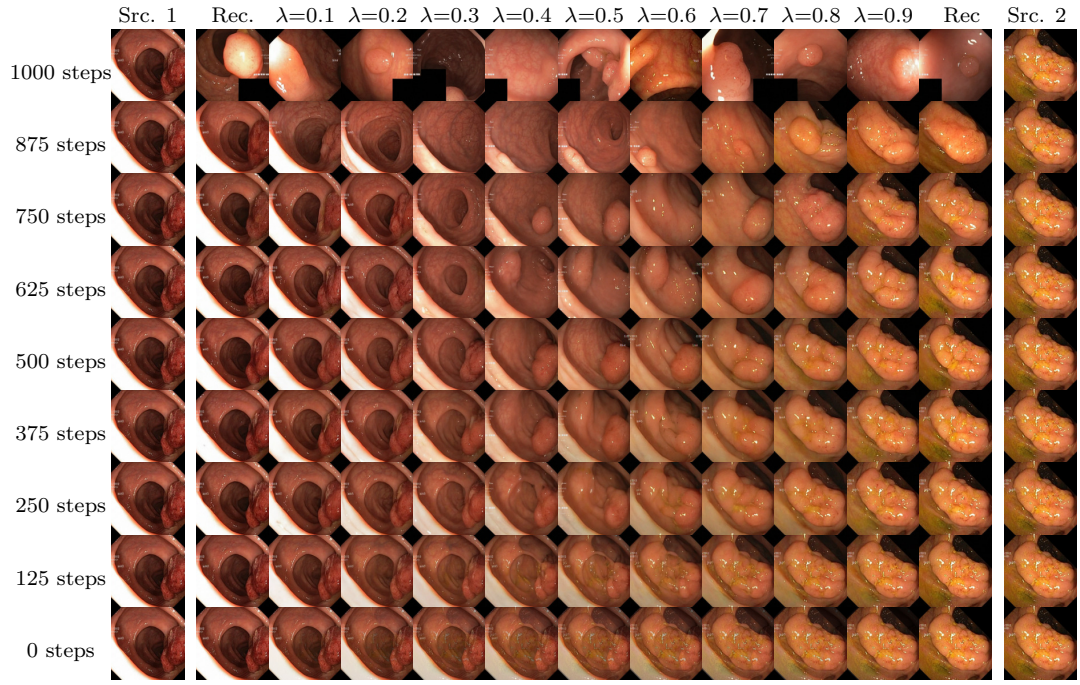


Figure 4.10: Interpolation in latent space between two different polyp images.

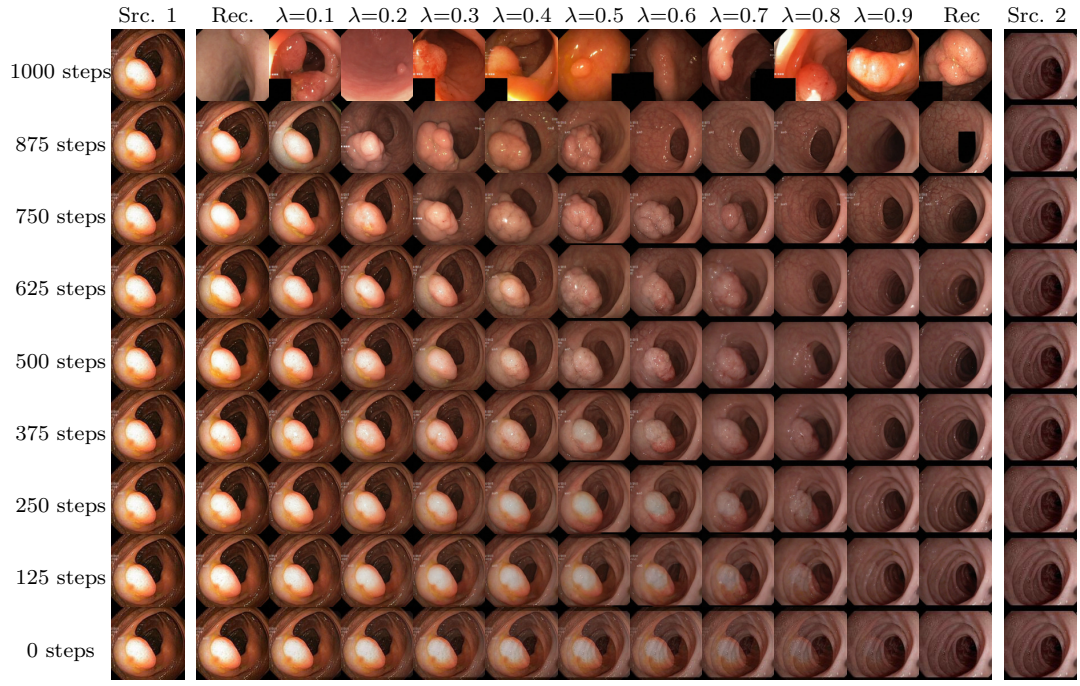


Figure 4.11: Interpolation in latent space between a polyp image and a clean colon image.

SEG dataset. The fake polyps are the synthetic-generated polyps from the best model in Section 4.3.1. All images both real and fake were randomly selected. The participants were to fill in a scale from 1-10 on how confident they were that the image is generated, polyp context, if background appears generated if polyp appears generated, polyp fitting background, and confidences regarding predicted histology. They were also asked to specify the kind of polyp the image showed. The only other information given to the participants was that a student was working on a thesis on synthetic polyp generation. Table 4.8 shows the results of whether or not the participants thought the image was generated.

Experience	TP	FP	FN	TN	Accuracy	Recall	Precision
5	3	5	2	0	30%	60%	37.5%
3	1	2	4	3	40%	20%	33.3%

Table 4.8: Results from the questionnaire on whether or not the participants think the image is real or generated.

Experience describes the years that participants have been in their position. True positive (TP) corresponds to the correct identification of a real polyp. False positive (FP) corresponds to a fake polyp identified as a real polyp. False negative (FN) corresponds to a real polyp identified as a fake polyp. True negative (TN) corresponds to the correct identification of a fake polyp. Participants giving a value of 5 or less were categorized as thought to be real and 6 or higher was categorized as fake.

In the questionnaire, it is evident that some of the generated polyps deceived the participants. However, the subjective assessment of the polyps' realism can differ greatly from person to person and the number of participants was low. Given a sufficient participant pool and the fake polyps being completely indistinguishable from actual ones, the accuracy should, on average, be around 50%. Our two participants got a combined average accuracy of 35% and if this trend were to exist with a larger number of participants could it suggest that synthetic-generated polyps are more general than real ones. It was mentioned that the images originally were of size 128×128, but enlarged to 256×256. This can have caused some distortion in the images that can have affected the participants' judgment of whether or not polyps are real.

4.7 Summary

In this chapter are various types of images generated and quality evaluated based on FID. The images generated include unlabeled images, polyps images, and clean colon images. We test DDPMs either using linear or cosine noise scheduler. The difference between linear and cosine noise schedulers was shown to be insignificant based on the FID. We also tested the generation of images with various amounts of dropout. It was shown that models with dropout overall performed better than those without, but simultaneously needed to be trained for more iterations. Interpolation was performed between real images to try to visualize features learned when generating polyps images. We do a test between polyp-to-polyp images and polyp-to-clean images. The polyp-to-polyp interpolation showed coarse to fine features and highlighted how a combination

of two polyps might look like. The best synthetic-generated polyps were presented to domain experts to assess realism. Cropped-out polyps are generated and assessed based on FID which, combined with clean colon generation, is central for the next chapter.

Chapter 5

Polyp Segmentation with Synthetic Data

The previous chapter focused on the generation of synthetic images but did not highlight what these synthetic images can be used for. This chapter tries to address the practicality of synthetic polyp images by using them to train a simple segmentation model. To generate polyps with corresponding segmentation masks was RePolyp introduced. RePolyp uses synthetic cropped polyps and inpaints a probabilistic background. For this to work well should both the cropped polyps and the inpainted background look real while simultaneously being semantically reasonable. The inpainting should not generate polyps in the background, but only a "clean" colon. RePolyp uses RePaint which makes use of pre-trained DDPMs. The RePolyp framework is shown in Figure 5.1.

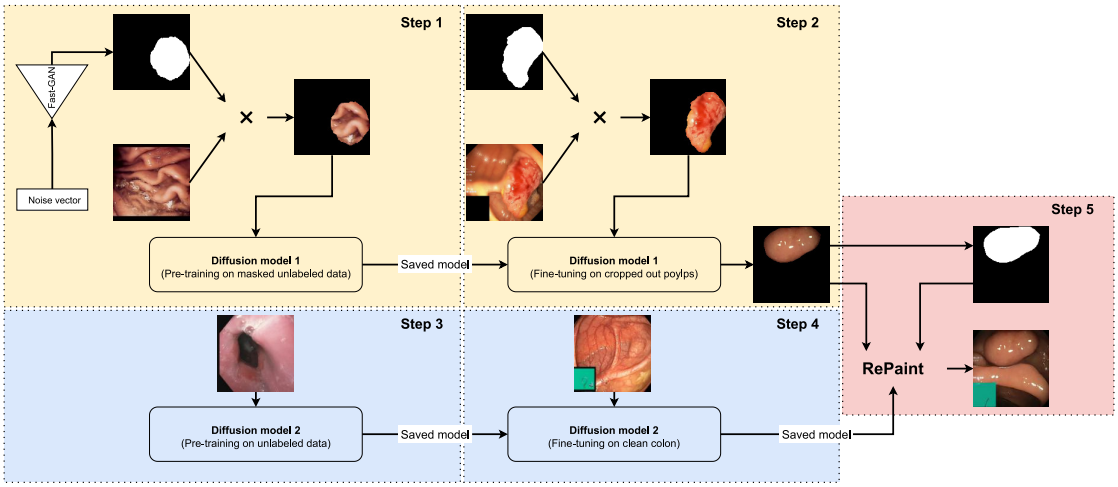


Figure 5.1: Framework to generate polyps with segmentation mask. **Step 1** Pre-training on masked images. **Step 2** Fine-tuning on cropped-out polyps. **Step 3** Pre-training a second diffusion model. **Step 4** Fine-tuning second model on clean colon. **Step 5** Inpainting using diffusion model 2 and cropped-out images.

To evaluate how well the generated images from RePolyp were a segmentation task set up in addition to calculating the FID scores. The segmentation task compared two U-Net models, one trained only on real images and the other on a mix of real and

synthetic images.

5.1 Results and Evaluation

Images generated from the RePolyp framework are both tested based on their FID in addition to being used to train a simple U-Net segmentation model. Generated polyps from the RePolyp framework are inpainted using 3 different DDPMs. Each model inpaints 1000 with a background for cropped-out polyps from images Section 4.3.2. To evaluate a U-Net for segmenting polyps was one model trained only on 800 real images and a second model on 800 real and 800 fake images. The validation for the segmentation models was performed on three different datasets.

5.1.1 Polyp Generation with Masks

Table 5.1 shows the quality of the polyps produced by RePolyp, and it is easy to see that choosing a diffusion model with polyp-specific fine-tuning based on the FID score is appealing. This, however, presents a concern because we cannot guarantee that the model won't produce more polyps. The clean background is consequently preferred since it prevents the occurrence of background polyps and has a FID score that is lower than the pre-trained model.

Model	FID
Pretrained unlabeled	138.38
Fine-tuned clean	128.83
Fine-tuned polyps	**93.43**

Table 5.1: FID score for the generated polyps from three different models using RePolyp.

The generated images in Figure 5.2 are somewhat realistic, but less so than the ones in Section 4.3.1. This indicates that there might be a better approach to generating polyps with masks. The main problem seems to come from sub-optimal generated cropped-out polyps. The generated background on the other hand seems to be realistic and somewhat concise with the generated cropped polyps.

The generated polyps often generate green boxes in the left or right corners of the images. This is caused since clean images use green boxes and polyp images in Kvasir-SEG mostly use black boxes. This can be changed by adding a post-processing step by turning green boxes black. It is also worth mentioning that generated cropped-out polyps can be in areas where our model wants to generate these green boxes. This can be observed in the second generated image in Figure 5.2 which causes some semantic inconsistencies. This can be adjusted in post-processing by completing the boxes the model tries to generate, but we would then similarly need to remove areas where the box should be in our segmentation masks.

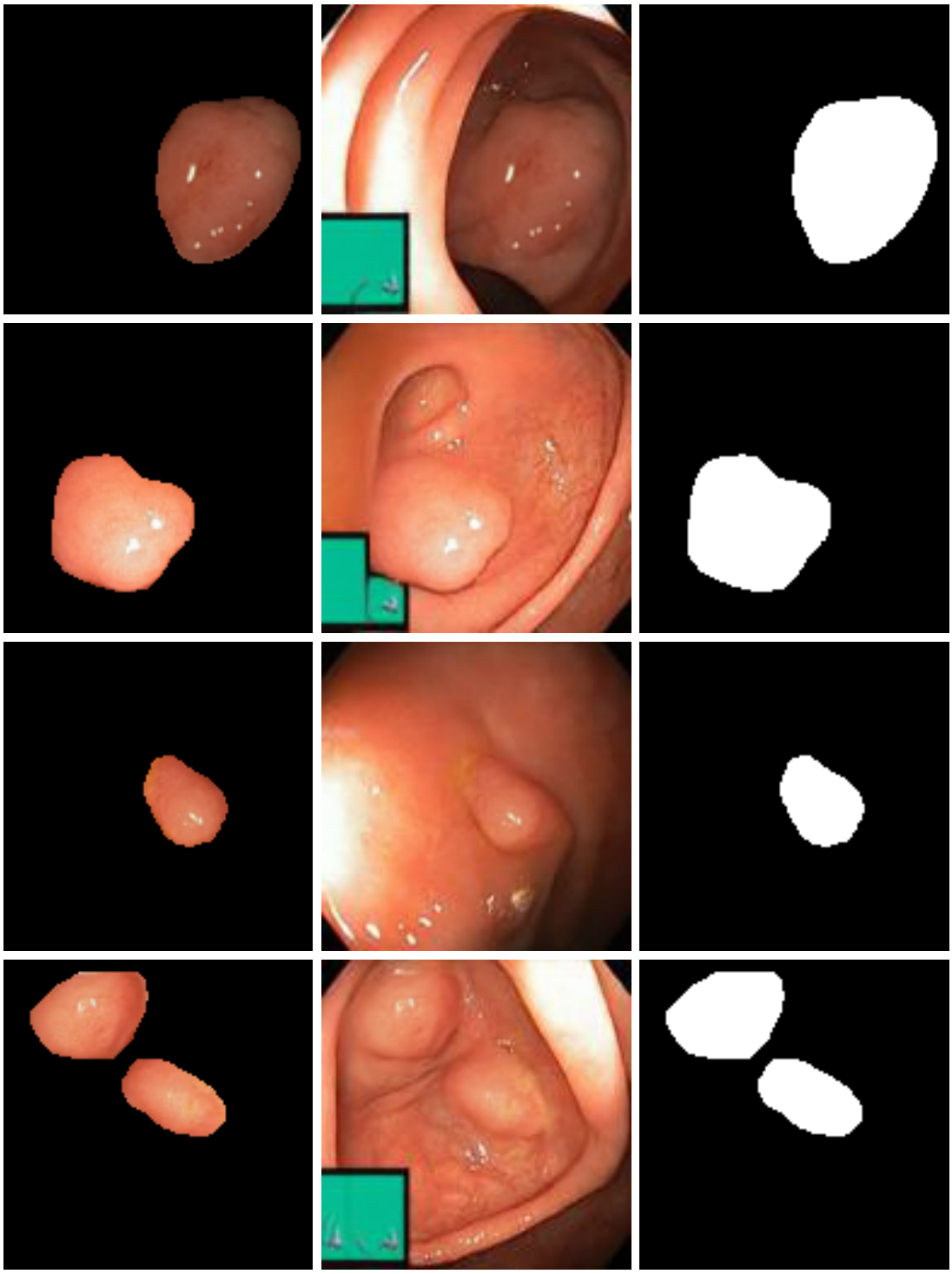


Figure 5.2: From left to right; Cropped-out generated polyp, Cropped-out polyp with clean inpainted background, segmentation mask derived from the cropped-polyp. Segmentation models use images from column two and three.

5.1.2 Polyp Segmentation

To evaluate the segmentation models were 20% of the Kvasir-SEG dataset used for validation. In addition, were the ETIS-LaribDB [49] and CVC-ClinicDB [48] datasets used for cross-dataset validation. A different way to divide the data would be to use Kvasir-SEG for training, ETIS-LaribDB for validation, and CVC-ClinicDB for testing resulting in only one model.

In Tables 5.2, 5.3, and 5.4, we observe the differences between our baseline dataset and our dataset with added synthetic images.

Table 5.2: Validation 200 Kvasir-SEG images

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.762	0.732	0.840	0.871	0.821
+800	**0.785**	**0.766**	**0.857**	**0.913**	**0.826**
Chg %	3.02%	4.64%	2.02%	4.82%	0.61%

Table 5.3: Validation ETIS Larib Polyp DB

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.351	0.470	0.408	0.583	0.709
+800	**0.396**	**0.492**	**0.451**	**0.604**	**0.727**
Chg %	12.82%	4.68%	10.54%	3.60%	2.54%

Table 5.4: Validation CVC-ClinicDB

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.642	0.628	0.735	0.831	0.720
+800	**0.654**	**0.660**	**0.738**	**0.869**	**0.733**
Chg %	1.87%	5.10%	0.41%	4.57%	1.81%

We see different improvements for various metrics and datasets. Good overall metrics such as the mIoU see an increase in mIoU of 4.64%, 4.68%, and 5.10% on the respective validation datasets. Simultaneously is an increase in DSC of 2.02%, 10.54%, and 0.41% observed with adding synthetic data. Precision also increases with adding synthetic images to our dataset, meaning that the model is more confident in predicting pixels of polyp pixels. However, precision should be seen together with recall. The effect of our added data on recall, on the other hand, is negligible overall. The increase in performance on the validation data is possible due to artificially increasing diversity in the training data. While adding synthetic data seems to increase generalization capabilities, it can possibly also increase optimization, but this was not looked into. It is also worth noting that synthetic data use horizontal flipping as mentioned earlier and may therefore offer more regularization than the original data.

Figure 5.3 shows the results of two segmentation models on 6 images from Kvasir-SEG images from the validation dataset.

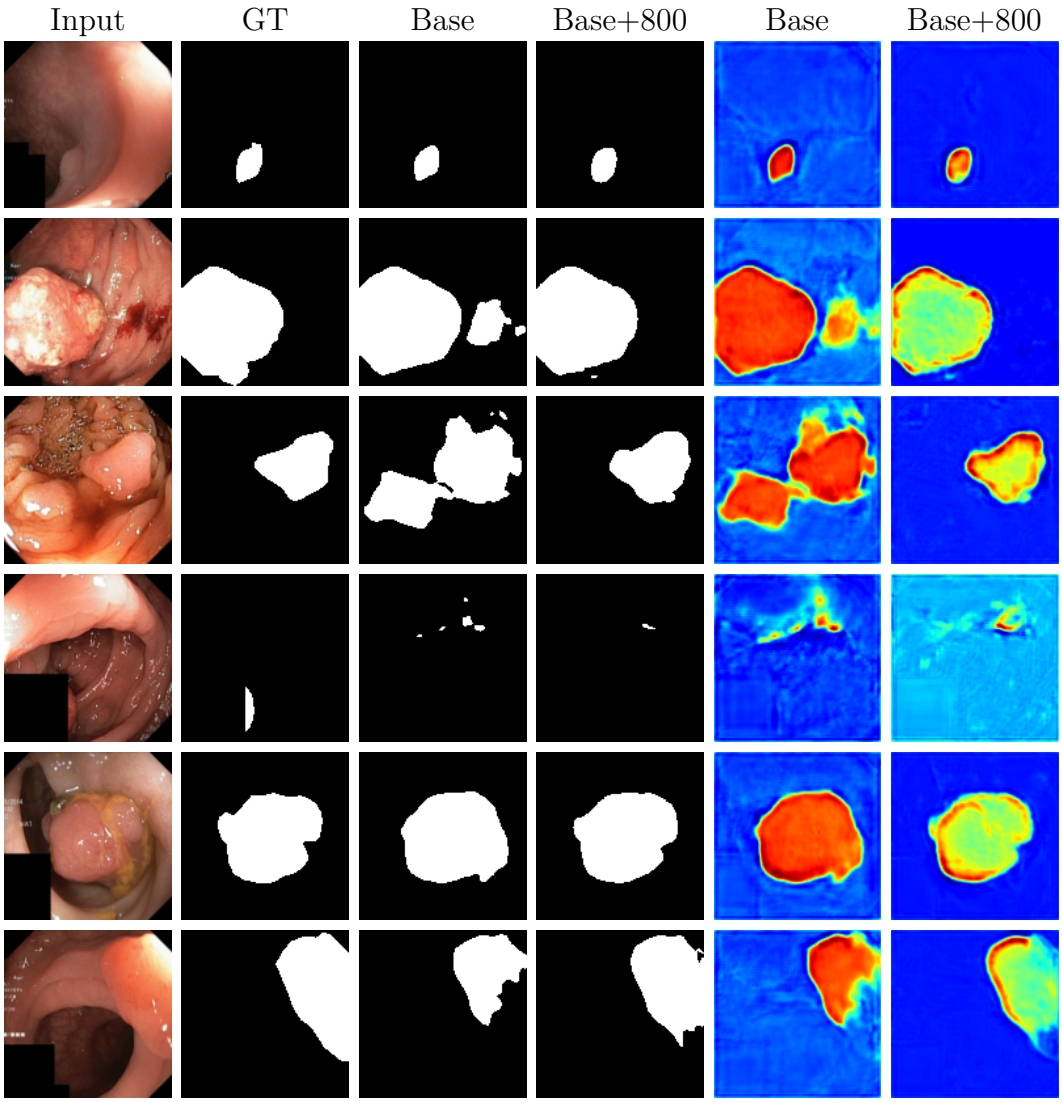


Figure 5.3: Visual segmentation performance on Kvasir-SEG images using a U-Net architecture. Black and white images are segmentation masks, and the last two columns represent heatmaps. From left to right; Input image, GT - Ground Truth, real images, real images + 800 synthetic images, real images, real images + 800 synthetic images.

The first 4 images are handpicked to highlight differences and/or hard-to-detect polyps and the last 2 images are randomly selected. The first image appears to be easy to segment. In the second and third images, we see that our baseline model classifies more of the clean as polyps which will result in a lower precision score. The baseline model seemingly segments based on the texture of polyps, giving a somewhat uniform value to pixels segmented as polyps. The dataset with added synthetic data, on the other hand, seems to have a better understanding of polyp edges with darker red pixels being more frequent near polyp edges and more uniform values for clean parts. The increase in performance might therefore be because the model with synthetic data has a better idea of what clean parts of a colon look like. The fourth image is a challenge with it being very hard to detect the polyp. Both models fall short of detecting the polyp, but the +800 model classifies fewer pixels as polyps. In both images five and six, we see that the +800 model outperforms the base model when it comes to segmentation. From

the heatmaps, it is also easy to see where the model detects strong edges that resemble polyps.

5.1.3 Statistical Analysis

The improvement shown in the previous section was further investigated by training all models five times ($n = 5$). In doing so were also boxplots plotted and p-values computed which can be seen in Figure 5.4.

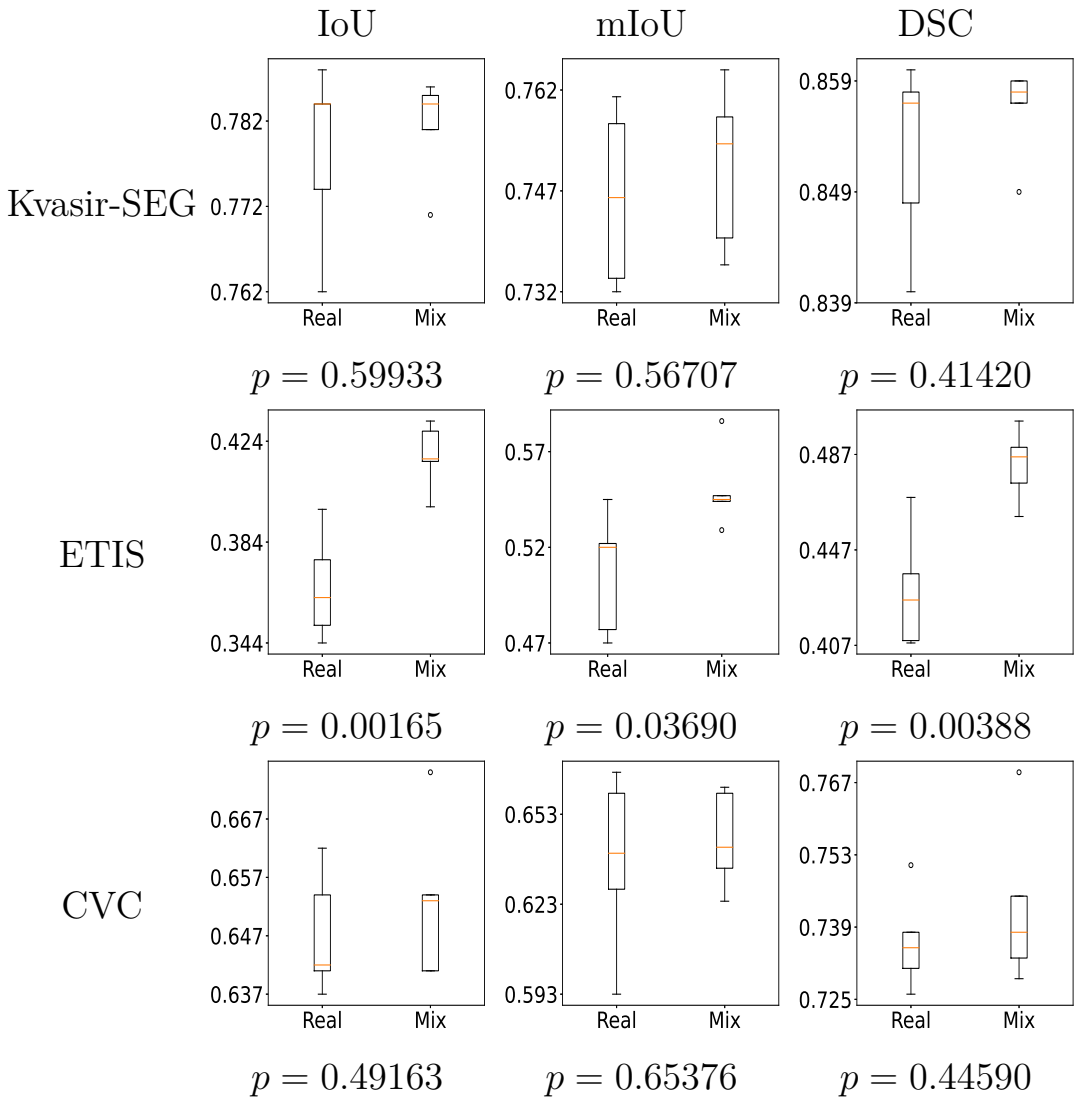


Figure 5.4: Boxplots and p-values used for comparing real and mixed data on key metrics IoU, mIoU, DSC on Kvasir-SEG [15], ETIS-Larib Polyp DB [49], and CVC-ClinicDB [48].

To evaluate if the difference between only training on real data versus a mix of real and fake was a null hypothesis and alternative hypothesis used seen in Equation 5.1.

$$H_0 : \mu_1 - \mu_2 = 0, H_a : \mu_1 - \mu_2 \neq 0; \tag{5.1}$$

The significance level we chose in our comparison is $\alpha = 0.05$. With this significance level, we can see that we only achieve a significant difference between only training on real versus mixed when validating on the ETIS-LaribDB dataset with p-values being $p < 0.05$. The p-values indicate that the improvements on the other datasets are random and thereby determine that there is not a significant difference between training on real or mixed data with validation on the Kvasir-SEG and CVC-ClinicDB dataset. It should be noted that only having five samples is very low when computing p-values and the minimal amount for boxplots and we should ideally have at least twenty to thirty samples to reduce uncertainty.

5.2 Discussion

Generated polyps in this section might be realistic, but still, we think that there is room for improvement. For example the FID score of generated polyps in Section 4.3.1 lower. To generate synthetic polyps with segmentation masks and achieve better FID scores, perhaps there is another method. Diffusion models have not been tested as much as compared to GANs and therefore might lack necessary frameworks and guides.

The segmentation experiment showed significant improvement on ETIS-LaribDB, but not on the other datasets. The reason that we might not achieve significant improvement on the datasets is that we train. It might also be because the ETIS-LaribDB dataset is the smallest of the three with low baseline scores. It is therefore more room for improvement from the baseline on ETIS-LaribDB compared to little room for improvement on the Kvasir-SEG and CVC-ClinicDB datasets.

Overall it is shown that synthetic images can improve segmentation model performance. Even in the worst cases does it, not appear that adding synthetic images significantly worsens the segmentation models. This indicates that the information learned from the generated images might at the very least be the same as real images.

5.3 Summary

In this chapter is the RePolyp framework introduced to generate polyp images with a segmentation mask. The generated images are used to increase the performance of a U-Net model, where we test one model only trained on real images and the other on a mix of real and generated images. The segmentation task is validated 3 times one against the Kvasir-SEG dataset as well as cross-dataset validation with the ETIS-LaribDB and CVC-ClinicDB datasets. We see a significant improvement on the ETIS-LaribDB dataset when training a segmentation model with synthetic polyps.

Chapter 6

Conclusion and Future Work

6.1 Summary and Contributions

Generative models have recently gained huge traction the recent years not only AI generated text but also AI generated images. Typically have GANs dominant when it comes to generative imagery, however in recent years have diffusion models challenged its dominance. This thesis has explored the use of diffusion models namely DDPM to generate synthetic polyps to address the data deficiency issue in the medical domain.

Early detection of polyps in the GI-tract is key to lowering the possibility and probability of developing deadly cancer. Between 14% to 30% polyps are missed during colonoscopy due to human error. Incorporating systems that use DL has shown to be great at reducing this miss rate. DL algorithms however require large amounts of data to be generalizable.

To train our generative models was pre-training on a large number of unlabeled images in the GI-tract proposed. Models capable of generating polyp images and clean colon images were then trained using transfer learning by fine-tuning the pre-trained models. The generalizability of these generative models was addressed by using regularization most notably dropout.

Generated polyps were presented to experts to asses the realism through a questionnaire. Furthermore, a novel framework to generate synthetic polyps with segmentation masks RePolyp was created. Generated polyps in conjunction with real polyps from this framework were then compared to only real polyps when training a simple segmentation model. The results from the segmentation showed a significant improvement for one dataset and inconclusive on two datasets. The two inconclusive cases see non-significant improvements.

The contributions can be summarized by addressing the objectives described in Section 1.2.

Objective 1 Generate synthetic images from the GI-tract by training diffusion

models on the data collected in the thesis. The generated samples should ideally be of the same quality and diversity as the data they were trained on. The generative models should be able to generate a complete image or use inpainting.

This objective relates to generating synthetic data. Throughout this thesis have we trained models able to generate general GI-tract images, polyp images, and clean colon images. The models were trained with varying amounts of dropout and images were evaluated based on their FID score. The completely generated images proved to achieve better results than inpainting.

Objective 2 The second objective is training segmentation models either on real data or a mix of real and synthetic data. Investigation of the performance of segmentation models when trained on real or mixed data.

This objective stems from possible usages for synthetic data. The result of this objective relies on the results of the previous objective. It was shown that adding synthetic data may improve segmentation models and sometimes even significantly improve results.

Objective 3 The third objective presents generated images to domain experts to assess realism. The results are a qualitative assessment that will indicate whether or not the synthetic images are indistinguishable from real images.

This objective comes from assessing generated polyps based on human perception. The results show strong signs that generated polyps are indistinguishable from fake ones. The assessment should however include more participants with a stronger background in the medical domain such as medical doctors or gastroenterology consultants to say anything with more certainty.

The research question this thesis tried to address was as we recall the following:

> *Can synthetic polyp images look realistic and be used to improve the performance of segmentation models?*

There seems to be evidence that synthetically generated polyps are of high realism. This is based on the qualitative assessment that our participants conducted through the questionnaire. The results however can be further justified by including more participants to reduce the uncertainty. The improvement of using synthetic polyps to train segmentation on the hand is split. The improvement observed in this thesis may only be comparable to that of introducing common generalization techniques.

6.2 Future Work

Super-resolution imaging Images generated in this thesis are of size 128×128. This is a relatively small resolution this can be solved by training a super-resolution to upscale $4x$ for example from 128×128 to 512×512 while still maintaining good image

quality. It is possible to upscale images even further than this as this might produce very visible artifacts. We chose not to any super-resolution both because of time limitations and because super-resolution can hallucinate image features and therefore needs to be carefully executed when used on medical data.

Conditional diffusion models This thesis has utilized unconditional DDPMs, however, conditional DDPMs have been shown to be quite effective. To train, a conditional DDPM are class labels required and you train both a diffusion and classifier. Conditional DDPM have been shown to generally overfit faster than unconditional and were therefore not used in this thesis for polyp generation and might be more suited with larger amounts of label data. We can on the other hand take advantage of the many different labeled classes and images in HyperKvasir mentioned in Section 3.2.1 to train a conditional DDPM.

Diffusion GAN When it comes to image generation is usually image quality a high priority, thus are GANs or diffusion models are preferred. It could be interesting to do a comparison between the two for image generation in the medical domain. The comparison could use state-of-the-art pre-trained polyp generation. Moreover, is combing two architectures and seeing how well they perform as suggested in [72] which uses a Diffusion-GAN an interesting idea. This architecture addresses the issue GANs we training stability and proposes to leverage forward diffusion chains to generate Gaussian-mixture distributed instance noise. The Diffusion-GAN consists of three main components, the adaptive diffusion process, a diffusion timestep-dependent discriminator, and a generator. The Diffusion-GAN architectures show promising results for giving consistent and helpful guidance for the generator.

Removing text from images Section 2.1.1.1.1 shows an example of a polyp from Kvasir-SEG. From that image can we see that is some text on the left side of the image that contains some sort of metadata at least date and time. Text on the left side of images is common in Kvasir-SEG and might therefore need to be addressed. This metadata might cause leaks when we try to generate synthetic images as models have "seen" this text. One approach to address this issue is to inpaint regions where the text is and thereby remove the text information.

Using synthetic data for classification task This thesis used synthetic polyps images with segmentation masks to improve segmentation models. The images however were not of the highest realism even though they improved the training of the U-Net segmentation models. This thesis also generated synthetic polyp images that were of much higher quality. These images could be used in an attempt to improve a classifier that determines if an image has a polyp or not (binary classification). The generation of images can be extended to generate images from the 23 different classes in the labeled part in HyperKvasir and then use them for classification (multiclass classification).

Bibliography

[1] Hyuna Sung et al. "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries." In: *CA: A Cancer Journal for Clinicians* 71.3 (2021), pp. 209–249. DOI: https://doi-org.ezproxy.uio.no/10.3322/caac.21660. eprint: https://acsjournals-onlinelibrary-wiley-com.ezproxy.uio.no/doi/pdf/10.3322/caac.21660. URL: https://acsjournals-onlinelibrary-wiley-com.ezproxy.uio.no/doi/abs/10.3322/caac.21660.

[2] Kevin J Moore, Daniel A Sussman, and Tulay Koru-Sengul. "Age-Specific Risk Factors for Advanced Stage Colorectal Cancer, 1981-2013." en. In: *Prev Chronic Dis* 15 (Aug. 2018), E106.

[3] Gilberto Lopes et al. "Early Detection for Colorectal Cancer: ASCO Resource-Stratified Guideline." In: *Journal of Global Oncology* 5 (2019). PMID: 30802159, pp. 1–22. DOI: 10.1200/JGO.18.00213. eprint: https://doi.org/10.1200/JGO.18.00213. URL: https://doi.org/10.1200/JGO.18.00213.

[4] Rebecca L Siegel et al. "Colorectal cancer statistics, 2020." en. In: *CA Cancer J Clin* 70.3 (Mar. 2020), pp. 145–164.

[5] Jeroen C. van Rijn et al. "Polyp Miss Rate Determined by Tandem Colonoscopy: A Systematic Review." In: *Official journal of the American College of Gastroenterology | ACG* 101.2 (2006). ISSN: 0002-9270. URL: https://journals.lww.com/ajg/Fulltext/2006/02000/Polyp_Miss_Rate_Determined_by_Tandem_Colonoscopy_.25.aspx.

[6] Kunio Doi. "Computer-Aided Diagnosis in Medical Imaging: Historical Review, Current Status and Future Potential." In: *Computerized medical imaging and graphics : the official journal of the Computerized Medical Imaging Society* 31 (June 2007), pp. 198–211. DOI: 10.1016/j.compmedimag.2007.02.002.

[7] Vajira Thambawita et al. "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.

[8] Anmol Arora and Ananya Arora. "Generative adversarial networks and synthetic patient data: current challenges and future perspectives." en. In: *Future Healthc. J.* 9.2 (July 2022), pp. 190–193.

[9] Anmol Arora and Ananya Arora. "Synthetic patient data in health care: a widening legal loophole." en. In: *Lancet* 399.10335 (Apr. 2022), pp. 1601–1602.

[10] P.J. Denning et al. "Computing as a discipline." In: *Computer* 22.2 (1989), pp. 63–70. DOI: 10.1109/2.19833.

[11] Terese Winslow. *NCI Dictionary of Cancer terms*. 2019. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/gastrointestinal-tract.

[12] Noam Shussman and Steven Wexner. "Colorectal polyps and polyposis syndromes." In: *Gastroenterology report* 2 (Feb. 2014), pp. 1–15. DOI: 10.1093/gastro/got041.

[13] Michael B Wallace. "Endoscopic removal of polyps in the gastrointestinal tract." en. In: *Gastroenterol. Hepatol. (N. Y.)* 13.6 (2017), pp. 371–374.

[14] T Muto, H J R Bussey, and B C Morson. "The evolution of cancer of the colon and rectum." en. In: *Cancer* 36.6 (Dec. 1975), pp. 2251–2270.

[15] Debesh Jha et al. "Kvasir-seg: A segmented polyp dataset." In: *International Conference on Multimedia Modeling*. Springer. 2020, pp. 451–462.

[16] Hanna Borgli et al. "HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy." In: *Scientific Data* 7.1 (2020), p. 283. ISSN: 2052-4463. DOI: 10.1038/s41597-020-00622-y. URL: https://doi.org/10.1038/s41597-020-00622-y.

[17] Terese Winslow. *NCI Dictionary of Cancer terms*. 2005. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/upper-endoscopy.

[18] Terese Winslow. *NCI Dictionary of Cancer terms*. 2022. URL: https://www.cancer.gov/publications/dictionaries/cancer-terms/def/colonoscopy.

[19] Afsaneh Jalalian et al. "Foundation and methodologies in computer-aided diagnosis systems for breast cancer detection." en. In: *EXCLI J.* 16 (Feb. 2017), pp. 113–137.

[20] Macedo Firmino et al. "Computer-aided detection system for lung cancer in computed tomography scans: review and future prospects." en. In: *Biomed. Eng. Online* 13.1 (Apr. 2014), p. 41.

[21] Hans-Dieter Wehle. "Machine Learning, Deep Learning, and AI: What's the Difference?" In: July 2017.

[22] Theodoros Evgeniou and Massimiliano Pontil. "Support Vector Machines: Theory and Applications." In: vol. 2049. Sept. 2001, pp. 249–257. ISBN: 978-3-540-42490-1. DOI: 10.1007/3-540-44673-7_12.

[23] Shi Na, Liu Xumin, and Guan Yong. "Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm." In: *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*. 2010, pp. 63–67. DOI: 10.1109/IITSI.2010.74.

[24] Maximilian Hüttenrauch, Adrian Šošić, and Gerhard Neumann. "Deep Reinforcement Learning for Swarm Systems." In: (2018). DOI: 10.48550/ARXIV.1807.06613. URL: https://arxiv.org/abs/1807.06613.

[25] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. DOI: 10.48550/ARXIV.1312.5602. URL: https://arxiv.org/abs/1312.5602.

[26] David Silver et al. *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*. 2017. DOI: 10.48550/ARXIV.1712.01815. URL: https://arxiv.org/abs/1712.01815.

[27] Gavin Adrian Rummery and Mahesan Niranjan. "On-line Q-learning using connectionist systems." In: 1994.

[28] Christopher Watkins and Peter Dayan. "Technical Note: Q-Learning." In: *Machine Learning* 8 (May 1992), pp. 279–292. DOI: 10.1007/BF00992698.

[29] Matthew J. Hausknecht and Peter Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs." In: *CoRR* abs/1507.06527 (2015). arXiv: 1507.06527. URL: http://arxiv.org/abs/1507.06527.

[30] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65 6 (1958), pp. 386–408.

[31] Frederic B. Fitch. “Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133.” In: *Journal of Symbolic Logic* 9.2 (1944), pp. 49–50. DOI: 10.2307/2268029.

[32] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. “Rectifier nonlinearities improve neural network acoustic models.” In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[33] J. Kiefer and J. Wolfowitz. “Stochastic Estimation of the Maximum of a Regression Function.” In: *The Annals of Mathematical Statistics* 23.3 (1952), pp. 462–466. ISSN: 00034851. URL: http://www.jstor.org/stable/2236690 (visited on 03/26/2023).

[34] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2017. DOI: 10.48550/ARXIV.1711.05101. URL: https://arxiv.org/abs/1711.05101.

[35] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[36] Keiron O’Shea and Ryan Nash. *An Introduction to Convolutional Neural Networks*. 2015. arXiv: 1511.08458 [cs.NE].

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[38] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. *Regularization for Deep Learning: A Taxonomy*. 2017. arXiv: 1710.10686 [cs.LG].

[39] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” In: *Journal of Machine Learning Research* 15 (June 2014), pp. 1929–1958.

[40] Connor Shorten and Taghi M. Khoshgoftaar. “A survey on Image Data Augmentation for Deep Learning.” In: *Journal of Big Data* 6.1 (2019), p. 60. ISSN: 2196-1115. DOI: 10.1186/s40537-019-0197-0. URL: https://doi.org/10.1186/s40537-019-0197-0.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning Internal Representations by Error Propagation.” In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. ISBN: 026268053X.

[42] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[43] Ian J. Goodfellow et al. “Generative Adversarial Nets.” In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Montreal, Canada: MIT Press, 2014, pp. 2672–2680.

[44] Jascha Sohl-Dickstein et al. “Deep Unsupervised Learning using Nonequilibrium Thermodynamics.” In: *CoRR* abs/1503.03585 (2015). arXiv: 1503.03585. URL: http://arxiv.org/abs/1503.03585.

[45] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. DOI: 10.48550/ARXIV.2006.11239. URL: https://arxiv.org/abs/2006.11239.

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. *Denoising Diffusion Implicit Models*. 2020. DOI: 10.48550/ARXIV.2010.02502. URL: https://arxiv.org/abs/2010.02502.

[47] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[48] Jorge Bernal et al. "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians." en. In: *Comput. Med. Imaging Graph.* 43 (July 2015), pp. 99–111.

[49] Juan Silva et al. "Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer." en. In: *Int. J. Comput. Assist. Radiol. Surg.* 9.2 (Mar. 2014), pp. 283–293.

[50] Konstantin Pogorelov et al. "KVASIR: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection." In: June 2017. DOI: 10.1145/3083187.3083212.

[51] Mathias Kirkerød. "Unsupervised Preprocessing of Medical Imaging Data with Generative Adversarial Networks." MA thesis. University of Oslo, 2019.

[52] Vajira Thambawita et al. "SinGAN-Seg: Synthetic training data generation for medical image segmentation." In: *PLOS ONE* 17.5 (May 2022), pp. 1–24. DOI: 10.1371/journal.pone.0267976. URL: https://doi.org/10.1371/journal.pone.0267976.

[53] Prafulla Dhariwal and Alex Nichol. *Diffusion Models Beat GANs on Image Synthesis*. 2021. DOI: 10.48550/ARXIV.2105.05233. URL: https://arxiv.org/abs/2105.05233.

[54] Jia Deng et al. "Imagenet: A large-scale hierarchical image database." In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[55] Fisher Yu et al. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop." In: *arXiv preprint arXiv:1506.03365* (2015).

[56] Robin Rombach et al. *High-Resolution Image Synthesis with Latent Diffusion Models*. 2022. arXiv: 2112.10752 [cs.CV].

[57] Andreas Lugmayr et al. *RePaint: Inpainting using Denoising Diffusion Probabilistic Models*. 2022. DOI: 10.48550/ARXIV.2201.09865. URL: https://arxiv.org/abs/2201.09865.

[58] Dorothy Cheng and Edmund Y. Lam. *Transfer Learning U-Net Deep Learning for Lung Ultrasound Segmentation*. 2021. DOI: 10.48550/ARXIV.2110.02196. URL: https://arxiv.org/abs/2110.02196.

[59] Guangyong Chen et al. "Rethinking the Usage of Batch Normalization and Dropout in the Training of Deep Neural Networks." In: *CoRR* abs/1905.05928 (2019). arXiv: 1905.05928. URL: http://arxiv.org/abs/1905.05928.

[60] Zhenxun Zhuang et al. *Understanding AdamW through Proximal Methods and Scale-Freeness*. 2022. arXiv: 2202.00089 [cs.LG].

[61] Mingkun Tan, Daniel Langenkämper, and Tim W Nattkemper. "The impact of data augmentations on deep learning-based marine object classification in benthic image transects." en. In: *Sensors (Basel)* 22.14 (July 2022), p. 5383.

[62] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[63] Tim Salimans et al. *Improved Techniques for Training GANs*. 2016. DOI: 10.48550/ARXIV.1606.03498. URL: https://arxiv.org/abs/1606.03498.

[64] Nicholas Carlini et al. *Extracting Training Data from Diffusion Models*. 2023. DOI: 10.48550/ARXIV.2301.13188. URL: https://arxiv.org/abs/2301.13188.

[65] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[66] Pavel Iakubovskii. *Segmentation Models Pytorch*. https://github.com/qubvel/segmentation_models.pytorch. 2019.

[67] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[68] Sara Hosseinzadeh Kassani et al. *Automatic Polyp Segmentation Using Convolutional Neural Networks*. 2020. arXiv: 2004.10792 [eess.IV].

[69] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. 2021. arXiv: 2102.09672 [cs.LG].

[70] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. "CIFAR-10 (Canadian Institute for Advanced Research)." In: (). URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[71] Tim Salimans et al. *PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications*. 2017. arXiv: 1701.05517 [cs.LG].

[72] Zhendong Wang et al. *Diffusion-GAN: Training GANs with Diffusion*. 2022. arXiv: 2206.02262 [cs.LG].

Appendices

Appendix A

Paper

RePolyp: A Framework for Generating Realistic Colon Polyps with Corresponding Segmentation Masks using Diffusion Models

Alexander K. Pishva[1,2], Vajira Thambawita[2], Jim Torresen[1], and Steven A. Hicks[2]

[1]*University of Oslo, Norway*
[2]*SimulaMet, Norway*

***Abstract*—The field of synthetic medical data has become increasingly important due to the urgent need for large and diverse datasets in the medical sector. Using diffusion models in data generation has created more authentic and varied medical data. In this study, a framework is presented that utilizes diffusion models trained on openly accessible data to generate realistic-looking colon polyps, along with their corresponding ground truth masks. The usefulness of the synthetic polyps is evaluated by using them to train segmentation models designed to segment colon polyps in real-world images. The results demonstrate that the generated synthetic data is highly accurate and suggest that including synthetic polyps in the training dataset improves the predictive performance and generalization of the segmentation models. When the training dataset consists of pre-generated synthetic data from our model, we achieve a mean intersection over union (mIoU) improvement of 4.64% on the validation data and a 4.14% mIoU improvement when testing across different datasets. These results indicate that generating synthetic medical data using diffusion models is valuable for addressing the need for diverse and extensive medical datasets.**

***Index Terms*—computer-aided diagnosis, deep learning, polyp generation, machine learning, and segmentation.**

I. Introduction

Colorectal cancer is the second leading cause of cancer-related deaths for both men and women worldwide, with more than $935,000$ deaths and $1,900,000$ new colorectal cancer (including anus) cases estimated to occur in 2020 [1], which accounts for about one in ten cancer cases (10.0%) and deaths (9.4%). Data shows that the risk of colon cancer increases with age, with most occurring in people older than 50 [2]. However, colorectal cancer is highly treatable when diagnosed at a localized stage [3], with a 5-year relative survival rate of 90%. About 36% of patients are diagnosed at this early stage [4].

For colorectal cancer, colonoscopy is considered the gold standard as they provide a detailed look at the rectum and the entire large intestine. A colonoscopy uses a small flexible tube with a camera attached to the end that is inserted through the rectum. The camera provides the operating doctor with a continuous video feed to look for abnormalities in the colon. Polyps in the colon are abnormalities of varying sizes and can be precancerous or cancerous. If a polyp is detected during a colonoscopy, it is usually removed and examined by a pathologist to determine if it contains cancerous or precancerous cells. However, polyp detection during colonoscopies is prone to human error with miss rates between 14% to 30% [5].

In response to the high polyp miss rates, the use of machine learning (ML)-based methods to assist medical doctors in detecting these polyps has become a popular area of research. These systems are commonly referred to as computer aided diagnosis (CAD) systems and are meant to assist doctors in making their jobs easier and more efficient. However, the models these systems are based on usually require a lot of data to be generalizable [6]. Datasets in the medical domain are often small or lack the variety (like a lack of true positive findings) needed to train a model that generalizes well to unseen data. Therefore, in this paper, we aim to expand the availability of polyp images by generating fake polyps that can be used to train ML models or other applications like student training. The main contributions of this paper are as follows:

1) A framework for generating realistic-looking colon polyps with associated ground truth masks that can be used for ML model development and training.
2) Experiments that showcase how the generated data from our generative model can improve segmentation model performance and generalizability using a polyp segmentation use case.

The rest of this paper is organized as follows. First, we provide additional background and details on previous colon polyp image generation research. Next, we provide a comprehensive description of the framework employed to generate the synthetic colon polyp images. Following this, we discuss the experiments conducted for generating the synthetic polyps, which include details on the datasets used, model implementation and configuration, and evaluation criteria. We then present preliminary experiments that use the generated polyps to train segmentation models designed to segment polyps collected from real-world colonoscopies. Lastly, we conclude with a discussion of the results obtained from our experiments and consider possible directions for future work.

II. Background and Related Work

Synthetic medical data generation has emerged as a crucial area of research in medical ML. Obtaining medical data can be expensive and raise privacy concerns, making synthetic data an attractive solution to increase the amount of data without

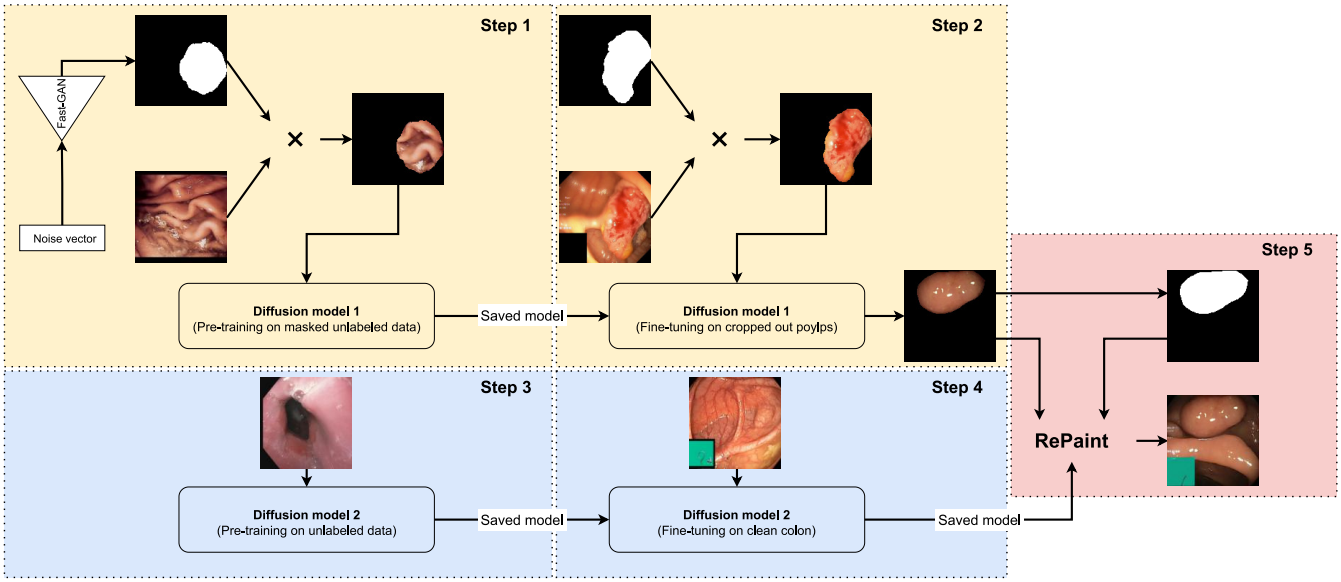


Fig. 1: Framework of RePolyp. **Step 1** Pre-training on masked data. **Step 2** Fine-tuning on cropped-out polyps. **Step 3** Pre-training a second diffusion model. **Step 4** Fine-tuning second model on clean colon. **Step 5** Inpainting using diffusion model 2 and cropped-out images.

compromising patient privacy. Examples of synthetic data include generating MRI images for brain image analysis [7] and ECG signals for detecting heart abnormalities [8], among others. The use of synthetic data has the potential to accelerate medical research and improve the accuracy of medical algorithms. However, it is essential to ensure that the synthetic data is realistic and representative of real-world scenarios to avoid negative impacts on clinical decision-making.

This paper focuses on generating synthetic data for the gastrointestinal (GI) tract, specifically colon polyps. Automatic polyp generation is a topic in medical imaging and CAD that involves the creation of realistic computer-generated polyps for research and training purposes. Various methods have been proposed for automatic polyp generation, with most relying on generative adversarial networks (GANs). For instance, Shin et al. [7] introduced a conditional GAN-based framework that generates realistic-looking colon polyps given a background image and mask. They demonstrated that the generated synthetic polyp images could be used as additional training samples to enhance the performance of polyp detection models. In 2022, Fagereng et al. presented PolypConnect [9], which is a pipeline that inpaints polyps into background images of clean colon. Their method is based on EdgeConnect [10], which is a two-stage adversarial model that first produces lines/edges before completing the image. Similar to Shin et al., the authors demonstrated that the synthetic polyps were of adequate quality to be used for training segmentation models for real-world colonoscopy images.

Our work aims to improve upon previous methods by using diffusion models for generating synthetic colon polyps. Diffusion models have demonstrated the ability to produce high-fidelity images with less noise than traditional GANs. Additionally, diffusion models are generally more stable than GANs, which are prone to mode collapse during optimization. However, the sampling process in diffusion models can be computationally expensive and slower than GANs. Nonetheless, we believe that using diffusion models for generating synthetic data for colon polyps will improve the quality and diversity of the generated images, ultimately enhancing the performance of segmentation models trained on this synthetic data.

III. Polyp Generation Framework

This section describes the polyp generation framework, shown in Figure 1, all the way from initial training to synthetically generated polyps. The framework uses guided diffusion models [11], one to generate synthetic cropped-out polyp images and the other to in-paint the generated partial images. The framework is based on the RePaint scheme [12]. Repaint is an inference scheme that relies on pre-trained unconditional Denoising Diffusion Probabilistic Models (DDPM) for inpainting generation. We use this scheme to inpaint clean colon background for synthetic cropped-out polyps. Partial images can be obtained by multiplying original images with masks indicated by $\times$ in Step 1 and Step 2 in Figure 1.

The polyp generation process can be broken down into five distinct steps as labeled in Figure 1.

In **Step 1**, we employ a FastGAN-based [13] model to generate masks. This model is trained on a substantial number of masked images, resulting in our first pre-trained model. Masking an image entails revealing only the white parts of an image mask in the corresponding original image. The pre-trained model's weights are saved for fine-tuning in the subsequent step.

Step 2 involves using the diffusion model trained in the previous step on actual cropped-out polyps. The cropped-out polyps are obtained by eliminating all parts of the images except for the ground truth region. This step fine-tunes the model to generate realistic cropped-out polyps.

In **Step 3**, a new diffusion model is trained on a large dataset of images, enabling the model to gain a general understanding of the appearance of the GI tract. This results in a pre-trained model capable of generating images resembling those found in the GI tract. The pre-trained model's weights are saved for fine-tuning in the following step.

Step 4 loads the diffusion model from the previous step and fine-tunes it on clean colon images. The model can now generate colon images without polyps. The fine-tuned model's weights are saved for use in the final step.

In **Step 5**, the final step, we first create a corresponding segmentation mask for our polyps from Step 2 using thresholding. We then employ our ground truth and the corresponding segmentation mask in conjunction with our diffusion model from Step 4. This is done using the RePaint inference scheme [12], which generates a clean probabilistic background for our cropped-out polyps while simultaneously incorporating the polyp image's segmentation mask.

IV. Polyp Generation Experiments

This section describes all experiments, including details on the datasets used, implementation of the ML models, framework configuration, and evaluation strategy. Polyp generation can be achieved through pre-training on unlabeled data and fine-tuning on images containing polyps. This can be seen as doing **Step 1** and **Step 2** in our pipeline without using segmentation masks. However, this approach will not have a corresponding segmentation mask for our generated polyp image, meaning that these synthetic polyp images can only be used for classification tasks, not segmentation tasks.

A. Datasets

To generate polyps and their corresponding probabilistic background, the HyperKvasir dataset was used. This dataset consists of $100,000$ unlabeled images from the GI tract and $1,000$ polyp images, each accompanied by a corresponding mask, collectively known as Kvasir-SEG. Additionally, 851 images featuring pathological findings (ulcerative colitis) and $1,018$ images showcasing anatomical landmarks (ileum and cecum) were selected to simulate a clean colon. The unlabeled data were utilized in both Steps 1 and 3 of our diffusion models to acquire a comprehensive understanding of the GI tract. Cropped polyps were employed in Step 2 for model fine-tuning. The dataset was split between 80% training and 20% validation for polyp images, 80% training and 20% validation for clean images, and 95% training and 5% validation for unlabeled data.

B. Experimental Setup

All model training and data sampling were performed on a single NVIDIA V100 with 32GB RAM, part of an NVIDIA DGX-2 with 16 V100 GPUs. The models were implemented using the ML framework PyTorch [14] version $1.12.1$ with CUDA version $11.3.1$. The diffusion models were trained unconditionally, meaning we did not condition on any context when generating an image; thus, generated images should only resemble its training data distribution. All models were trained using the AdamW [15] optimizer, and loss was calculated using the L_{simple} loss function introduced by Ho et al. [16]. The batch size used during training is 32, and the input images were scaled to a pixels size of 128×128. The reason for the small image size is computational efficiency. If images need to be larger, a super-resolution diffusion model can be trained to upscale images $4\times$ the original size without losing quality. The implementation of the experiments is available on GitHub[1].

TABLE I: FID score for cropped polyps inpainted with different backgrounds.

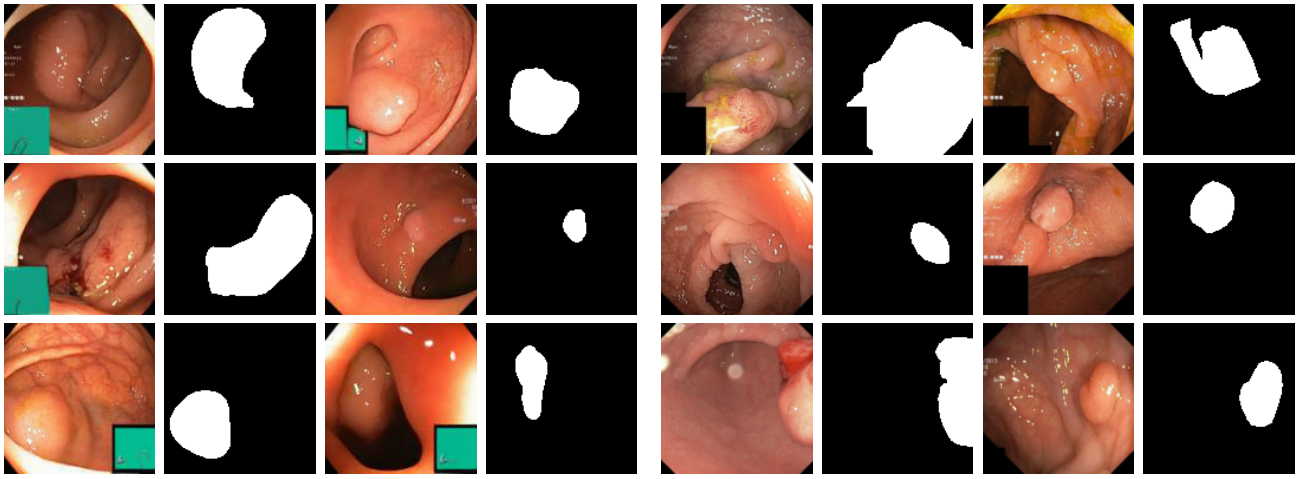
Model	FID
Pretrained unlabeled	138.38
Fine-tuned clean	128.83
Fine-tuned polyps	**93.43**

C. Experiments

This section describes the experiments generating synthetic polyp using the presented RePolyp framework. The Kvasir-SEG dataset comprises numerous polyp images, each accompanied by a corresponding segmentation mask. To capitalize on the vast amount of data in the unlabeled portion of HyperKvasir, we preprocess all images in this dataset by applying masks generated using a GAN model, as detailed in Step 1 presented in Section IV. This approach enables our diffusion model to learn the generation of images with a black background and specific features from the unlabeled dataset. Fine-tuning is conducted on cropped polyp regions to develop a DDPM capable of generating synthetic cropped polyps. We implement various dropout rates, a standard regularization technique in neural networks, to mitigate overfitting by randomly deactivating a subset of neurons during training.

To create a probabilistic background for the cropped polyps, diffusion models are trained using different fine-tuning images. The objective is to ensure that the resulting unconditional DDPM generates a background distinct from polyp features. In summary, our method generates a background based on the cropped polyp, while the approach in [9] generates a polyp given its background. The differences between these methods extend beyond inpainting background versus polyp; they also involve the underlying generative architectures, GANs and DDPMs. While DDPMs have slower sampling speeds, GANs struggle with diversity and mode coverage. Both architectures, however, can produce high-quality samples, as evidenced by the trilemma in generative models.

[1]https://www.github.com/simula/repolyp



(a) Generated polyp images from our model and their corresponding segmentation masks.

(b) Polyp images from Kvasir-SEG and their corresponding segmentation masks.

Fig. 2: Comparison between our generated and Kvasir-SEG polyp images.

D. Results and Discussion

The primary metric used to evaluate the synthetic polyp images was fréchet inception distance (FID) [17]. FID is a non-negative score that compares the distribution between generated images with the distribution of a set of real images. Generated images that result in FID scores closer to 0 are considered to be more similar to the real distribution, thereby better and more realistic. FID has been the de-facto standard metric for capturing both variety and fidelity in generative models. Therefore, we use it as our default metric for overall sample quality assessments while not directly assessing intra-label diversity.

Considering the FID scores presented in Table I, it might be tempting to choose a diffusion model fine-tuned on polyps. However, this would not guarantee that the model would not inadvertently generate additional polyps. As a result, we opt for the clean background, ensuring no polyps are generated in the background and achieving a FID score of 128.83, which surpasses the performance of the pre-trained model.

V. Poylp Segmentation Experiments

This section describes the experiments for validating the synthetic data by training models using a combination of real and synthetic images of the human colon.

A. Datasets

Similar to the polyp generation experiments, we use several open datasets to train the polyp segmentation model, including Kvasir-SEG [18], ETIS Larib Poylp DB [19], and CVC-ClinicDB [20]. Kvasir-SEG contains $1,000$ polyp images with corresponding segmentation masks, where we used 80% of the dataset for training and 20% for validation. The polyp images from Kvasir-SEG used in Section IV-B are the same used in this here with the same training validation split. The ETIS Larib Polyp DB [19] and CVC-ClinicDB [20] datasets were also used, containing 196 and 612 images, respectively, with corresponding ground truth. These datasets were used for validation across different datasets.

B. Experimental Setup

Similar to the experimental setup used to generate synthetic polyps described in Section IV-B, we use a similar setup for the polyp segmentation experiments. Notable differences include that we use a batch size of 4 during training, a batch size of 1 during validation, and the Adam [21] optimizer rather than AdamW. AdamW is a modified version of Adam that uses weight decay which is another regularization technique. Hardware and software dependencies remain the same as described in Section IV-B.

C. Experiments

To train a U-Net-based model [22] for segmenting polyp images in colonoscopy frames, it is necessary to obtain images containing polyps and their corresponding ground truth annotations. We perform three evaluations, each involving the training of two U-Net models. One model is exclusively trained on real data, while the other is trained on a combination of real and synthetic data. Both models are trained for 20 epochs using early stopping and without incorporating any data augmentation techniques. The metrics employed to assess segmentation performance include Intersection over Union (IoU), mean Intersection over Union (mIoU), Dice coefficient (DSC), precision, and recall.

In the first evaluation, we compare the performance of a model trained on a dataset of 800 real images to that of a model trained on a dataset containing the same 800 real images supplemented with 800 synthetic images. Both models are validated against a set of 200 real images from Kvasir-SEG. In the second and third evaluations, we employ the same two

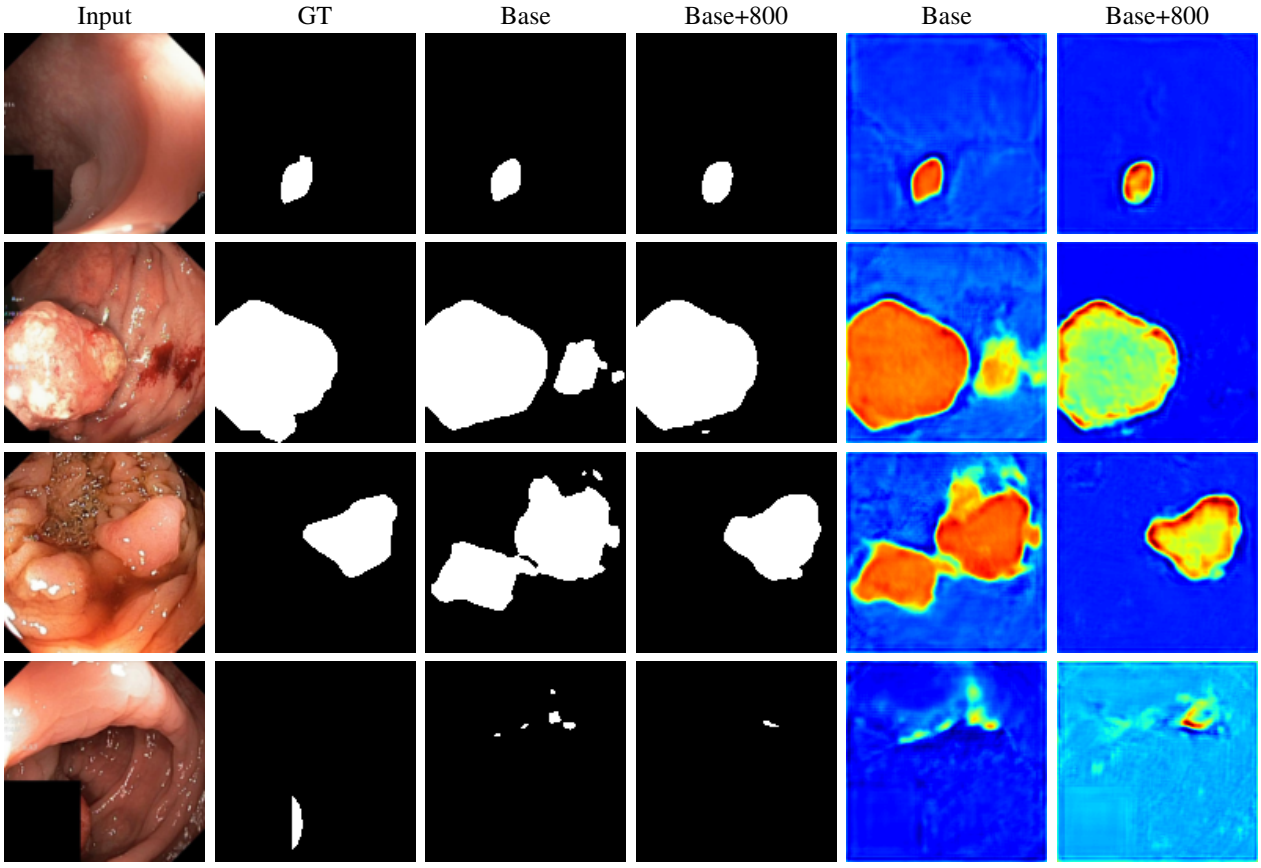


Fig. 3: Visual segmentation performance on Kvasir-SEG images using a U-Net architecture. Black and white images are segmentation masks, and the last two columns represent heatmaps. From left to right; Input image, GT - Ground Truth, real images, real images + 800 synthetic images, real images, real images + 800 synthetic images.

datasets used in the first evaluation but test their performance against the ETIS and CVC datasets, respectively.

D. Results and Discussion

Figure 2 compares six synthetic polyps and their corresponding ground truth images, as well as six images with ground truth from the Kvasir-SEG dataset. The first column displays examples of realistically generated synthetic polyps, while the third column demonstrates poorly generated synthetic polyps. In Tables II, III, and IV, the differences between the baseline dataset and the dataset augmented with synthetic images are examined. We observe an improvement in the mean intersection over union (mIoU) by 4.64%, 4.68%, and 5.10% on the respective validation datasets. Concurrently, there is an increase in the Dice similarity coefficient (DSC) by 2.02%, 10.54%, and 0.41%. Precision, which reflects the model's confidence in predicting polyp pixels, also increases by adding synthetic images to the dataset. However, it is essential to consider precision in conjunction with recall. The impact of the added synthetic data on recall is minimal. The improved performance on the validation data can be attributed to the increased diversity in the training data provided by the synthetic images. This augmentation appears to enhance the model's generalization capabilities. While adding synthetic data may also improve optimization, this aspect was not investigated in this study.

TABLE II: Validation 200 Kvasir-SEG images

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.762	0.732	0.840	0.871	0.821
+800	**0.785**	**0.766**	**0.857**	**0.913**	**0.826**
Chg %	3.02%	4.64%	2.02%	4.82%	0.61%

TABLE III: Validation ETIS Larib Polyp DB

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.351	0.470	0.408	0.583	0.709
+800	**0.396**	**0.492**	**0.451**	**0.604**	**0.727**
Chg %	12.82%	4.68%	10.54%	3.60%	2.54%

Figure 3 shows the results of two segmentation models on selected Kvasir-SEG images from the validation dataset. The first image appears to be easy to segment. In the second and third images, we see that our baseline model classifies more of the clean as polyps which will result in a lower precision

TABLE IV: Validation CVC-ClinicDB

Dataset	IoU	mIoU	DSC	Precision	Recall
Baseline	0.642	0.628	0.735	0.831	0.720
+800	**0.654**	**0.660**	**0.738**	**0.869**	**0.733**
Chg %	1.87%	5.10%	0.41%	4.57%	1.81%

score. The baseline model seemingly segments based on the texture of polyps, giving a somewhat uniform value to pixels segmented as polyps. The dataset with added synthetic data, on the other hand, seems to have a better understanding of polyp edges with darker red pixels being more frequent near polyp edges and more uniform values for clean parts. The increase in performance might therefore be because the model with synthetic data has a better idea of what clean parts of a colon look like. The last image is a challenge with it being very hard to detect the polyp. Both models fall short of detecting the polyp, but the +800 model classifies fewer pixels as polyps.

The generated polyps often generate green boxes in the left or right corners of the images. This is caused since clean images use green boxes and polyp images in Kvasir-SEG mostly use black boxes. This can be changed by adding a post-processing step of making green boxes black. It is also worth mentioning that generated polyps can be in areas where our model wants to generate these green boxes. This can be observed in our second generated image in Figure 2 that causes some semantic inconsistencies. This can be adjusted in post-processing by completing the boxes the model tries to generate, but we would then similarly need to remove areas where the box should be in our segmentation masks.

VI. Conclusion

This paper presents RePoylp, a novel framework for generating realistic colon polyps using diffusion models. The generated polyps are used in combination with real polyps to improve the training of polyp segmentation models. The results demonstrate improvements in all metrics on three distinct validation datasets when adding synthetic polyps to the training data, with the most improvements seen in mean intersection over union (mIoU) and precision. However, the precise reason for these improvements is uncertain, but perhaps the enhanced generation of clean colon regions compared to polyps may have contributed to the gains. This hypothesis is supported by the data imbalance, where clean colon pixels are much more prevalent than polyp pixels. Nonetheless, the generated data is still beneficial for training systems for polyp segmentation. We hope this framework can contribute to more robust and generalizable models by extending existing datasets with synthetic data.

VII. Acknowledgments

The research has benefited from the Experimental Infrastructure for Exploration of Exascale Computing (eX3), which is financially supported by the Research Council of Norway under contract 270053.

References

[1] H. Sung, J. Ferlay, R. L. Siegel *et al.*, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: A Cancer Journal for Clinicians*, vol. 71, no. 3, pp. 209–249, 2021.

[2] K. J. Moore, D. A. Sussman, and T. Koru-Sengul, "Age-Specific risk factors for advanced stage colorectal cancer, 1981-2013," *Prev Chronic Dis*, vol. 15, p. E106, 2018.

[3] G. Lopes, M. C. Stern, S. Temin *et al.*, "Early detection for colorectal cancer: Asco resource-stratified guideline," *Journal of Global Oncology*, no. 5, pp. 1–22, 2019, pMID: 30802159.

[4] R. L. Siegel, K. D. Miller, A. Goding Sauer *et al.*, "Colorectal cancer statistics, 2020," *CA Cancer J Clin*, vol. 70, no. 3, pp. 145–164, 2020.

[5] J. C. van Rijn, J. B. Reitsma, J. Stoker *et al.*, "Polyp miss rate determined by tandem colonoscopy: A systematic review," *Official journal of the American College of Gastroenterology*, vol. 101, no. 2, 2006.

[6] V. Thambawita, D. Jha, H. L. Hammer *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*, vol. 1, no. 3, pp. 1–29, 2020.

[7] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers *et al.*, "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)*, 2018, pp. 1–11.

[8] V. Thambawita, S. A. Hicks, J. Isaksen *et al.*, "Deepsynthbody: the beginning of the end for data deficiency in medicine," in *2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, 2021, pp. 1–8.

[9] J. Fagereng, V. Thambawita, A. M. Storas *et al.*, "Polypconnect: Image inpainting for generating realistic gastrointestinal tract images with polyps," in *2022 IEEE 35th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE Computer Society, 2022, pp. 66–71.

[10] K. Nazeri, E. Ng, T. Joseph *et al.*, "Edgeconnect: Structure guided image inpainting using edge prediction," in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, 2019.

[11] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," 2021.

[12] A. Lugmayr, M. Danelljan, A. Romero *et al.*, "Repaint: Inpainting using denoising diffusion probabilistic models," 2022.

[13] V. Thambawita, P. Salehi, S. A. Sheshkal, S. A. Hicks, H. L. Hammer, S. Parasa, T. d. Lange, P. Halvorsen, and M. A. Riegler, "Singan-seg: Synthetic training data generation for medical image segmentation," *PLOS ONE*, vol. 17, no. 5, pp. 1–24, 2022.

[14] A. Paszke, S. Gross, F. Massa *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[15] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2017.

[16] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.

[17] M. Heusel, H. Ramsauer, T. Unterthiner *et al.*, "Gans trained by a two time-scale update rule converge to a nash equilibrium," *CoRR*, vol. abs/1706.08500, 2017.

[18] D. Jha, P. H. Smedsrud, M. A. Riegler *et al.*, "Kvasir-seg: A segmented polyp dataset," in *International Conference on Multimedia Modeling*, 2020, pp. 451–462.

[19] J. Silva, A. Histace, O. Romain *et al.*, "Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 9, no. 2, pp. 283–293, 2014.

[20] J. Bernal, F. J. Sánchez, G. Fernández-Esparrach *et al.*, "WM-DOVA maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians," *Comput. Med. Imaging Graph.*, vol. 43, pp. 99–111, 2015.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

Appendix B

Interpolation

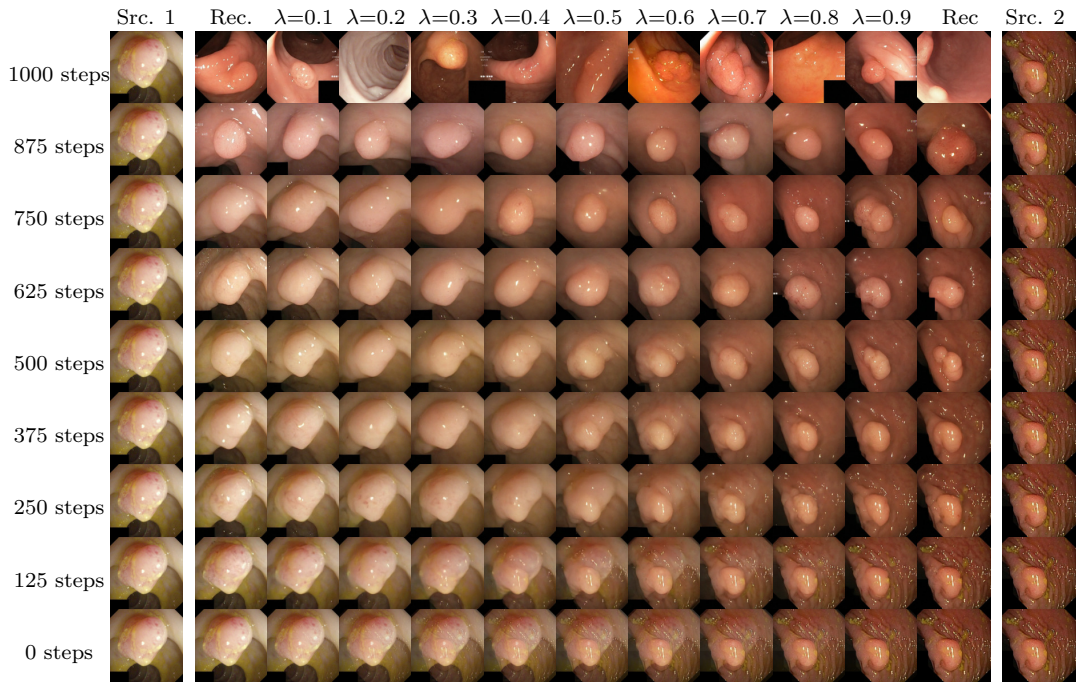


Figure B.1: Interpolation in latent space between two different polyp images.

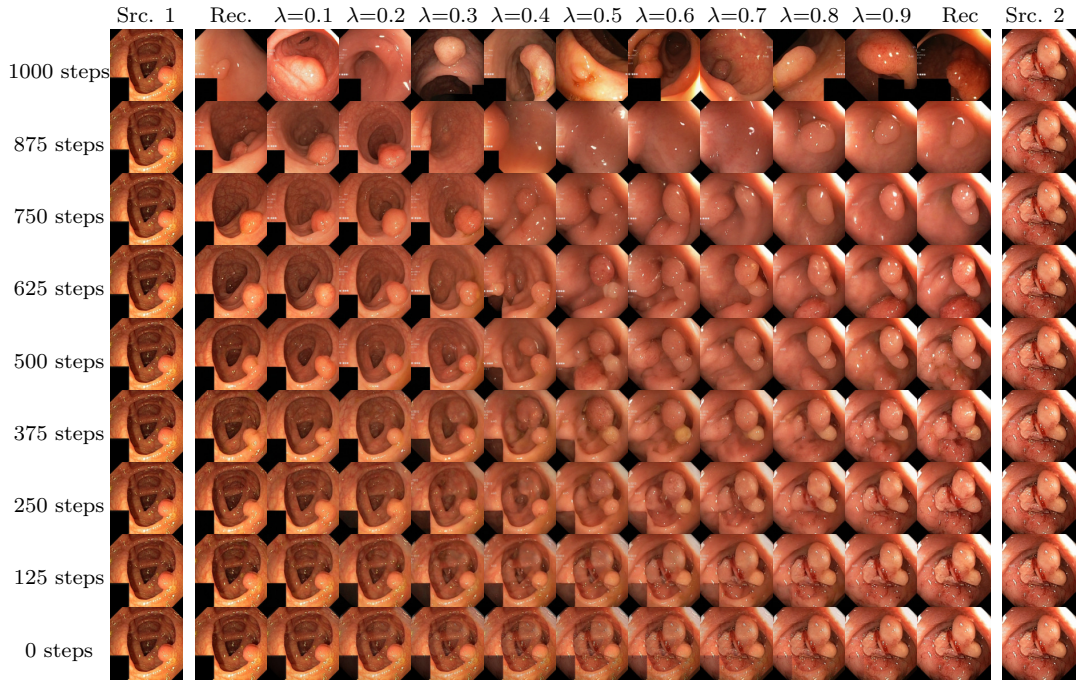


Figure B.2: Interpolation in latent space between two different polyp images.

Appendix C

Questionnaire

Polyps rating questionnaire

This study will present you with ten images of different polyps. Some images are real polyps, and some are generated (synthetic) polyps. Please look at the image carefully and answer some questions! Thanks a lot for your participation!

* indikerer at spørsmålet er obligatorisk

Please follow the following guidelines carefully before filling the form:
1. Please spend about 10s to look at an image.
2. Do not zoom images to inspect them. Use the original size of the image as given in the form.

What is your job title? *

Svaret ditt

How many years did you work with colonoscopy? *

Svaret ditt

Side 1 av 12

Neste

Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 1

Is this a real or a generated image? *

	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 2 av 12

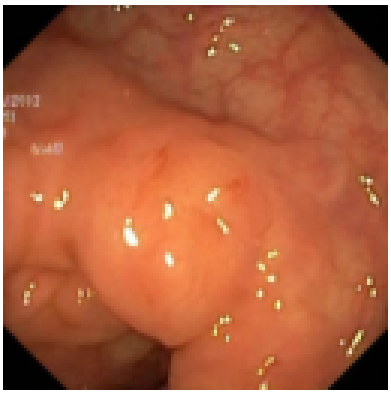
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 2

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 3 av 12

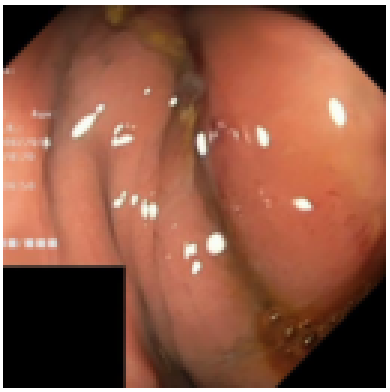
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 3

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 4 av 12

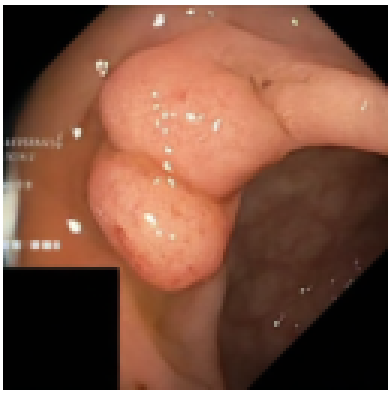
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 4

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 5 av 12

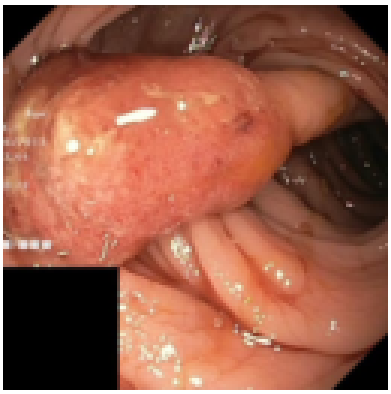
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 5

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 6 av 12

Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 6

Is this a real or a generated image? *

	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 7 av 12

Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 7

Is this a real or a generated image? *

	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 8 av 12

Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 8

Is this a real or a generated image? *

	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very less confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 9 av 12

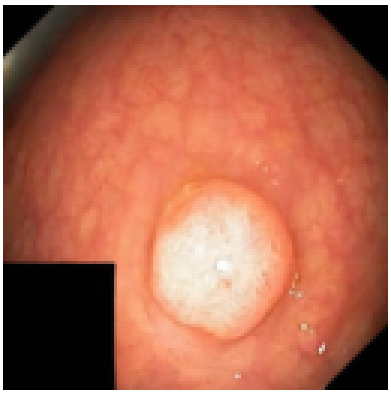
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 9

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 10 av 12

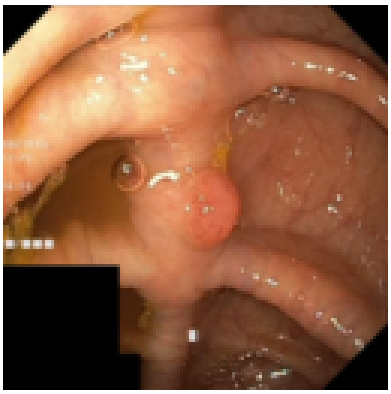
Tilbake Neste Tøm skjemaet

Polyps rating questionnaire

* indikerer at spørsmålet er obligatorisk

Image 10

Is this a real or a generated image? *



	1	2	3	4	5	6	7	8	9	10	
I am completely sure it is real	○	○	○	○	○	○	○	○	○	○	I am completely sure it is generated

What type of polyp does the image contain? *

Svaret ditt

Is the size of the polyp appropriate in context to its surroundings? *

	1	2	3	4	5	6	7	8	9	10	
Not appropriate	○	○	○	○	○	○	○	○	○	○	Very appropriate

Does the background appear generated? *

	1	2	3	4	5	6	7	8	9	10	
Completely sure it is a real background	○	○	○	○	○	○	○	○	○	○	Completely sure it is a generated background

Does the polyp appear generated? *

	1	2	3	4	5	6	7	8	9	10	
I am sure this is a real polyp	○	○	○	○	○	○	○	○	○	○	I am sure the polyp is generated

Does the polyp appear fitting into the given background and anatomy? *

	1	2	3	4	5	6	7	8	9	10	
It does not fit	○	○	○	○	○	○	○	○	○	○	It fits perfectly

How confident are you regarding your predicted histology? *

	1	2	3	4	5	6	7	8	9	10	
Very low confidence	○	○	○	○	○	○	○	○	○	○	Very high confidence

Side 11 av 12

Tilbake Neste Tøm skjemaet