Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

Mathias Ramm Haugland

Deep Learning for Polyp Detection from Synthetic Narrow-Band Imaging

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Ilangko Balasingham
Co-supervisor: Hemin Ali Qadir
June 2022

NTNU
Norwegian University of Science and Technology

Mathias Ramm Haugland

Deep Learning for Polyp Detection from Synthetic Narrow-Band Imaging

Master's thesis in Electronics Systems Design and Innovation
Supervisor: Ilangko Balasingham
Co-supervisor: Hemin Ali Qadir
June 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electronic Systems

NTNU
Norwegian University of
Science and Technology

Abstract

Colorectal cancer (CRC) has become a prevalent cancer type in developed countries, and screening programs have gained popularity due to their demonstrable preventive effect. Colonoscopy is regarded as the best-performing procedure for CRC screening, and different tools have been developed to ease the detection of adenomatous polyps during this examination. Among these are automatic polyp detection and narrow-band imaging (NBI).

In this thesis, the method CycleGAN is used to create different synthetic narrow-band imaging (SNBI) datasets from regular white-light imaging (WLI) data. The different datasets have been used independently to train the state-of-the-art object detection network EfficientDet-D0 for both one-class (polyp) and two-class (hyperplastic polyps vs. adenomas) detection. The best performing SNBI models have then been compared to models trained on the original WLI, as well as real NBI of the same polyps.

The results show that the proposed SNBI is able to detect polyps, especially hyperplastic polyps, easier than WLI. These findings, and results with real NBI, also indicate that NBI is a better modality for polyp detection and classification in general.

Despite flaws in, for instance, the datasets used, the experiments conducted with the generated SNBI show that GAN-based methods can be used for modality transformation in colonoscopy imaging. The generation of SNBI has an inference time of 5.3ms, making it applicable for real-time post-process image enhancement. Applying such techniques can be considered a novel approach to endoscopic image enhancement with a great potential for further development.

Sammendrag

Kolorektal kreft (CRC) har blitt en utbredt krefttype i utviklide land, og screeningprogrammer har blitt populære på grunn av sin beviselig preventive effekt. Koloskopi er ansett som den beste screeningmetoden for CRC, og forskjellige verktøy har blitt utviklet for å forenkle deteksjonen av adenomatøse polypper under denne eksaminasjonen. Blant disse er automatisk polyppdeteksjon og narrow-band imaging (NBI).

I denne oppgaven er metoden CycleGAN benyttet til å lage forskjellige syntetiske NBI (kalt SNBI) datasett fra vanlig white-light imaging (WLI) data. De forskjellige datasettene har blitt brukt uavhengig til å trene objektdeteksjonsnettverket EfficientDet-D0 for både polyppdeteksjon (one-class detection) og deteksjon som skiller mellom adenomer og hyperplastiske polypper (two-class detection). De beste SNBI-modellene har deretter blitt sammenlignet med modeller trent på orginal WLI, samt ekte NBI av de samme polyppene.

Resultatene viser at den foreslåtte SNBI-en greier å detektere polypper, spessielt hyperplastiske polypper, lettere enn WLI. Disse funnene, samt resultater med ekte NBI, indikerer også at NBI generelt er en bedre modalitet for polyppdeteksjon og -klassifisering.

Tross feil i blant annet datasettene som ble brukt viser eksperimentene som ble gjennomført med generert SNBI at GAN-baserte metoder kan bli brukt til modalitetstransformasjoner i koloskopiavbildning. Genereringen av SNBI har en kjøretid på 5.3ms, som gjør den anvendelig for post-prosesserende bildeforbedring i sanntid. Anvendelsen av slike teknikker kan ansees som en ny tilnærming til endoskopisk bildeforbedring og har stort potensiale for videre utvikling.

Preface

This thesis was carried out in the spring of 2022 under the supervision of Prof. Ilangko Balasingham and Dr. Hemin Ali Qadir. The topics it covers build on some of the work from my project report from the autumn 2021 [1]. This document concludes my master's degree in technology and five years of study at NTNU in Trondheim.

I would like to thank my supervisor Prof. Ilangko Balasingham for trusting me with this project. Although being a freshman in both deep learning and the field of medicine, I have been given an immense load of trust and freedom to conduct my experiments. My co-supervisor Dr. Hemin Ali Qadir deserves a big thanks for helping me navigate through the challenges that I have faced along the way. I hope my findings can somehow contribute to your research.

Most of this project was carried out in Oslo at Oslo University Hospital (OUS), and I want to thank prof. Balasingham's research group there for the four months I spent with them. Few master's students get the opportunity to gain such inspiring insight into scientific research and routines as I have. My sharp-witted office mate Dr. Martin Damrath deserves an extra shout-out for his valuable input on my project, as well as the great everyday discussions we shared.

I want to thank my classmates and friends in Trondheim for keeping spirits up through five fantastic years. Finally, and most of all, I want to thank my family for always having my back.

Contents

Acronyms

Adam adaptive moment estimation. 12

AI artificial intelligence. 3, 4, 8, 9

ANN artificial neural network. 10

AP average precision. 7, 8, 22

BCE binary cross-entropy. 13

BiFPN bidirectional feature pyramid network. 35, 36

BLI blue light imaging/blue laser imaging. 8, 61

CNN convolutional neural network. 9, 15, 16, 18, 23, 24

CRC colorectal cancer. iii, v, 1, 2, 3, 5, 7, 28

CV computer vision. 15, 20

CycleGAN cycle-consistent adversarial network. iii, v, 6, 9, 26, 27, 29, 34, 36, 37, 38, 39, 40, 45, 53, 59, 61

DL deep learning. 3, 6, 9, 10, 13, 14, 15, 20, 25, 33, 37, 59, 61

FCN fully convolutional network. 18

FFNN feedforward neural network. 10

FN false negative. 21, 22

FP false positive. 21, 22, 47

FPN feature pyramid network. 24, 35, 36

GAN generative adversarial network. iii, v, 9, 15, 25, 26, 56, 59, 61

HDI human development index. 1

IoU intersection over union. 22, 23, 24, 38, 40, 41, 42, 43, 47, 61

i-Scan i-Scan. 8

mAP mean average precision. 22, 23, 38, 40, 41, 42, 45

ML machine learning. 9, 10, 13, 14, 15

NBI narrow-band imaging. iii, v, 2, 4, 5, 6, 7, 8, 9, 27, 28, 29, 31, 34, 37, 39, 43, 44, 47, 48, 49, 50, 53, 54, 55, 59, 61

NICE NBI International Colorectal Endoscopic. 2, 4, 6, 28, 29

NMS non-maximum suppression. 24, 39

NN neural network. 10, 11, 13, 16, 17, 25, 33

OUS Oslo University Hospital. vii, 29, 49, 52

PICCOLO PICCOLO Widefield. 29, 30, 38, 39, 40, 49, 56, 57

PRC precision-recall curve. 22, 23

ReLU rectifier linear unit. 13, 14, 19, 20, 36, 37

SGD stochastic gradient descent. 12

SiLU sigmoid-weighted linear unit. 14, 20

SNBI synthetic narrow-band imaging. iii, v, 5, 6, 7, 34, 36, 37, 38, 39, 40, 41, 43, 44, 45, 47, 50, 51, 52, 53, 54, 59, 61

TP true positive. 21, 22, 47

WCE wireless capsule endoscopy. 3, 4, 5, 7, 61

WLI white-light imaging. iii, v, 2, 4, 5, 6, 7, 8, 27, 28, 29, 31, 34, 37, 39, 40, 41, 43, 44, 45, 47, 48, 49, 50, 51, 52, 53, 56, 59, 61

Chapter 1

Introduction

This chapter gives some background for the project, leading up to the problem description and a brief explanation of how the problem has been approached. Some related works are then presented, and finally, the outline of the thesis is given.

1.1 Background

Today, cancer is ranked as the primary cause of premature death in most of the western world [2]. As more countries continue to develop socioeconomically, it is expected that cancer will become the major barrier to a longer life in these as well. Since the outburst of the COVID-19 pandemic, there has been a dip in new cancer cases because less screening and treatment has been conducted globally. Because of this, an abnormally high number of developed cancer cases and deaths is expected in the next few years.

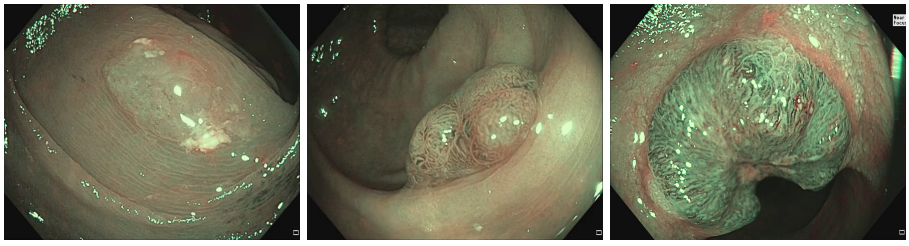
In 2020, colorectal cancer (CRC) accounted for 10% of all new cancer cases, ranking it third overall. With a death rate of 9.4%, it ranks number two, only surpassed by lung cancer. Incidence rates of CRC are increasing with increasing human development index (HDI). Norway has the highest incident rate in the world for women and ranks third overall (for both genders). The disease is therefore being linked to the western lifestyle, where lack of physical activity and high alcohol- and meat consumption are considered factors. In addition to advising people to adapt a healthier way of living, screening for CRC has proven successful. Therefore, national screening programs are being rolled out in more and more countries.

CRC usually develops from protrusions in the colon or rectum called polyps [3]. Polyps have different shapes, sizes, colors, etc. Some are not dangerous (benign), some are cancerous, and some may develop into cancer (adenomatous). Most symptoms are usually not present until the cancer has developed to an advanced and aggressive stage where it is too late for treatment [4]. Therefore, the best preventive action, and the primary goal of screening, is to detect the adenomatous polyps and remove them before they possibly develop cancer.

The most important procedure in CRC screening is called colonoscopy [5]. In a colonoscopy, a long soft hose is manually guided into the anus, through the rectum, and into the large intestine. At the tip of the hose is a video camera, as well as a white light source and a tool for polyp removal (resection). The process of capturing color images with a white light source is called white-light imaging (WLI). By looking at real-time color video from inside the bowel, the endoscopist can look for polyps, remove them, and send them to a laboratory for biopsy so that the polyp type can be confirmed and further treatment determined. Because colonoscopy is an expensive procedure, less invasive tests like fecal occult blood tests and sigmoidoscopy are initially performed [6]. If found necessary, a colonoscopy can be performed next.

Although all polyps usually are resected if detected, many are benign and do not need removal. This has led to development of supporting tools to aid the endoscopist in the difficult task of categorizing (classifying) the polyp while performing the colonoscopy. One of these tools is an image enhancement method called narrow-band imaging (NBI) [7]. In NBI equipment, the endoscopist can apply a filter on the white light source such that only specific wavelengths of green and blue light illuminate the colon [8]. This light is designed to highlight vascular patterns, which tend to be prominent in tumors and cancerous tissue. When performing a colonoscopy with equipment containing gear for NBI, the endoscopist can switch to this modality when a polyp is detected and more easily determine what type it is and whether resection is necessary.

NBI eases the identification of polyp features based on their superficial colors and vascular patterns. From these features, different classification systems for polyps have been developed. A simple but frequently used system is the NBI International Colorectal Endoscopic (NICE) system, which divides the polyps into three types/classes, shown in Figure 1.1.



(a) **(b)** **(c)**

Figure 1.1: a) Type 1: Hyperplasia, **b)** Type 2: Adenoma, **c)** Type 3: Adenocarcinoma (deep submucosal invasive cancer). Images are borrowed from the PICCOLO set [9].

In NICE, polyps are divided into three categories [10]:

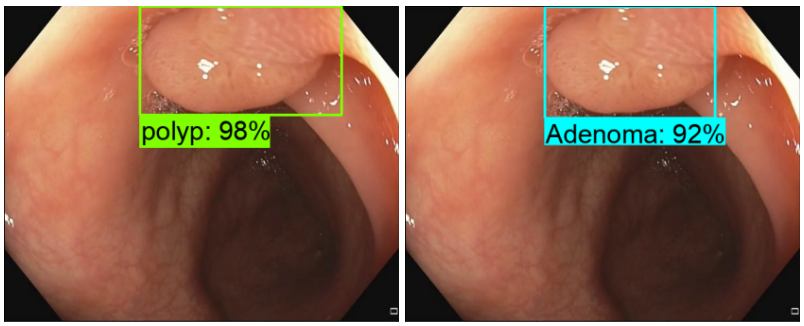
- Type 1: Hyperplasia, benign and do not need resection unless they are larger than 6 mm. [1]

[1]Type 1 also comprises sessile serrated polyps, but this project does not differentiate between

- Type 2: Adenoma, not dangerous at the moment, but may develop cancer. In general, all polyps suspected of being adenomas should be resected.
- Type 3: Deep submucosal invasive cancer. Removal is necessary, and resection by surgery may be considered.

If type 3 is detected, the prognosis is typically very poor because the cancer has developed too far or spread. Therefore detection and removal of type 2 polyps imposes a huge preventive effect and is the main reason for the successfulness of CRC screening.

In recent years, artificial intelligence (AI) using deep learning (DL) has grown to become a giant field and proven useful on many different problems [11]. Among these are medical diagnosing and automatic object detection; two useful topics in CRC screening. Automating polyp detection and classification has therefore grown to be a significant field of research. Initially, these methods are thought to serve as a tool for easing the endoscopist's job during colonoscopy [11]. Figure 1.2 shows examples of detection models for polyps.



(a) One-class polyp detection. **(b)** Two-class polyp detection.

Figure 1.2: From **(a)** a one-class polyp detection model and **b)** a two-class detection model that discerns hyperplastic polyps from adenomas. The confidence score is given in percent.

A well-functioning detection model for polyps can draw what is called a *bounding box* around the polyp. The model can be implemented as one of the following:

- A one-class detector (Figure 1.2a): Detects the class *polyp* with a given confidence (confidence score).
- A multi-class detector (Figure 1.2b): The model can detect polyps of different types/classes with a given confidence.

Implementing such a model in the colonoscopy equipment is thought to aid the endoscopist in both detecting and classifying polyps in real-time while examining the patient.

Although being the state-of-the-art technique for CRC screening, colonoscopy is, in addition to being expensive, an uncomfortable procedure for the patient [3]. This has led to the recent development of wireless capsule endoscopy (WCE),

these and hyperplastic ones.

popularly known as pill cameras. WCE was initially developed for endoscopy of the small bowel because endoscopic equipment could not reach this far inside the body [12]. It is a non-invasive method where a pill camera is swallowed and captures video of the digestive system, see Figure 1.3[2].

Figure 1.3: Medtronic PillCam™ SB 3 Capsule for endoscopy of the small bowel, capable of capturing up to 6 frames per second.

Because of its expensiveness and patient discomfort, WCE has also been developed for colorectal use [13]. Ongoing research tries, among other things, to improve the image quality and frame rate of these capsules, as well as implement wireless real-time automatic polyp detection.

1.2 Motivation

Several meta-analyses show that manual polyp detection (one-class) does not improve with NBI versus WLI [6]. Others show the opposite [8]. This may be because NBI has developed, and analyses of the first generation gear show worse performance than second- and third generation equipment. Despite this, good results are, for both modalities, dependent on properly cleansing of the bowel in advance of the colonoscopy [14, 15]. This is, however, often not the case. Bowel preparation is a complicated procedure and may be a problem if not being conducted properly, no matter what modality is used [5]. Moreover, it is in practice impossible to make a perfect comparison of NBI and WLI because a polyp image cannot be captured with both modalities simultaneously from the exact same position.

Today, NBI is, in practice, mainly used for the classification of polyps. The endoscopist switches from WLI to NBI when a polyp is detected to evaluate the need for resection. The NICE standard is a relatively easy and widely used classification system for doing class evaluation when performing colonoscopy with NBI. NBI gear is, however, not available everywhere, and most colonoscopies are performed with WLI only.

AI-based polyp detection systems are additional tools that can be helpful for endoscopists while performing a colonoscopy, easing their challenging and important job of detecting and classifying the polyps. In this case, the doctor would still benefit from having the NBI modality available for manual classification. Whether

[2]Image from: *https://www.medtronic.com/covidien/en-us/products/capsule-endoscopy/pillcam-sb-3-system.html* (6.6.22).

an automatic polyp detector would be more easily able to detect and classify polyps captured with NBI is not clearly answered. This is probably because of the already mentioned nature of capturing colonoscopic content, as well as the lack of public datasets.

WCE has the potential of being the initial CRC screening procedure such that manual expertise could be focused on follow-up of serious cases. To avoid that doctors have to look through 7 hours of gastrointestinal recordings and manually detect polyps, an automatic detection model would be highly beneficial for this use [15]. If NBI was shown to improve automatic polyp detection and classification, having this modality available in WCE would be beneficial. Capsules do, however, have minimal space for hardware, and, although methods have been proposed, no commercial capsules today have an option for NBI [16].

1.3 Problem Description

Several of the challenges mentioned above can be solved if one could create NBI artificially from real WLI videos. There exist different post-processing techniques for enhancing the WLI images, but no known method that tries to mimic NBI, that is, to create synthetic narrow-band imaging (SNBI) (see Figure 1.4).

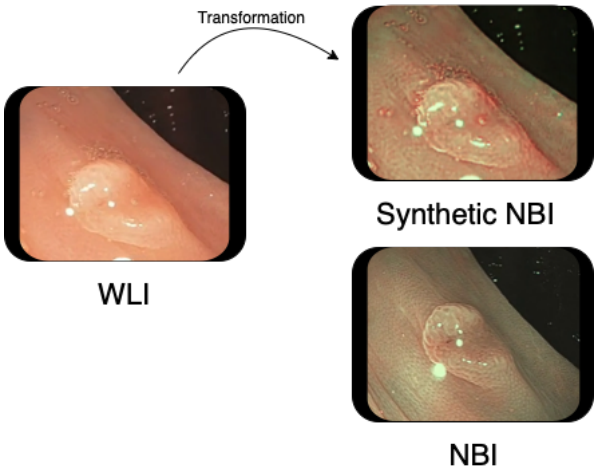


Figure 1.4: A polyp captured with WLI, transformed to SNBI that resembles real NBI. The problem is to find such a transformation.

1.4 Objectives

The objective of this thesis is to investigate and try to solve the problem of creating SNBI from WLI when NBI is not available for acquisition, as well as evaluate its usefulness and resemblance to real NBI. More specifically, the following questions will be answered:

1. **Is automatic polyp detection and classification easier on NBI compared to WLI?**
   Because of its properties, NBI is used for manual classification of polyp type when NBI equipment is available. One can therefore hypothesize that classification, and maybe also detection, may improve on an automatic detector as well.
   Polyp images and videos where each polyp has been captured with both NBI and WLI will be gathered. These will be used to make two as identical datasets as possible; one with WLI and one with NBI. Each of these sets will be used to train, evaluate and test the same state-of-the-art object detector, both for one-class (polyp) and two-class (adenoma and hyperplasia) detection. Although these datasets will not be exactly similar, one will get a good indication of how NBI performs in automatic detection in contrast to WLI by comparing their test results.

2. **Can SNBI be created by post-processing WLI? How?**
   To answer this question, a way to transform polyp images from WLI to NBI will be needed. This can be regarded as an unpaired image-to-image translation problem. In the autumn of 2021, a DL method called cycle-consistent adversarial network (CycleGAN) was proposed for creating SNBI from WLI [1]. In this thesis, the CycleGAN method will be investigated further. By testing the SNBI on the NBI detection model from **1.**, an indication of how well it resembles real NBI can be obtained.

3. **Can a modality transformation from WLI to SNBI be made such that automatic polyp detection and classification improves?**
   This question depends on whether **1.** is true, i.e., one can only expect SNBI detection and classification to be as good as NBI. Similar to the models mentioned in **1.**, one-class and two-class detection models can be trained on SNBI data created from WLI. By comparing their test performance to the WLI detection models, one can know for sure whether the transformed images improve detection and classification. If this is the case, one can also conclude that NBI is easier to detect or/and classify than WLI.

4. **Can the proposed WLI-to-SNBI transformation be used in real-time colonoscopy?**
   The inference time of the modality transformation by the proposed CycleGAN implementation will be measured to answer this question.

To evaluate the questions above, special image datasets of polyps captured with both NBI and WLI are needed. For the detection part, these images also need to be clinically annotated with polyp localization and NICE class.

1.5 Contribution

This year (2022), a national screening program for CRC is being rolled out in Norway[3]. Initially, a manual colonoscopy will not be offered to everyone. Because of its massive costs of manual expertise, time, and money, one can question whether it ever will be. WCE is regarded as a promising solution to this problem and is currently under development. Automatic polyp detection is key to making WCE screening efficient and truly easing the workload for the healthcare system. Because capsules capture content with WLI, a post-process transformation that creates SNBI may be the easiest solution to enable NBI for WCE. This might be useful for manual inspection of the captured content, but it is especially interesting if the transformed video also improves automatic detection and classification.

Skilled clinicians will, however, still be needed for the further colorectal examination of the potentially serious cases found by WCE screening. Having SNBI as an optional modality in real-time manual detection would be helpful for the endoscopist. Instead of having special NBI gear, he or she can switch to SNBI purely with software. He or she will also have the option to look at both modalities simultaneously.

1.6 Related Works

Below are some related works presented, considered relevant for this thesis.

1.6.1 Frame-based polyp multi-class detection

In sequence-based detection models, predictions from previous frames are used when predicting the object class in the current frame. Frame-based detection, on the other hand, does only base its prediction on the current frame. In this thesis, frame-based detection has been conducted.

Compared to one-class detection and segmentation of colorectal polyps, less research has been conducted for multi-class detection [17]. This is probably partially because of the lack of datasets with class label annotations. Another problem is that there are many ways to classify polyps, making a comparison of different studies difficult. Despite this, most studies focus on discerning adenomas in the test data, e.g., adenomas vs. non-adenomas or adenomas vs. hyperplastic polyps.

One study did both one-class (polyp) and two-class (adenoma vs. hyperplasia) frame-based detection on different detection models [15]. Their results were based on the F1 score and average precision (AP) of both classes and showed that adenomas on all models were easier to detect than hyperplastic polyps. They argued that this is as expected because adenomatous polyps usually are larger, with vascular textures that make them more visible than hyperplastic ones. A similar

[3] *https://kreftforeningen.no/forebygging/screening-og-masseundersokelser/tarmscreeningprogrammet/* (6.6.22).

study that achieved very good results tried detecting and classifying adenomas and non-adenomas[4] [18]. This study also got better AP results for the adenomas.

Two other studies worth mentioning did a five-class polyp detection. In the first one, the most notable findings were indications that using NBI improved classification [19]. They could, however, not confirm it. The second one showed that one-stage detectors could compete with two-stage detectors for polyp detection [20]. They also showed that the prediction errors in their models were usually because of wrong detections, not classification.

1.6.2 Colonoscopy image enhancement methods

In addition to three generations of NBI imaging equipment, there exist different techniques for enhancing patterns that ease the detection and classification of anomalies in endoscopic images [14]. Often, NBI equipment have an option for magnification as well [21]. This improves manual detection and classification performance in general. One can also inject liquid substances to further enhance the superficial structures of the colon. By, for instance, using acetic acid in addition to NBI with magnification, manual diagnosis accuracy can be improved even more. Liquid pigments can also be injected into the colon to colorize its surface and improve visualization. This technique is called chromoendoscopy [8].

Similar to NBI are some methods called blue light imaging/blue laser imaging (BLI) [22]. BLI uses other wavelengths than NBI and special blue light sources instead of filtering the white light.

For post-process enhancement of WLI, different tools are in use. These are not based on AI but on more traditional signal processing techniques. One is called i-Scan (i-Scan) and is integrated with colonoscopy equipment from PENTAX [23]. In i-Scan, the image is divided into its RGB components. Different mapping functions are applied to magnify or suppress the colors to enhance the contrasts in the image. FICE is another enhancement technique where specific combinations of wavelengths (colors) are enhanced [24]. FICE has shown competitive results to NBI, while i-Scan reportedly improves the detection of adenomas.

1.7 Outline

Chapter 2 provides the theoretical background for the methods used in this thesis. This is followed by Chapter 3, where a presentation is given of the datasets that have been used. Methodology and implementation are given in Chapter 4. A selection of the numeric results is given in Chapter 5, and a thorough discussion of them is provided in Chapter 6. Images from the experiments are included in this chapter to substantiate the discussion. Chapter 7 gives a conclusion of the thesis, and some suggestions for future work are written in Chapter 8. After the bibliography, Appendix A is included, containing tables with all numerical results.

[4]The study differentiates between adenomas and polyps, but one can interpret this as adenomas and non-adenomas.

Chapter 2

Theory

This chapter contains a thorough description of the theoretical foundation for the chosen methods. After reviewing DL and convolutional neural networks (CNNs), the theory behind automatic object detection models and the metrics used to evaluate them are given. Following this is an introduction to generative adversarial network (GAN) and CycleGAN. Finally, the idea and physics behind NBI are explained.

2.1 Machine Learning

AI is a term used for the techniques that let computers mimic the behavior of humans [25]. Today, AI comprises techniques that let computers perform specific tasks, often exceeding human capabilities in terms of efficiency and accuracy. Machine learning (ML) is a branch of AI techniques where computers improve their performance with experience. This means that as the computer is given more data, it will get more experienced and *learn* the patterns in the data better. There are three main types of ML:

- *Supervised learning*: One wants to find the relationship between two domains X and Y, that is, a mapping $f : X \to Y$. The model is then trained on *labeled* data, $(X_{train}, Y_{train}) = (x_1, y_1), ..., (x_n, y_n)$. In other words, for each input x_i, the ML model is told what the desired output y_i should be. After being trained on enough train data, the goal is that the model now can map new unseen data X_{test} to reasonable predictions in Y. In this thesis, supervised learning techniques have been used.
- *Unsupervised learning*: These methods are typically used when labeled data is unavailable, i.e., only X_{train}. One wants to find patterns in this data, such as how it can be grouped or sorted.
- *Reinforcement learning*: Here the model is given only a set of rules, a current state, and a goal. Using a reward system, the model learns what actions to perform from trial and error. This method is famously used in chess-playing AIs.

As already mentioned, when training and testing an ML model, one must keep the training and test data separate. This is important because when testing a model, it should never have seen this exact data before. In addition, it is common to use a validation set. This set is used to test the model during training or compare different models. One can use the validation set to choose the best model for testing or the best model parameters. Often, one only has one set of data and therefore splits it into train (e.g., 70-80%), validation (10-15%), and test (10-15%) before using it in an ML problem.

2.2 Deep Learning

Although being very successful in many tasks, a limitation of conventional ML techniques is their ability to make use of the raw data directly [26]. Usually, one has to feed these models with carefully extracted raw data features, which in many cases can be very difficult. The methods that try to overcome this problem are called representation learning methods. Here the model itself finds abstract features in the data that enable it to interpret it correctly.

Deep learning (DL) comprises representative learning methods that extract complex patterns in the input by using multiple layers of abstraction called hidden layers. These models are called artificial neural networks (ANNs), here also called neural networks (NNs), because of their resemblance to neuron patterns in the brain [27]. A typical NN is shown in Figure 2.1.

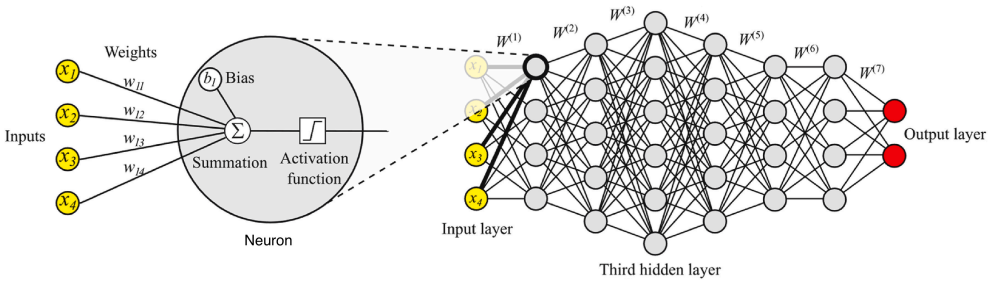


Figure 2.1: Right: A multilayer perceptron. Left: a perceptron/artificial neuron. Borrowed and edited from [27].

A NN is usually composed of an input layer containing the raw data, followed by a series of hidden layers before ending in an output layer. If the network has more than one hidden layer, it is said to be *deep*. The network will learn simple patterns in the first hidden layer and then combine these into the next to learn more complex features. The output layer contains an abstract representation of the input data that enables it to be, e.g., classified. A NN that has information flow from the input to the output and does not contain any cycles is called a feedforward neural network (FFNN) [26].

Each layer is made up of a number of artificial neurons [27]. The neurons weigh the inputs from the neurons in the previous layer and produce an output

that serves as input for the neurons in the next layer. The weighted sum of a neuron's inputs can be written as $b + \Sigma_i w_i \cdot x_i$, where w_i is the *weigth* of each input x_i and b is an offset value called *bias*. In the vectorized form, the output a^l of a layer l can be written as in (2.1).

$$a^l = \sigma^l(W^l a^{l-1} + b^l) \tag{2.1}$$

Here a^{l-1} denotes the outputs of the previous layer, W^l and b^l are the weights and biases of the current layer, and σ is the layer's activation function. When training a NN, i.e., when the NN *learns*, it is essentially just updating its weights and biases.

An *iteration* is when a training sample, or a batch of samples, is run through the network. Its output is used to update the network parameters. To train the model for one *epoch* means to iterate through the complete training set one time. One usually trains a model for tens to hundreds of epochs. One can also train a model for a number of *steps*. For one epoch, the number (#) of steps can be calculated as:

$$\#\ of\ steps = \frac{\#\ of\ samples}{batch\ size}$$

2.2.1 Backpropagation and optimizers

An update of weights and biases happens after each iteration of training [28]. This is generally done using backpropagation, a technique based on the difference between the desired network output vector $y(x)$ and, given a network of L layers, the predicted output $a^L(x)$. A loss function $\mathcal{L}$ that measures this difference can be defined, e.g., the quadratic loss function given in (2.2).

$$\mathcal{L}_{L2} = \frac{1}{2}\sum_x \|y(x) - a^L(x)\|^2 \tag{2.2}$$

This loss is also called the L2-loss. $\mathcal{L}$ is the loss of the network for a given training sample. Because of (2.1), and the fact that y is a fixed parameter, $\mathcal{L}$ is a function of weights (and biases). The goal is to minimize the difference between predicted and desired output; thus, one needs to find the weights that minimize $\mathcal{L}$. In backpropagation, the gradient of $\mathcal{L}$, $\nabla\mathcal{L}$, is a function of the partial derivatives $\frac{\partial\mathcal{L}}{\partial w}$ and $\frac{\partial\mathcal{L}}{\partial b}$, which means that $\nabla\mathcal{L}$ shows how $\mathcal{L}$ changes with w and b. To minimize $\mathcal{L}$, the updated values of w and b are calculated by gradient descent, a technique based on (2.3) and (2.4).

$$w_k \rightarrow w_k' = w_k - \eta\frac{\partial\mathcal{L}}{\partial w_k} \tag{2.3}$$

$$b_l \rightarrow b_l' = b_l - \eta\frac{\partial\mathcal{L}}{\partial b_l} \tag{2.4}$$

Here η denotes the *learning rate* of the model. The idea of gradient descent optimization is illustrated in Figure 2.2.

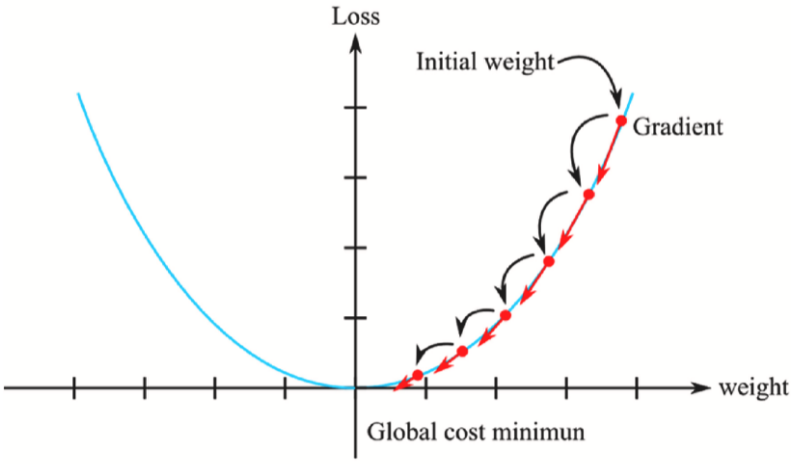


Figure 2.2: Gradient descent optimization [27]. Given some initial weight(s), the gradient of the loss function is used to find the weight(s) that minimize(s) the function, also called the global cost minimum.

Because $\mathcal{L}$ is a function of the network output, the weights are updated by propagating backward through the network. In practice, there are different techniques for minimizing $\mathcal{L}$. These are called optimizers. In one called stochastic gradient descent (SGD), a random training sample is chosen every iteration, and the weights are updated. One can also select the complete set of training samples (batch), or a part of it (mini-batch), and use the averaged $\nabla\mathcal{L}$ for gradient descent, called batch gradient descent and mini-batch stochastic gradient descent, respectively [28]. Another more recent optimizer is the adaptive moment estimation (Adam) optimizer [29]. Adam is based on SGD but introduces a per-parameter learning rate based on the initial learning rate and the square of the gradient. It is very efficient and widely used.

As shown in (2.3) and (2.4), the learning rate η is a parameter that decides how large the parameter updating shall be. Learning rates can be kept constant during all training or scheduled, which means that it changes with a function at the different epochs [30, 31].

2.2.2 Loss functions

When a training sample has been run through the model, a loss function is used to quantify the difference between the predicted and desired output [27]. In addition to the $\mathcal{L}_{L2}$ in (2.2), there are different loss functions in use. Because the choice of loss function will affect the efficiency and accuracy of the model during training, the different functions have different uses.

Binary cross-entropy (BCE) loss is often used in binary problems, for instance, image segmentation. This function calculates the entropy of the two classes, e.g., pixels in the region of interest vs. background, and measures loss based on their difference. BCE is given in (2.5).

$$\mathcal{L}_{BCE}(p_t) = -log(p_t), \quad \text{where} \quad p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise.} \end{cases} \tag{2.5}$$

Here p_t is the class prediction. This function is very efficient but suffers in problems with class imbalance. This was solved by introducing weighted BCE, where a weighting factor α_t compensates for the unbalanced data. α_t is defined similarly to p_t and can be calculated from inverse class frequency [32].

Focal loss was introduced for object detection and made huge improvements on one-stage detection models. It is a weighted BCE loss with another weighting term that automatically adjusts the weight factor based on the prediction confidence (confidence score, see Section 2.4). In this way, the most uncertain objects detected will be weighted as more important. The focal loss is defined in (2.6).

$$\mathcal{L}_{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma} log(p_t) \tag{2.6}$$

Here γ is a fixed parameter.

2.2.3 Activation functions

So far, the neurons in a NN have been described by linear functions. In many DL problems, however, the network needs to learn non-linear patterns [33]. It is therefore beneficial to introduce non-linearity to the network, so that such patterns can be learned. Another important thing to control is that when updating the weights and bias of a neuron, the output should not change too much [28]. An activation function is applied to the weighted input of the neuron to control these factors. Some activation functions are illustrated in Figure 2.3.

The Sigmoid function is a function used in ML and DL, nowadays mainly in the output layer of a NN, for probability predictions. This is because it maps the input to a value between 0 and 1. Therefore it can be viewed as a smooth unit-step function. The Sigmoid function is shown in (2.7).

$$\sigma_{Sig}(x) = \frac{1}{1+e^{-x}} \tag{2.7}$$

While the Sigmoid function often is used in binary classification tasks, a similar function called Softmax is used in multi-class problems. Another activation function is called the ReLU function, given in (2.8).

$$\sigma_{ReLU}(x) = max(0, x) \tag{2.8}$$

The ReLU function sets all negative numbers to zero while leaving the positive ones unchanged. Because it contains a lot of linear properties and is very

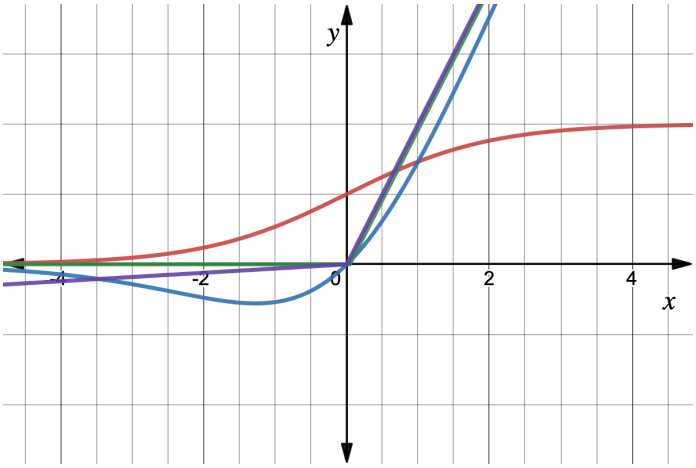


Figure 2.3: Different activation functions that are popular today. Sigmoid in red, sigmoid-weighted linear unit (SiLU) in blue, rectifier linear unit (ReLU) in green, and leaky ReLU in purple.

computationally efficient, the ReLU function is used a lot in DL. Zeroing all non-positive values leads, however, in some cases, to what is called "dead neuron issues". Therefore, a variant called leaky ReLU was suggested, adding a slight slope for the negative values.

The SiLU function is a variant of the Sigmoid function, given in (2.9).

$$\sigma_{SiLU}(x) = x \cdot \sigma_{Sig}(x) = \frac{x}{1+e^{-x}} \tag{2.9}$$

Here $\sigma(x)$ is the Sigmoid function of x. The SiLU function was developed for reinforcement learning but has recently proven useful in other applications as well.

2.2.4 Overfitting and underfitting

Overfitting and underfitting are terms related to the training of an ML model, as illustrated in Figure 2.4.

Underfitting is the problem when the model performs bad on both the training and the validation set [25]. This can occur for different reasons, for instance, the model is too simple for the data. Another reason might be that the model simply is not trained enough. Overfitting, on the other hand, is the problem when the model works well on the training data but not on the validation data. This indicates that the model has learned the specific training data too well. There are different ways to cope with overfitting:

- Reduce the number of training iterations/epochs/steps, i.e., stop the training when the validation loss starts to rise.
- Obtain more training data. More training data will give a better model.

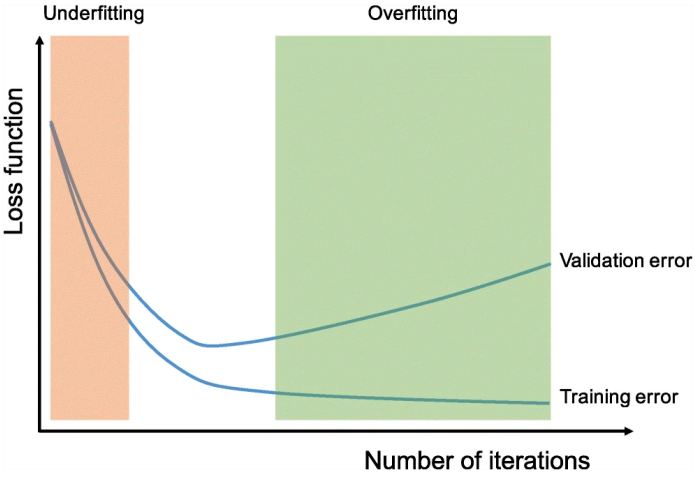


Figure 2.4: Under- and overfitting [25].

- Use augmentation on the available training data. This can be done in different ways. Traditional methods in image augmentation involve randomly flipping, rotating, skewing, zooming, etc., of the input images, as well as changing colors, contrast, and adding noise. More recently, GANs has been used to create more new images artificially. The augmentation can be done during or before training. In the latter, it becomes a part of the data preprocessing for enlarging the training set before training.
- Apply different regularization techniques. These are techniques that make the model better at generalizing. Dropout is a regularization technique where the activation of a few random nodes in the model outputs zero for each iteration. This makes the model more robust.
- Use batch normalization, a technique that normalizes the input data in each layer, evidently improving the ML models and avoiding overfitting.
- Apply transfer learning to the model [34]. In transfer learning, one initializes the model with weights from a pre-trained model. The specific training is, therefore, only a fine-tuning of the model to make it work for a specific task. Usually the pre-trained weights are acquired from an extensive training on huge datasets like ImageNet [30].

2.3 Convolutional Neural Networks

The term computer vision (CV) is used for the techniques for automatic detection and classification of objects in images [34]. In recent years, convolutional neural networks (CNNs) has been the standard approach in DL-based CV problems [35]. A CNN extracts features from the input on different abstraction layers and combines these to learn patterns in its content. The main building blocks/layers of a CNN are illustrated in Figure 2.5.

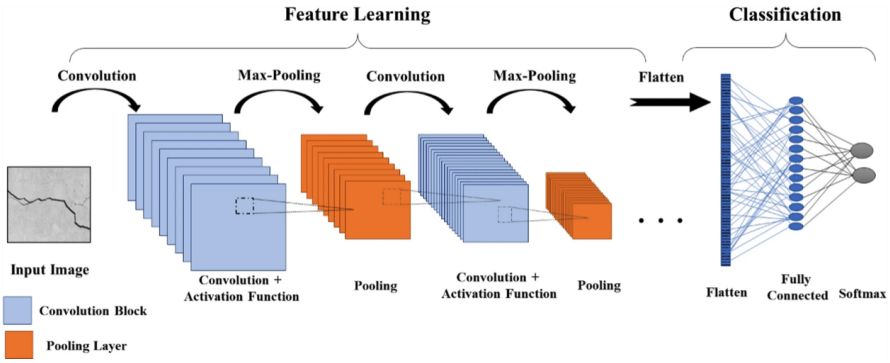


Figure 2.5: The main layers of a CNN [25].

2.3.1 The main layers

The input of a CNN is represented as an array of numbers. When processing color images, the input will be an array of three dimensions, for instance, 512x512x3 (width, height, RGB). A CNN consists of three main layers/building blocks [25].

Convolutional layers

A convolutional layer is showed in Figure 2.6.

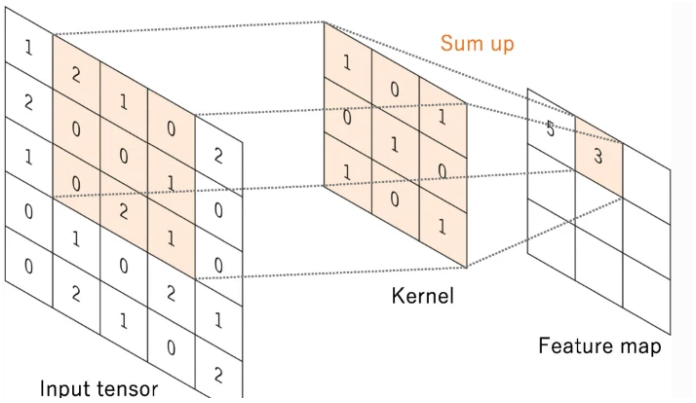


Figure 2.6: A convolutional layer [25]. The kernel is a weighting matrix that is convolved with the input tensor, producing a feature map.

In the convolutional layer, a *kxk* array called a kernel/filter is slid across the input array (tensor), calculating the elementwise product of each *kxk* array that exists in the input. The values of the kernel thus work as weights on the input. The kernel output is a new array called a feature map. Because the same weights are applied for creating one feature map from the whole image, this is a much more efficient method than regular NNs. In practice, several kernels are applied in each layer, producing different feature maps. In the beginning, these feature maps will contain quite simple information about the image, e.g., horizontal and

vertical edges. By combining them in deeper layers, more complex feature maps can, however, be created. The network learns and adjusts the weights of each kernel during training. As with the regular NNs, non-linearity is used by applying a non-linear activation function after the convolution.

A convolutional layer that, e.g., uses kernels with size 3x3, can be written as a Conv3x3 layer. A Convkxk on an image with dimensions *w* and *h* will give feature maps of dimensions (w-k+1) and (h-k+1). To preserve the input image size, one can use a technique called padding, where values are added around the input image, as illustrated in Figure 2.7.

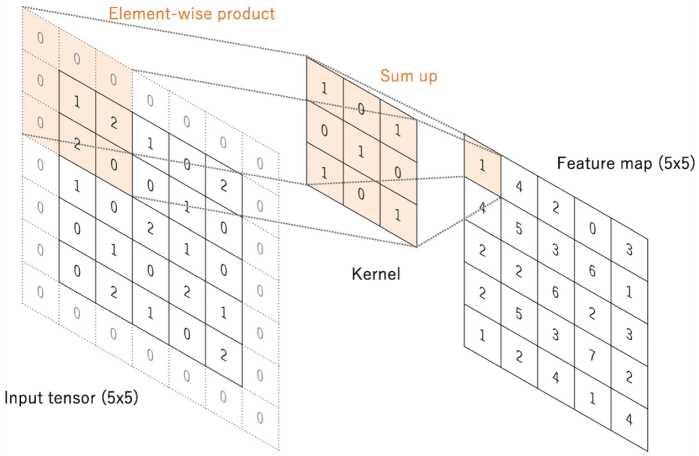


Figure 2.7: Zero padding [25].

Pooling layers

The function of the pooling layers is to down-sample the feature maps, as illustrated in Figure 2.8.

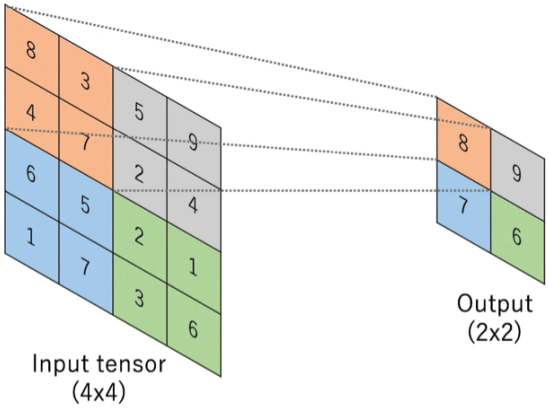


Figure 2.8: Max pooling with stride 2 [25].

There are no trainable parameters in the pooling layers. The filters that are applied are often 2x2, which means that they reduce each 2x2 part of the input into one value. Often is a stride of 2 also used. This means that the filter "jumps" 2 places on the input before being applied again. In a max-pooling layer, the maximum in each sub-array is sent to the down-sampled feature map, while in an average-pooling layer, the average of them are forwarded.

Fully connected layers

While convolutional layers and pooling layers alternate throughout a CNN, a fully connected layer can appear at the network's end. Before the fully connected layer, the feature maps are typically flattened into a one-dimensional array called a feature vector. This is passed to one or more fully connected layers that finally produce an understandable output, e.g., a class prediction or all class probabilities. The fully connected layers are followed by an activation function. In the final layer, this is often a Sigmoid or Softmax function.

2.3.2 Encoders, decoders, and fully convolutional networks

The CNN architecture described above is usually what is called an *encoder* [26]. The encoder consists of repeated blocks of alternating convolutional and pooling layers, which finally are flattened out to a feature vector.

In many cases, e.g., image segmentation or image transformation, the desired output is also an image. Using fully connected layers on the feature vector is therefore not helpful. The feature vector can instead be given to a *decoder*, which somehow does the opposite of the encoder. The decoder also consists of convolutional and pooling layers and can interpret the feature vector from the encoder to produce a new image. In an encoder-decoder network, the flattening can be omitted, leaving the complete architecture consisting only of convolutional and pooling layers. Such networks are called fully convolutional networks (FCNs). Because fully connected networks are computational heavy, FCNs can also be used in, e.g., classification tasks. A configuration of Conv1x1 blocks then replaces the fully connected layer(s).

2.3.3 From residual blocks to MBConv blocks

Residual blocks were introduced as a solution that could improve learning in deep CNNs, as these tend to suffer from what is called the degradation problem [36]. A residual block is illustrated in Figure 2.9.

In short, the residual block uses *shortcut connections* to add shallow features $\mathbf{x}$ to a deeper layer. In this way, the deeper features $\mathcal{F}(\mathbf{x})$ will not produce a higher error than the shallow part $\mathbf{x}$ does. Residual blocks are the foundation of ResNets, which is a popular group of CNNs.

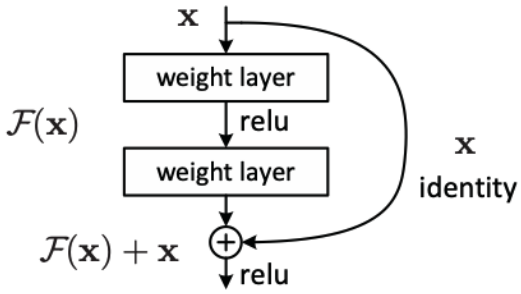


Figure 2.9: A residual block [36].

To increase the efficiency of the residual blocks, "bottleneck blocks" were introduced [36]. A bottleneck block versus a regular residual block is illustrated in Figure 2.10.

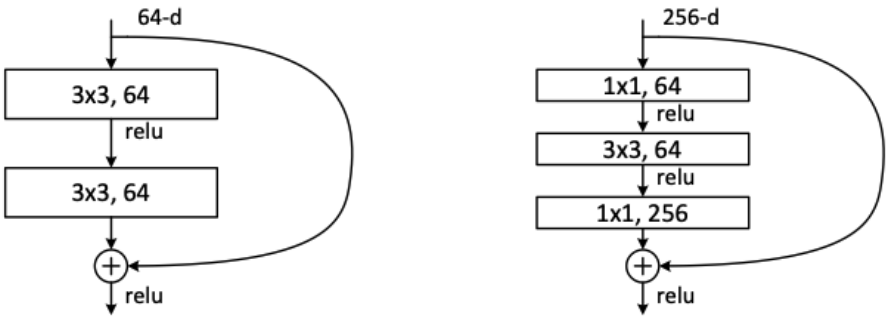


Figure 2.10: Left: A residual block. Right: A residual bottleneck block [36].

Bottleneck blocks reduce the number of feature maps (channels), i.e., the width, by applying a Conv1x1. Then they apply a Conv3x3, followed by another Conv1x1 layer to re-scale the number of feature maps.

With the introduction of MobileNetV2 in 2019, inverted residual blocks with linear bottleneck layers were introduced [37]. The inverted residual blocks upscale (instead of down-scale) the number of channels before performing the Conv3x3, and then down-scale back to the number of initial channels. "Linear bottleneck layers" means removing the non-linear function (ReLU) at the end of the block. To reduce the number of convolutional parameters in the MobileNetV2, depth-wise separable convolution was also introduced. Instead of using a regular Conv3x3 across all channels, a Conv3x3 is applied to each channel independently, before a Conv1x1 is used on all channels. Such inverted residual blocks, with linear bottleneck layers, depth-wise separable convolution, and also adding normalization to all layers, are more commonly known as MBConv blocks. A MBConv block is illustrated in Figure 2.11.

Here H, W, and C are the height, width, and number of channels, respectively.

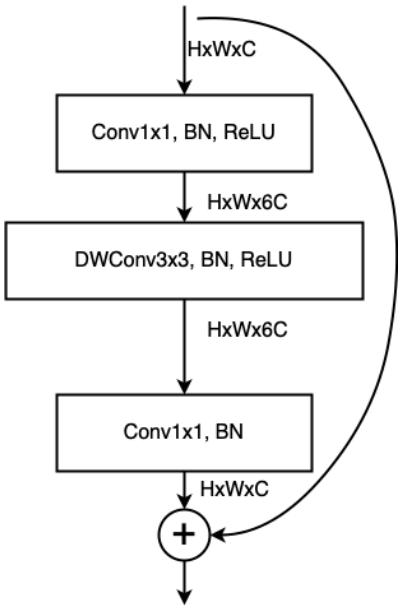


Figure 2.11: The original MBConv block.

With the introduction of EfficientNet in 2020, a modification of the MBConv block was introduced [38]. Squeeze-and-excitation optimization was added so that the importance of each channel could be learned, as well as stochastic depth, which is a kind of dropout technique on layer level. Finally, the ReLU activation function was replaced by a SiLU function.

2.4 Object Detection

In CV, object detection is the task of detecting objects of a predefined class in digital images [34]. Today, the standard approach when doing object detection is by using DL. It differs from image segmentation, where each pixel is classified as to whether it contains the desired object or not, resulting in a *mask* of the same size as the input image. In object detection, a bounding box is drawn around the object, and the certainty of box localization and object class is predicted with a *confidence score* between 0 and 1, as illustrated in Figure 2.12.

2.4.1 Metrics

In object detection, one can define the ground-truth bounding boxes as positives and all other possible boxes as negatives, which gives rise to the following terms [40]:

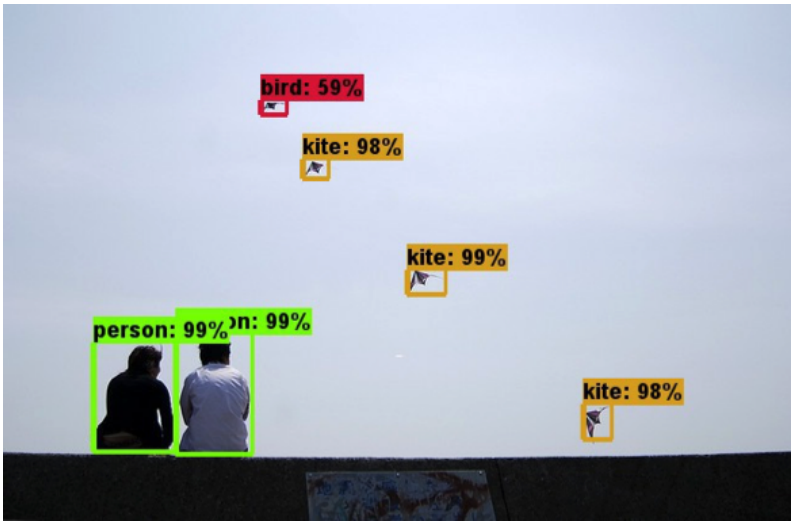


Figure 2.12: Example of object detection with confidence scores [39].

- True positive (TP): The prediction matches with the ground-truth box.
- False positive (FP): The prediction does not match with the ground truth, meaning that the prediction either contains a non-existing object or is being misplaced.
- False negative (FN): There is no prediction, i.e., there is a ground truth that was not detected.

The total TPs and FPs are used to calculate how many positive predictions were correct. This metric is called precision and is defined as in (2.10).

$$Pr = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} = \frac{\sum_i TP_i}{\text{all detections}} \tag{2.10}$$

Another metric, recall, is a measurement of how many ground-truth boxes were predicted, defined in (2.11).

$$Rc = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} = \frac{\sum_i TP_i}{\text{all ground-truths}} \tag{2.11}$$

As mentioned earlier, an object detector gives a confidence score for each box prediction. One can now define a score threshold τ that discards predictions below this threshold. Thus TP, FP, and FN, and therefore also precision and recall, can be regarded as functions of τ. Precision and recall can now be redefined as in (2.12) and (2.13).

$$Pr(\tau) = \frac{\sum_i TP_i(\tau)}{\text{all detections}(\tau)} \tag{2.12}$$

$$Rc(\tau) = \frac{\sum_i TP_i(\tau)}{\text{all ground-truths}} \tag{2.13}$$

Note that the denominator in (2.13) is not affected by different τs. If τ is lowered, more predictions will be made. This may increase the FPs but reduce the FNs, hence decrease precision and increase recall. For different confidence thresholds, one, therefore, gets different precision-recall pairs which can be plotted against each other in what is called a precision-recall curve (PRC), illustrated in Figure 2.13[1].

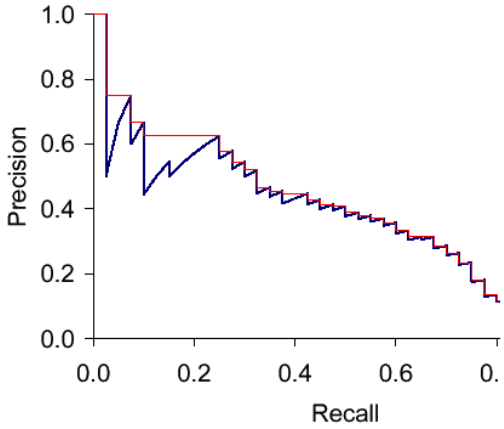


Figure 2.13: The blue curve shows the PRC. PR-AUC is the area under this curve. AP is calculated as the area under the red curve, plotted for each class independently.

The area under this curve is called the PR-AUC. If the PRC of one class is quantized by a step function, the area under the new curve can be calculated as the average precision (AP). By averaging the AP for all classes the mean average precision (mAP) is calculated, defined in (2.14) where C is the number of classes.

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i \tag{2.14}$$

Note that if only one class is to be detected, mAP = AP.

When dealing with detection boxes, the amount of overlap between the boxes also determines whether a detection is a TP or not. To measure this overlap, a metric called intersection over union (IoU) is used, defined as in (2.15) and illustrated in Figure 2.14 [40].

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \tag{2.15}$$

IoU can vary between 0 and 1, where a higher value means a better overlap between the prediction and the ground-truth box. By calculating the IoU between a

[1]Figure borrowed from *https://deshanadesai.github.io/notes/Evaluation-of-Results-using-Mean-Avg-Precision* (6.6.22).

$$\text{IOU} = \frac{\text{area of overlap}}{\text{area of union}} =$$

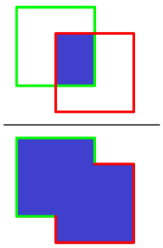


Figure 2.14: IoU, given B_p = red and B_{gt} = green [40].

prediction B_p and ground-truth box B_{gt}, one can compare this to an IoU threshold (0.5 is typically used) to determine whether the prediction is true or false.

mAP and the PRC is calculated for a specific IoU threshold. For instance, at 0.5, the mAP@.5 is calculated. An important metric, used when evaluating detection model performance on the COCO dataset, is called mAP@[.5:.05:.95] [41]. What this does is averaging the mAPs with IoU thresholds ranging from 0.5 to 0.95 and a step size of 0.05. In this thesis, this metric is named mAP for simplicity.

2.4.2 Two-stage vs. one-stage detectors

The first CNN-based object detection networks were what are called two-stage detectors [34]. Here, the model first proposes a set of object candidate boxes, whose contents are given to a CNN model which extracts features from these regions. These features are secondly classified to determine if this box contains an object and what object it is.

In a one-stage detector, there is no candidate box proposal followed by a verification. The classification and bounding-box generation is done in parallel, drastically decreasing the computational time. Despite one-stage detectors' efficiency, the two-stage detectors used to outperform them in means of accuracy. This difference was drastically reduced with the introduction of focal loss, which compensates for the class-imbalance between foreground (objects to be detected) and background [32].

2.4.3 Anchor boxes and non-maximum suppression

Most state-of-the-art one-stage detection models today make use of anchor box generation, also called multi-reference detection [34]. In this method, a set of reference boxes are predefined and used to make prediction boxes. Each anchor box has one loss for object localization and one loss for classification. This can be combined with multi-resolution detection, that is, combining detections from different layers of the feature extractor, into what is called multi-scale feature fusion (see 2.4.4).

After final predictions are made, one would expect many box predictions, especially around the object. Therefore object detectors use an important post-

processing step called non-maximum suppression (NMS) [34]. There are different ways to do this, but the simplest and most used one is called *greedy selection*. First, all boxes with a confidence score below a chosen score threshold are removed. Of the remaining predictions, the one with the highest confidence is chosen, and all boxes with a certain overlap (IoU higher than a threshold value) are rejected. This greedy process is then continued until a maximum number of box predictions is reached.

2.4.4 Multi-scale feature fusion

Because different layers of the encoder extract different features, a general approach to object detection involves combining these features. Then, both the stronger semantics important for the object classification in the higher, low-resolution layers and localization information in the lower layers are being used. This is called multi-scale feature fusion. An important method for multi-scale feature fusion in object detection is called feature pyramid network (FPN), shown in Figure 2.15 [42].

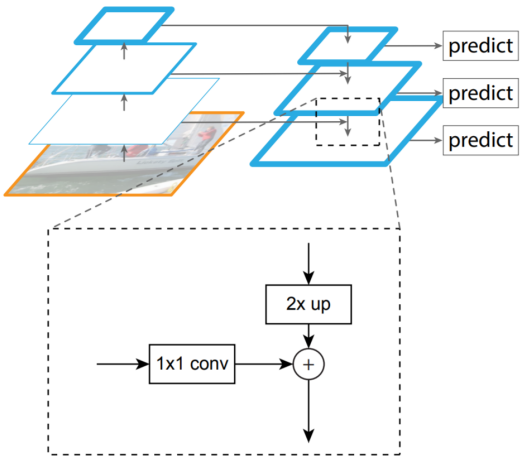


Figure 2.15: The original FPN [42].

The left "bottom-up" pyramid is a regular CNN encoder, while the "top-down" pyramid to the right is the decoding part, used to combine the features. The $2x$ up-scaling and Conv1x1 layer is there for matching the spatial size and number of channels, before the feature maps are added together element-wise. To improve the performance and efficiency of the object detector, this top-down part has been subject to change over the last years.

2.5 Generative Adversarial Networks

Learning input data patterns so that more realistic-looking data could be created was for long considered a difficult task, even when using DL. Generative adversarial networks (GANs) were proposed in 2014 as a technique for overcoming this problem [43]. A GAN is illustrated in Figure 2.16.

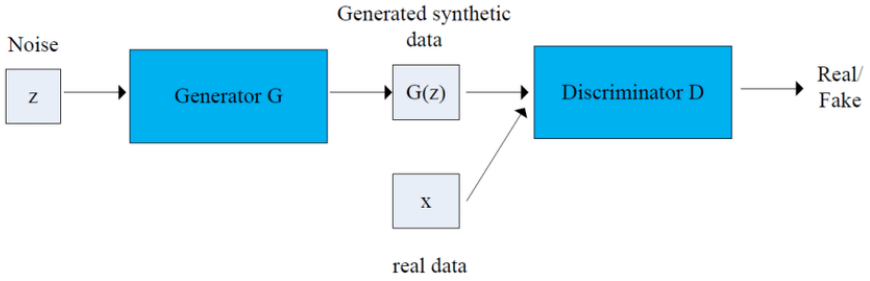


Figure 2.16: A GAN [44].

A GAN consists of two NNs; a generative model G that generates synthetic data from a noise vector z, and a discriminative model D that tries to discern this generated data $G(z)$ from some real data x. These networks are trained simultaneously based on what is called adversarial loss. The ultimate goal is to create a G that produces real-looking synthetic data for some specific problem.

2.5.1 Deriving the objective

In a GAN, the aim of G is to fool D, while D wants to not get fooled by G. Adversarial loss is used to train both networks. Given m training samples, the cost function of D is shown in (2.16).

$$\frac{1}{m}\sum_{i=1}^{m}[logD(x_i)+log(1-D(G(z_i)))] \tag{2.16}$$

Here, x_i is a real input sample and $G(z_i)$ is a fake sample generated by G. D wants to classify x_i as being real (first term) and $G(z_i)$ as being fake (second term). Hence D wants to maximize this function. The cost function of G is the second term of (2.16), given in (2.17).

$$\frac{1}{m}\sum_{i=1}^{m}log(1-D(G(z_i))) \tag{2.17}$$

G wants D to classify the generated data as real; thus, it wants to minimize this function. These two functions can be combined to derive the objective function of a GAN. This is given in (2.18), where $p_{data}(x)$ is the probability distribution of x and $p_z(z)$ is the distribution of z.

$$\min_G \max_D \mathcal{L}_{GAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \quad (2.18)$$

As can be seen, this is indeed an adversarial function; G wants to minimize it, and D wants to maximize it. When such a model is trained, one can expect D to be better at discerning the real data and G to be better at creating realistic-looking data. GANs have proven to be extremely useful in many problems, for instance, in data augmentation. The adversarial loss is also working very well for creating real-looking images [31].

2.5.2 CycleGAN

The CycleGAN is a GAN-based model that is designed for unpaired image-to-image translation [31]. Suppose two image domains X and Y of unpaired images, for instance, Monet paintings and photographs. The aim is now to learn a pattern that connects these two domains such that a photography can be turned into a realistic-looking Monet painting of the same motif, i.e., to transfer the *style* of the image but preserve its content. Using a traditional GAN, where z is replaced with the images to be transformed, will not suffice because content preservation cannot be controlled. G may therefore end up learning to create one painting that will fool D, regardless of its input. CycleGAN is solving this problem by introducing a cycle-consistency loss. Figure 2.17 shows the principles of this model.

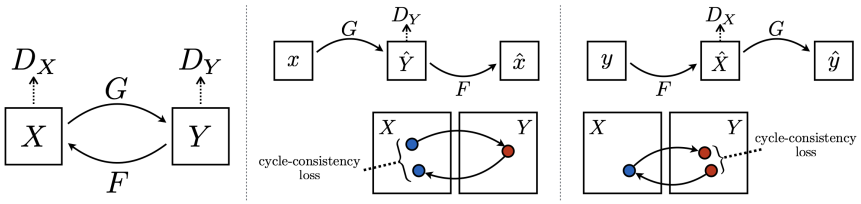


Figure 2.17: From left to right: Components of CycleGAN, forward cycle-consistency loss, backward cycle-consistency loss [31].

CycleGAN is made up of two generators, G and F, and two discriminators, D_X and D_Y. The aim is to learn the mapping $G : X \rightarrow Y$. To secure that the content of the image is not being altered, F converts each generated sample $\hat{y} = G(x)$ back to X. The difference between x and $\hat{x} = F(\hat{y}) = F(G(x))$ is then calculated to get the forward cycle-consistency loss. The concept is illustrated in the middle of Figure 2.17 and expressed in (2.19).

$$\mathcal{L}_{fcyc}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}[\|F(G(x)) - x\|] \quad (2.19)$$

Vice versa (as shown in Figure 2.17 right), a backward cycle-consistency loss can be calculated as shown in 2.20.

$$\mathcal{L}_{bcyc}(F,G) = \mathbb{E}_{y\sim p_{data}(y)}[\|G(F(y)) - y\|] \quad (2.20)$$

Although these functions are measures of simple L1-loss, they have proven to suffice. Adding these together gives the total cycle-consistency loss $\mathcal{L}_{cyc} = \mathcal{L}_{fcyc} + \mathcal{L}_{bcyc}$. By using (2.18) for G and D_Y, and F and D_X, respectively, the full loss function can be defined as in (2.21).

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F) \quad (2.21)$$

λ is a weighting factor between the adversarial losses and the cycle-consistency loss. Finally, the goal is to solve:

$$\arg \min_{G,F} \max_{D_X,D_Y} \mathcal{L}(G, F, D_X, D_Y)$$

Although not relevant for this project, it should be mentioned that an identity loss is also possible to add to preserve colors in the transformation. Figure 2.18 shows how CycleGAN models trained on photographs (X) and different paintings (Y) from famous artists can generate synthetic paintings from the respective painters.

Figure 2.18: From left: Original photo, followed by synthetic paintings of Claude Monet, Vincent van Gogh, and Paul Cézanne [31].

2.6 Narrow-Band Imaging

Most colonoscopy equipment uses a monochromatic camera, as well as a xenon white light source to illuminate the colon [45]. To create a color image, a red (R), a blue (B), and a green (G) filter are in turn applied. Because the three filters have wide bandwidth, these images can, in turn, be put together to construct a color image. This method is called WLI and is the conventional colonoscopy method. In NBI, another filter is placed in front of the light source to filter out specific wavelengths, see Figure 2.19.

The wavelengths used in NBI are around 415 nm (blue) and 540 nm (green) because these are absorbed by hemoglobin in the blood [7]. By filtering out all other wavelengths from the light source, capillaries and vessels in the tissue will appear darker and in higher contrast to the tissue surrounding them. Cancerous tissue tends to be vascular and can therefore easier be identified superficially.

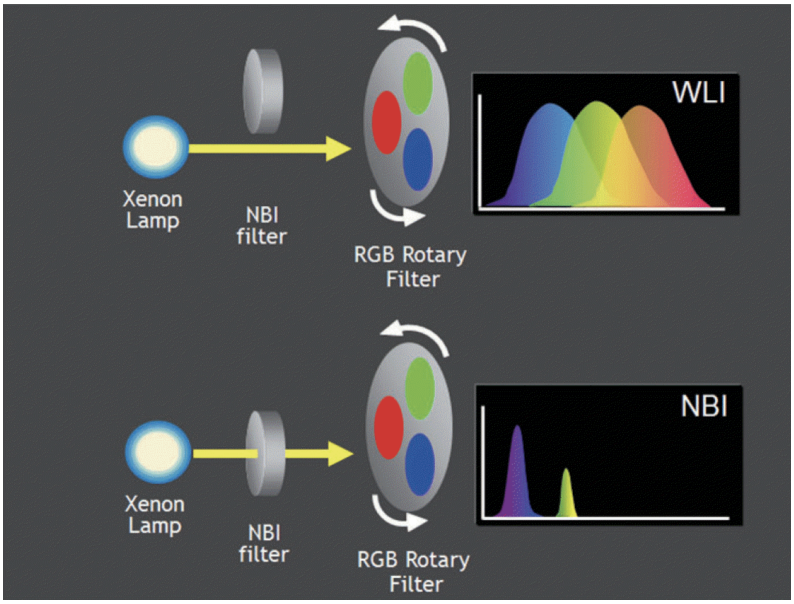


Figure 2.19: The procedure of capturing WLI versus NBI [7].

2.6.1 NICE

Because the superficial structures of cancerous and adenomatous tissue are different from benign ones, the NICE system was invented [10]. The standard was meant to aid endoscopists in identifying the polyp type when looking at images/video captured with NBI. There are three types of polyps, which can be discerned from each other by looking at the color, vessels and surface patterns.

1. Type 1: Hyperplasia. Color is similar or lighter than the background. Vessels are lacy or non-existent. Dark or white spots may appear.
2. Type 2: Adenoma. Color is darker than background. Vessels are brown/dark and surround white structures.
3. Type 3: Deep submucosal invasive cancer. Color is dark and may also contain white areas. Vessels are dark, but some are missing or disrupted.

Polyps of the three types are displayed in Figure 1.1. Although there are other more complicated classification systems, NICE is often used. It suffices for discerning adenomas, which is the essential part of CRC screening.

Chapter 3

Datasets

In this chapter, a presentation is given of the datasets that were used. Some of the general modifications applied to them are also included. In this project, the data needed to satisfy certain criteria:

- Clinical NICE classification annotations (Adenoma and Hyperplasia)
- Clinical detection annotations, either by a binary segmentation mask or bounding box.
- Preferably WLI and NBI images of the same polyps.

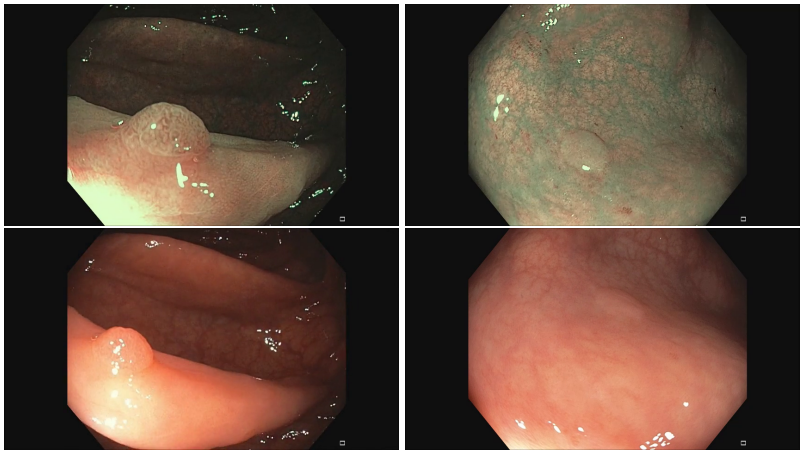
This data would be used to train, evaluate, and test networks for object detection. In addition, images of polyps captured with both NBI and WLI equipment were needed to train the CycleGAN model.

3.1 PICCOLO

The PICCOLO Widefield (PICCOLO) dataset is a Spanish set created by the Spanish Basque Biobank in 2020 [9]. The set contains images of 76 lesions from 40 patients. The dataset contains both NBI and WLI for most polyps, NICE classification, and clinically annotated segmentation masks. All type 3 polyps were removed from the dataset. The PICCOLO dataset was used as train and validation data for the object detection, and prepared and used differently in different experiments. Some examples are presented in Figure 3.1.

3.2 OUS-NBI-ColonVDB

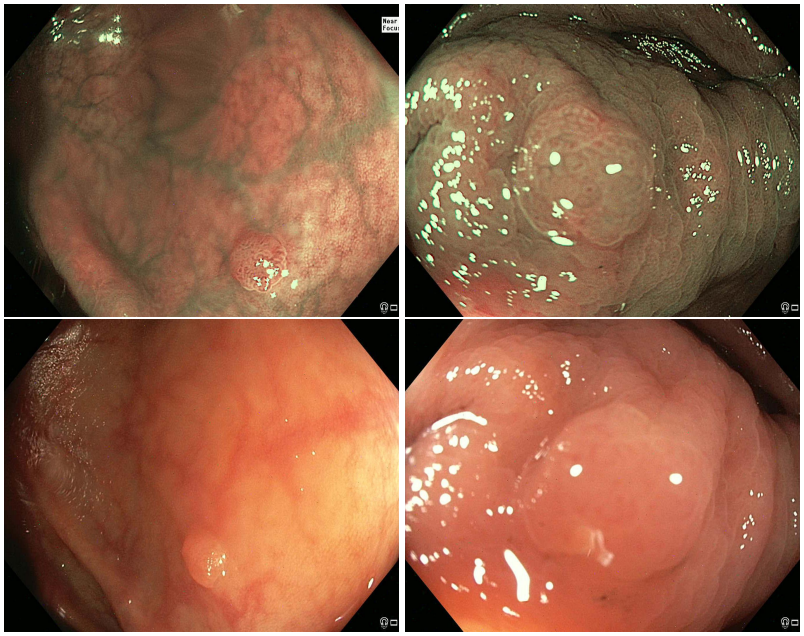
The OUS-NBI-ColonVDB dataset (also called the OUS-set) contains high-quality images from videos of eleven hyperplastic and ten adenomatous polyps. The videos were captured at Oslo University Hospital (OUS) with both NBI and WLI. The polyps are annotated by clinicians as binary segmentation masks. A "cleaned" and reduced version of the dataset was used (as proposed in [1]). All blurry images were removed from the set, and each video was reduced to < 300 images with



(a) Adenoma. **(b)** Hyperplasia.

Figure 3.1: Examples of the (uncut/original) images from both modalities of the PICCOLO dataset [9].

sizes of $1350x1072$ or $620x546$. The dataset was used for validation and testing of the object detection network and used differently for different experiments. Some examples are presented in Figure 3.2.

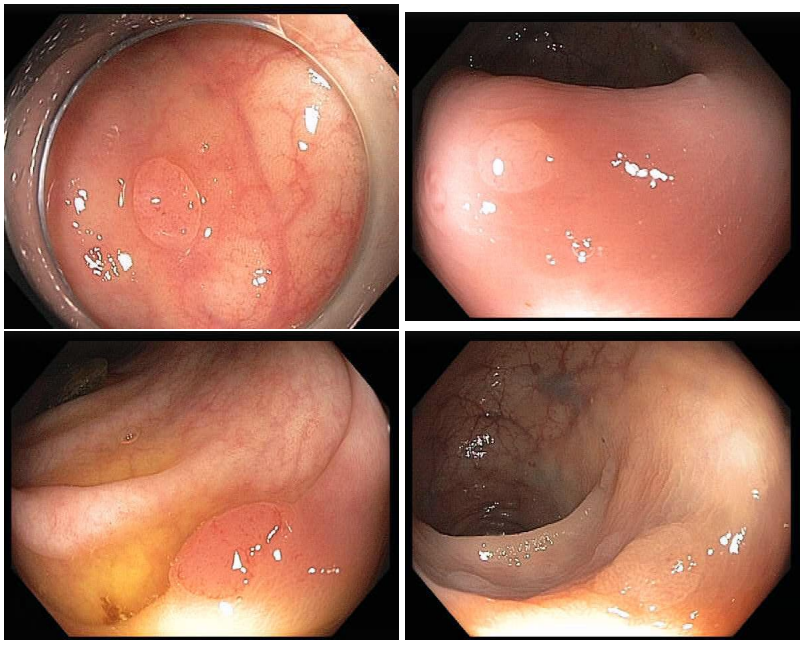


(a) Adenoma. **(b)** Hyperplasia.

Figure 3.2: Example images of two polyps, captured with both modalities, of the OUS-NBI-ColonVDB dataset.

3.3 KUMC

The KUMC dataset is a set of 80 low-quality polyp videos from the University of Kansas Medical Center, published in 2021 [15]. The set contains a mix of NBI and WLI frames from these videos. Each frame has been manually classified into adenoma or hyperplasia and annotated with a bounding box for each polyp present. Some examples are given in Figure 3.3.

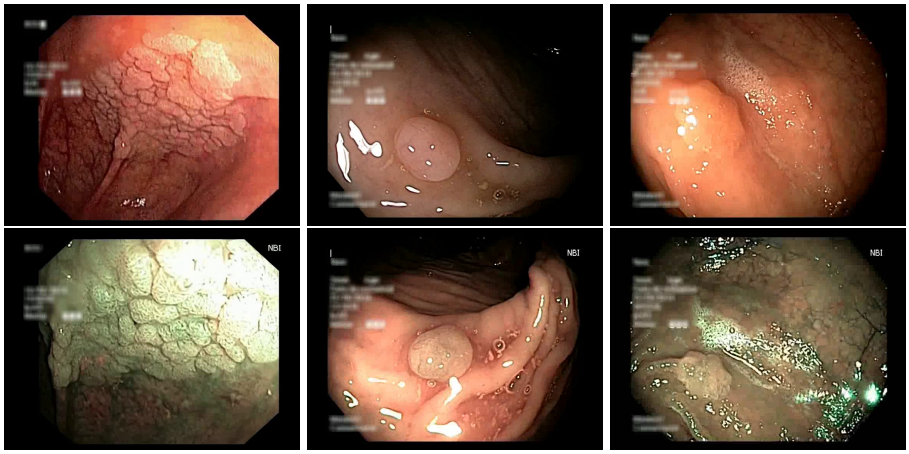


(a) Adenomas. **(b)** Hyperplasias.

Figure 3.3: Examples of WLI frames from four polyps in the KUMC dataset [15].

3.4 Mesejo Videos

A Spanish dataset built by Mesejo et al. in 2016 contains videos of 15 serrated, 21 hyperplastic, and 40 adenomatous polyps, captured with both NBI and WLI [46]. These videos were converted to single-frames (10fps) with size of $768x576$. Blurry images were removed, as well as hyperplastic video #20 because it was a copy of #19. Some example frames are presented in Figure 3.4.



(a) Adenomas. **(b)** Hyperplasias. **(c)** Serrated polyps.

Figure 3.4: Examples of polyps captured with both modalities, from the Mesejo video dataset [46].

Chapter 4

Methods and Implementation

The following chapter describes what methods were used and their implementation. After a quick overview, the DL methods used are presented. The detection experiments conducted are divided into two parts, where the second one follows the first.

4.1 Overview

To answer the questions in this thesis (Section 1.4), the flowchart in Figure 4.1 will be followed.

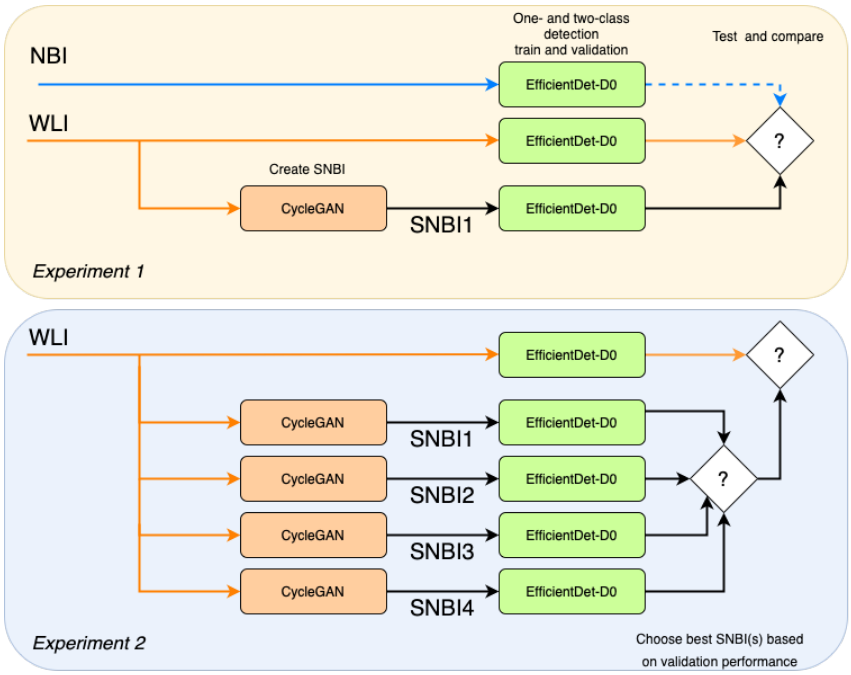


Figure 4.1: Flowchart of of how the NNs are being used.

The experiments are divided into two parts. In short, datasets of polyp images captured with both WLI and NBI will be created from existing datasets. A CycleGAN model will be trained in different ways to create different types of SNBI from the WLI data. Next, to see how the best SNBI compares to the original WLI, independent detection models for both one- and two-class detection will be trained, and their results compared. In experiment 1, an effort will also be made to compare NBI with WLI. However, because NBI consists of other but similar images, the results from the detection models using this modality will mainly be used as a reference and to verify the SNBIs' resemblance to real NBI.

4.2 EfficientDet-D0 for Object Detection

To evaluate the SNBIs, they will be compared to the original WLI. This comparison is done using the EfficientDet, which is a state-of-the-art object detection network proposed by Google AI Lab in 2020 [30]. There are different EfficientDet models with different complexity and performance. Because many experiments were to be conducted, the simplest and fastest one, called EfficientDet-D0, was chosen. Although this network is a one-stage detector, its performance on the COCO dataset is reportedly competitive with two-stage detectors. The EfficientDet-D0 is illustrated in Figure 4.2.

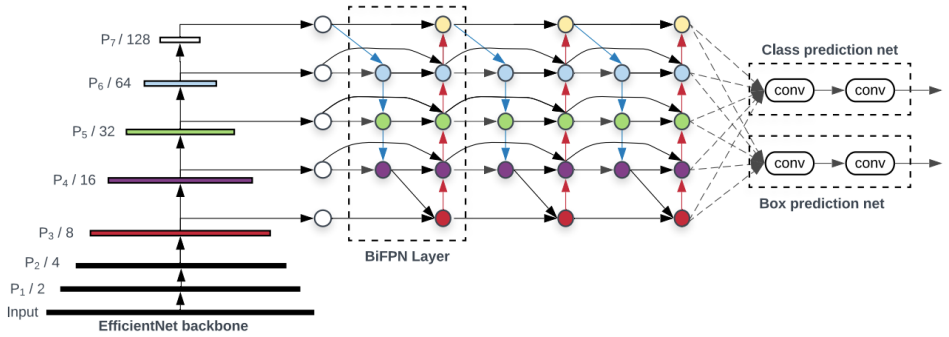


Figure 4.2: Original illustration of the EfficientDet-D0 [30].

EfficientDet-D0 is composed of three main parts, which are further explained below.

EfficientNet backbone

EfficientDet is based on the classification network EfficientNet and uses the EfficientNet backbone to generate feature maps from the raw images [30]. The proposal of EfficientNet contains a new way of scaling the network called compound scaling. Compound scaling suggests a way to balance the scaling of both depth (number of hidden layers), width (number of feature maps in each layer), and

resolution relative to each other. The authors show that this benefits the model's accuracy and efficiency and reports astounding results.

EfficientDet-D0 uses the EfficientNet-B0 baseline, illustrated in Figure 4.3.

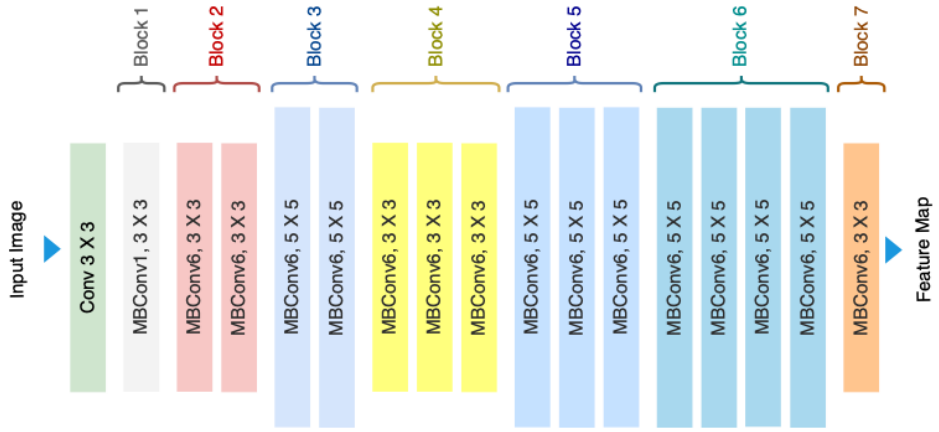


Figure 4.3: Illustration of the EfficientNet-B0 baseline [47].

The baseline consists of a convolutional layer followed by a series of modified MBConv blocks/layers. The output from each (color-coded) block serves as a feature input to the next part of EfficientDet — the BiFPN layer. This backbone has an option for transfer learning, initializing the model with pre-trained weights from an extensive training on ImageNet.

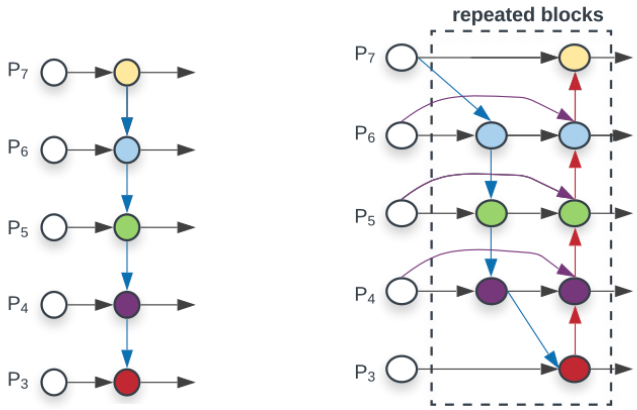
BiFPN layer

EfficientDet-D0 proposes a new method for multi-scale feature fusion, roughly based on the FPN and called bidirectional feature pyramid network (BiFPN). The two are illustrated in Figure 4.4.

Compared to the original FPN, BiFPN consists of repeated blocks with bidirectional cross-scale connections [30]. A weighted feature fusion is also used so the network can learn the importance of the individual features. Like in the backbone, depth-wise separable convolution is applied in the feature fusion to improve efficiency. The number of times the repetitive blocks are repeated is determined by compound scaling. As shown in Figure 4.2, there are three BiFPN blocks in EfficientDet-D0.

Box and Class prediction nets

The class prediction is made simultaneously as the bounding box prediction by using anchor box generation. From the final layer of the BiFPN, its outputs are fused in a softmax classifier using weighted sigmoid focal loss and a bounding box predictor using weighted smooth L1-loss. Smooth L1-loss can be interpreted



(a) Original FPN. **(b)** BiFPN in EfficientDet-D0.

Figure 4.4: FPN vs BiFPN [30].

as a (linear) L1-loss for large (absolute) values of the argument and a (quadratic) L2-loss for small values.

4.3 CycleGAN for Creating Synthetic Images

For creating the SNBI, CycleGAN will be used. This is because this method can enhance images without altering their content. The CycleGAN design is illustrated in Figure 4.5.

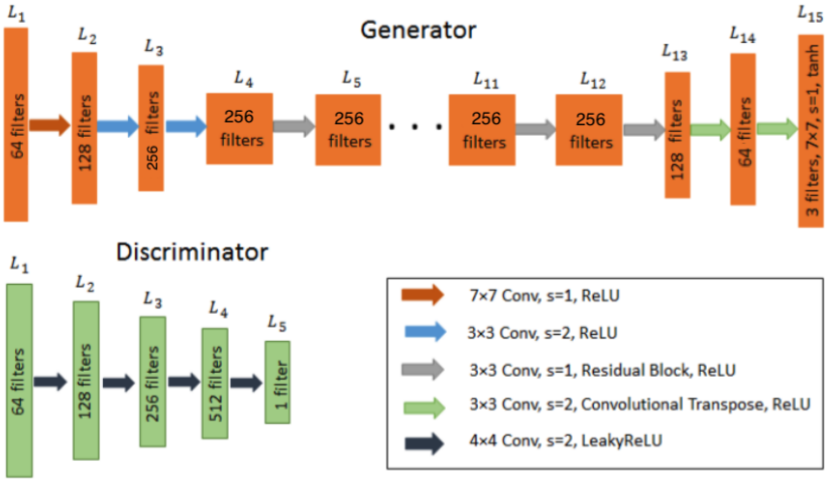


Figure 4.5: CycleGAN block design [1].

The generator begins with an encoder made up of three convolutional layers followed by ReLUs ($L_1 - L_3$). The encoder is followed by a series of residual blocks ($L_4 - L_{12}$) (as in Figure 2.9). The purpose of these is to transform the image. A

decoder ($L_{13} - L_{15}$) follows the transformer and up-samples the feature maps, as well as reduces the number of maps down to three channels (red, green, blue).

The discriminator is a network called PatchGAN [31]. It is made up of convolutional blocks followed by leaky ReLUs.

4.3.1 Implementation and improvements

The CycleGAN was written in the programming language Python, using the DL library PyTorch, and based on an implementation by Aladdin Persson[1][48]. Much of the code was, however, rewritten[2]. A test function was written, using only the WLI-to-NBI generator and a custom Dataset class. Its inference time was measured to see whether the transformation could be used in real-time. Functionality for learning rate scheduling was also added. To preserve more details in the images, resizing was set to 512x512. In the test function, the images were also converted back to their original size after being transformed. Except for the learning rate and the number of epochs, the parameters were set according to the original paper [31].

Data augmentation was applied during training. In addition to resizing and normalization, horizontal flip, vertical flip, random rotation, zoom in/out, and transpose were all implemented with a probability of 0.5 (like in [1]).

During training, CycleGAN is given one WLI and one NBI image for each iteration. Previous experiments showed that although being designed for unpaired data, a kind of "semi-pairing" of the training images would improve the CycleGAN performance [1]. The data and the code was therefore organized such that for each iteration, the two images originated from the same video/colon.

The training of the CycleGAN models was run on an 11GB NVIDIA GeForce GTX 1080 Ti. Training of $15 - 20$ epochs on ca. 10000 images took around 24 hours.

4.3.2 Synthetic NBI datasets

CycleGAN was used to create four different SNBI datasets from real WLI. The datasets examined are presented below:

- **SNBI1:** Trained on the Mesejo set images for 100 epochs with a constant learning rate of 0.0002.
- **SNBI2:** SNBI1, further trained on the OUS-NBI-ColonVDB for 30 epochs with a constant learning rate of 0.0002, followed by a linearly decaying learning rate for another 30 epochs.
- **SNBI3:** SNBI1, further trained on the OUS-NBI-ColonVDB for 60 epochs with a constant learning rate of 0.0002, followed by a linearly decreasing

[1] *https://github.com/aladdinpersson/Machine-Learning-Collection/tree/master/ML/Pytorch/GANs/CycleGAN* (6.6.22).

[2] Complete code for CycleGAN can be found at GitHub: *https://github.com/mathiarh/TFE4940-Masters-Thesis.git* (7.6.22).

learning rate for another 60 epochs (Similar to SNBI2, but retrained longer on the OUS-set).

- **SNBI4:** Trained on the Mesejo set images for 150 epochs with a constant learning rate of 0.0002.

4.4 Experiments

The detection and classification were done by using the TensorFlow 2 Object Detection API[3]. All experiments were run on an 11GB NVIDIA GeForce GTX 1080 Ti. For evaluation and testing, the following metrics were used:

1. mAP was used for both validation and testing.
2. PRC-AUC was used for validation.
3. Precision, recall, and F1 @IoU=.5 were used for validation and testing.

In both experiments, all models were trained separately. Then their precision-recall curve @IoU=.5 was plotted, and the confidence score yielding the best balance between the two was chosen. When testing, this score was used to calculate precision, recall, and F1.

4.4.1 Pre-processing

Different pre-processing was needed before being able to run experiments. All scripts were written in the programming language Python and are available on GitHub.[4]

The first task was to create the datasets, which in some cases had to be done manually (the train, validation, and test sets of experiment 1, the CycleGAN training sets, and the KUMC test set in experiment 2).

Two of the datasets had binary segmentation mask annotations, and a script was written to convert these into a JSON file of bounding boxes. Because most of the masks contained white pixel noise, a lower limit of 300 clustered white pixels was chosen for making a bounding box. Another script converted the KUMC annotation files of bounding boxes from .xml format to a JSON file.

TensorFlow networks need all data to be stored in special binary files called "record" or "TFrecord" files. A Jupyter Notebook was written to convert JSON files of bounding boxes to .csv files, which then could be converted to .record files.

There were also written smaller scripts for specific purposes. A script for calculating aspect ratio of the training images in experiment 2 was written. Another script was made for cutting away the black "curtains" of the PICCOLO set. Also, different code for organizing and renaming all the images in use and their ground truth masks were written.

[3] *https://github.com/tensorflow/models/tree/master/research/object_detection* (6.6.22).
[4] *https://github.com/mathiarh/TFE4940-Masters-Thesis.git* (7.6.22).

4.4.2 Post-processing

A Jupyter Notebook was written for testing and evaluating the detection results. Testing functions were written for making prediction images and storing prediction boxes in a .txt file. These were based on an implementation by Anton Morgunov[5] The .txt files with predictions were used along with the ground-truth JSON files for calculating precision and recall, plotting precision-recall curves, and calculating the PR-AUC under these curves.

4.4.3 Experiment 1: One- and two-class detection

In this experiment, both one-class (*polyp*) and two-class (*Hyperplasia* and *Adenoma*) detection was conducted. The goals were the following:

1. Try to compare WLI with NBI to get an indication of how different their detection performance is. This is why as identical as possible WLI and NBI datasets were used.
2. Compare WLI with SNBI(1) made from the WLI data to see whether images transformed by CycleGAN could improve the automatic detection.
3. Try to tell how well SNBI actually resembles real NBI in terms of detection scores. To assess this, the real NBI models were given SNBI test data. This is named SNBIx.

Preparation of datasets

To be able to compare NBI and WLI imaging, two as identical datasets as possible were constructed. PICCOLO was used as training data, while the OUS-NBI-ColonVDB was split into validation (4 of the 21 videos available) and test (the remaining 17). To get the datasets as similar as possible implied reducing both NBI and WLI images in the two sets so that there were equally many of each modality for each polyp. The training sets ended up having 741 images each. Because both the EfficientDet-D0 and CycleGAN resize the images to 512x512, using more square-shaped images were believed to preserve the information in the images better.

The WLI data was run through the CycleGAN test function to create SNBI1. The models were tested for frame-wise detection on each independent test video and the complete set.

Detection model parameters and augmentation

The EfficientDet-D0 parameters were, in general, based on the original values [30]. Three boxes with width/height ratios of 0.5,1.0, and 2.0 were used for the anchor box generation. The learning rate was linearly increasing for the first 2500 steps, followed by a decrease to zero by the cosine decay rule. NMS was applied

[5] *https://app.neptune.ai/anton-morgunov* (6.6.22).

with an IoU threshold of 0.2, and a maximum of six detections were allowed per image. The training batch size was 8, which was the maximum possible for the GPU used. For one-class detection, the models were trained for 30000 steps, and for two-class detection, 10000 steps. These parameters were found by trial and error based on the validation loss.

Because the training sets only contained 741 images, different augmentation was applied while training. In addition to random vertical and horizontal flipping, random 90° rotation and zoom in/out were applied. Because only 207 of the images contained hyperplastic polyps, a weight of 0.4 was added to the adenoma images when creating the two-class detection training set. This was meant to compensate for the class imbalance as a weight for the classification loss function.

4.4.4 Experiment 2: One- and two-class detection

Experiment 2 was an effort to improve detection results in general, as well as improve the SNBI data. The following was being investigated:

1. Can the SNBI generation be improved? Will retraining the CycleGAN on more high-quality images improve SNBI, i.e., SNBI1 vs. SNBI2, SNBI3, and SNBI4?
2. How does the best SNBI perform compared to the original WLI?

Preparation of datasets

First and foremost, the goal of experiment 2 was to use more high-quality data in an effort to make the CycleGAN even better. Therefore most of the OUS-data were used to train the CycleGAN from experiment 1 even more. All the WLI data (type 1 and 2) from PICCOLO was chosen for training, validation, and testing in EfficientDet-D0. As the dataset was already divided into these three, this partition was kept, although some of the validation data were added to the test and training data to get 48 polyps (ca. 70%) for train and 10 (ca. 15%) for validation and test [15]. Important to note is that the datasets had to be split polyp-wise (not image-wise) such that images from the same polyp/video were not in, for instance, both train and test data. The PICCOLO test set ended up not being used because inconsistent mask annotations were discovered (see 6.6.2). Instead, a test set was manually created by handpicking images of 154 hyperplastic polyps and 154 adenomas from the low-quality KUMC set (from here called the "KUMC-based" set).

Improvements

To make the anchor generation as good as possible for the data in use, height/width and width/height (aspect ratio) histograms were plotted from the training data, see Figure 4.6.

From inspection of these, three new anchor boxes with ratios of 0.7, 1.0, and 1.5 were defined. Using this showed a slight increase in validation mAP during

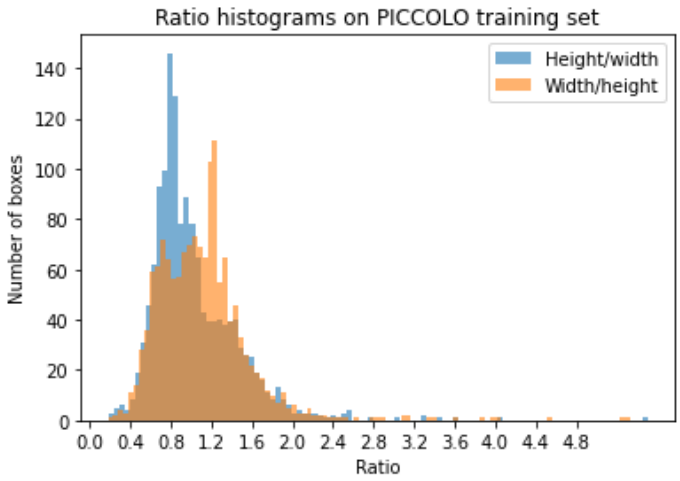


Figure 4.6: Histogram of ratios in the training set of experiment 2.

training, and they were therefore used in all models of experiment 2. The classification models were trained for 15000 steps, found by trial and error. In addition to the augmentation from experiment 1, augmentation for randomly altering the brightness of the images was added. Apart from this, the parameters and training were done exactly like in experiment 1.

Choosing the best transformation

One-class and two-class models were trained for SNBI1, SNBI2, SNBI3, SNBI4, and WLI. By looking at the validation results of the four different synthetic image sets, SNBI2 and SNBI4 got the best performance with regards to two-class and one-class detection, respectively. These were therefore chosen for further testing, along with the WLI set.

One-class detection

WLI, SNBI2, and SNBI4 were tested against each other on the KUMC-based set. The mAP and F1, precision and recall @IoU=.5 were measured and compared.

Two-class detection

It was discovered that the class weighting added in experiment 1 did not seem to work properly. All models tended to predict "Adenoma" on both hyperplastic and adenomatous polyp images. Early tests of experiment 2 confirmed this. Therefore augmentation was done on the training sets in pre-processing, where more augmentation (zoom in/out and flipping) was applied to the hyperplastic images. Classification models were now trained on pre-augmented WLI, SNBI2, and SNBI4. The learning rate was changed to linearly increase for from 0 to 0.03

for the first 1500 steps, before following the cosine decay rule. The augmentation during training was kept, allowing the models to be trained for 17000 steps without overfitting. The models were tested on the KUMC-based test set. The mAP and F1, precision and recall @IoU=.5 were measured and compared.

Chapter 5

Results

From the different experiments, different metrics were calculated. Most importantly is the precision and recall @IoU=.5. In some places where the precision-recall balance was good, only the F1 score is given. However, the precision and recall values are included in Appendix A, where complete results also are provided.

5.1 Experiment 1

Here are the results of the NBI, WLI, and SNBI1 models in one- and two-class detection, respectively.

5.1.1 One-class detection

Table 5.1 shows the results from experiment 1.

Table 5.1: Test results from all videos of experiment 1, one-class detection.

	NBI	WLI	SNBI1	SNBI1x
Precision	0.757	0.55	0.634	0.457
Recall	0.594	0.52	0.537	0.426
F1	0.666	0.535	0.581	0.441

In SNBI1x, the SNBI1 test data is tested on the NBI-trained model.

Because the test set(s) consisted of 17 videos, these metrics were also calculated for each video independently. Here are some observations:

- **WLI vs NBI**: For most of the videos, NBI and WLI got similar results. In some cases, presented in Table 5.2, the difference was larger.
  In total, six of eight adenoma videos were better detected with WLI, while seven of nine hyperplastic polyp videos were better detected by NBI.
- **WLI vs SNBI1**: SNBI1 was getting better results than WLI in eleven of 17 videos; five adenomas and six hyperplasias. There were no considerable

Table 5.2: Test results where difference in both precision and recall was larger than 0.3.

		Precision		Recall	
Video	Type	NBI	WLI	NBI	WLI
#7	Hyperplastic	0.887	0.13	0.8	0.13
#12	Hyperplastic	0.758	0.212	0.676	0.21
#16	Adenoma	0.073	0.421	0.065	0.421
#20	Hyperplastic	0.839	0.481	0.834	0.377

differences between the two models.

- **SNBI1 vs NBI**: SNBI1 beat NBI on seven adenoma videos and one hyperplastic video.
- In some videos, all models performed poorly.

5.1.2 Two-class detection

The test results of two-class detection (detection with classification) is given in Table 5.3.

Table 5.3: Test results from experiment 1, two-class detection.

	NBI	WLI	SNBI1	SNBI1x
Precision	0.655	0.362	0.438	0.496
Recall	0.358	0.307	0.336	0.338
F1	0.463	0.332	0.38	0.402

Here are some observations from the results:

- **WLI vs NBI**: Table 5.4 shows some cases where the results were considerably large between WLI and NBI.

Table 5.4: Test results where differences in both precision and recall was larger than 0.2.

		Precision		Recall	
Video	Type	NBI	WLI	NBI	WLI
#4	Adenoma	1	0.77	0.782	0.504
#10	Adenoma	0.98	0.095	0.467	0.107
#13	Hyperplastic	0	0.543	0	0.227
#14	Hyperplastic	1	0.231	0.928	0.237
#20	Hyperplastic	0.671	0.05	0.62	0.036
#24	Adenoma	0.07	0.504	0.014	0.294

In total, NBI got three adenomas and three hyperplasia videos better than WLI. WLI got five adenomas and one hyperplasia better than NBI. Both mod-

els had precision and recall close to zero for the five remaining videos. These were all hyperplastic.

- **WLI vs SNBI1**: If excluding the results where both models had precision or recall close to zero, the SNBI1 model beat WLI in three adenoma cases and two hyperplastic cases. On the other hand, WLI was better in four adenoma cases and one hyperplastic case.

5.2 Experiment 2

Here are the results from experiment 2 presented.

5.2.1 One-class detection

Table 5.5 shows the test results in terms of F1 score and mAP.

Table 5.5: Test results on the KUMC-based set.

	WLI	SNBI2	SNBI4
mAP (all)	0.439	**0.454**	0.448
F1 (all)	0.685	**0.71**	0.674
F1 (adenomas)	0.734	**0.744**	0.692
F1 (hyperplasias)	0.658	**0.675**	0.656

5.2.2 Two-class detection with pre-process augmentation

Table 5.6 shows the results from classification when class imbalance compensation had been applied as augmentation in the pre-processing.

Table 5.6: Test results on the class-balanced KUMC-based set.

	WLI	SNBI2	SNBI4
mAP (all)	**0.289**	0.242	0.211
F1 (all)	**0.464**	0.387	0.419
F1 (adenomas)	**0.71**	0.486	0.637
F1 (hyperplasias)	0.212	**0.288**	0.203

5.2.3 CycleGAN inference time

The inference time of the CycleGAN test function was measured when creating the SNBI2 training set. Without the time taken for saving the image (254 ms) and the time for creating the Dataset, the generation of one frame of SNBI2 from WLI took 5.3 ms on average.

Chapter 6

Discussion

Here follows the discussion of the results. Since a big part of the results are images, which were not included in the previous chapter, relevant images will be included to support the discussion. To ease the reading, relevant precision and recall values are also included (mainly in the figure descriptions) to support the discussion. The chapter is divided into subsections that compare the different modalities against each other. This precedes a visual evaluation of the SNBIs and a comparison with related work. A final note about errors is then given.

6.1 WLI vs. NBI

One-class detection

As results in experiment 1 indicate, NBI seems to outperform WLI both in the one-class and two-class detection cases. In the one-class detection models from experiment 1, NBI works better for hyperplastic polyps. Hyperplastic polyps are harder to detect manually than adenomas, indicating that the narrow-band light highlights patterns that ease the detection for the automatic model. An example where NBI outperforms WLI is in video #7 from the test set (metrics given in Table 5.2). Figure 6.1 shows images from this video that are considered representative of the two modalities.

As can be seen, the WLI model has problems with discerning the polyp from the background or finding the edge of the polyp. A similar problem can be seen for NBI in video #16, see Figure 6.2.

In this case, the prediction box also covers the ground-truth polyp but is too large. Therefore, the IoU becomes less than 0.5, and the prediction is counted as an FP. However, one can question whether the model prediction is more accurate than the clinical ground-truth annotation. Figure 6.3 shows a WLI prediction from the same video.

Observing the WLI images from video #16 shows that the model predicts the same region as a polyp as the NBI model does. The difference is that the polyp is captured at a different location, making the IoU > 0.5 and the prediction a TP.

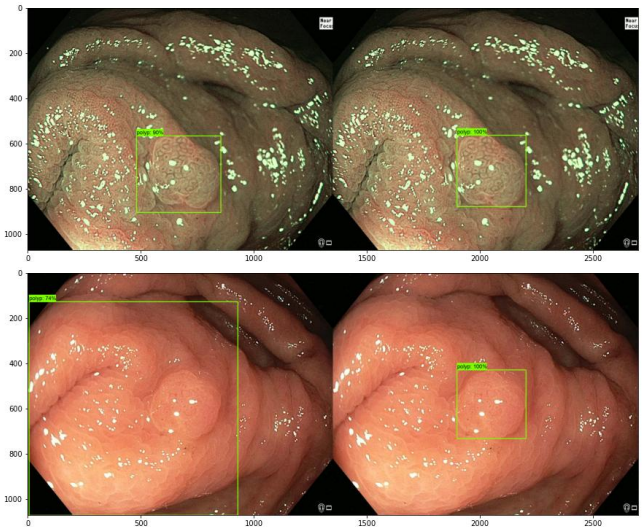


Figure 6.1: From video #7 of a hyperplastic polyp. Predictions to the left and ground truths to the right. Precision/recall was 0.887/0.8 for NBI and 0.13/0.13 for WLI.

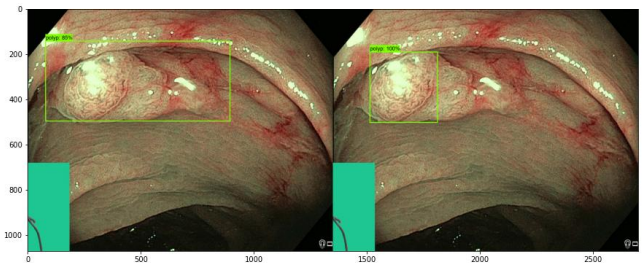


Figure 6.2: From NBI video #16. Prediction to the left and ground-truth to the right. Precision/recall was 0.073/0.065

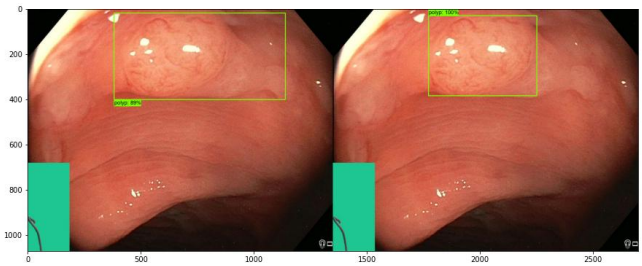


Figure 6.3: From WLI video #16. Prediction to the left and ground-truth to the right. Precision/recall was 0.421/0.421.

The hyperplastic video #20 is also an example where NBI outperforms WLI. A WLI prediction can be seen in Figure 6.4.

While the NBI model in this video has a more steady close-up polyp video,

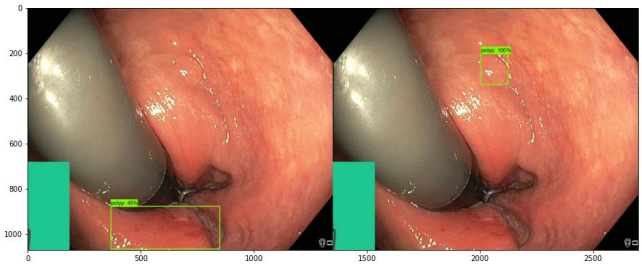


Figure 6.4: From WLI video #20. Prediction to the left and ground-truth to the right. Precision/recall was 0.481/0.377. For NBI of the same colon, the precision/recall was 0.839/0.834.

the WLI suffers from being far away and then very small. The PICCOLO set used in training contains mainly close-up images of the polyps, which has led to all models struggling to detect small polyps in the OUS-set.

The NBI and WLI sets contain images of the same polyps and the datasets were designed to be as similar as possible such that their performance could be compared. However, small differences seem to be crucial for performance in some cases, leaving it unclear whether NBI outperforms WLI in terms of detection. Despite this, one thing that may explain why NBI beats WLI all over is that it tends to be better at finding the polyp region, especially in the hyperplastic polyp videos (like Figure 6.1).

Two-class detection

In two-class detection (Table 5.3), NBI seems to get better results than WLI generally. A case where both modalities performed badly is video #1. While they had one-class detection F1's of 0.98 and 1.0, respectively, their two-class detection F1's on the same video were 0.295 and 0.0. A WLI example is given in Figure 6.5.

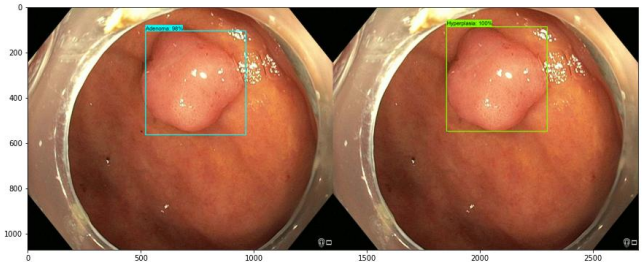


Figure 6.5: From WLI video #1 of a hyperplastic polyp. Prediction to the left and ground truth to the right. For NBI the results were similar.

By inspection of the predictions, both models have no problem detecting the polyp. It is when determining their type that they both perform poorly. Despite the bad performance, note that NBI is better at classifying the polyp than WLI.

The problem of correctly classifying the polyps is also present in other videos,

for instance, #4 (Table 5.4), where the WLI model guesses the polyp to be hyperplastic when it is not. Video #4 is also one of two cases where WLI has better one-class detection results but is worse than NBI for two-class detection. Since no results show the opposite, this indicates that NBI improves classification.

Video #12 and #15 are also examples of where the one-class detection performance is non-zero, but the two-class detection gives zero out for both modalities. This may, at first glance, indicate that the model predictions are so wrong that one could start to question the ground-truth annotations. But by looking at the predictions, one can see that the WLI detects the polyp in these cases but guesses the wrong class. NBI, on the other hand, tends not to create any bounding box at all. This indicates that its classification confidence is below the confidence threshold and therefore neglected. In these two hyperplastic cases, NBI seems less confident than WLI in that the polyps are the wrong class.

6.2 WLI vs. SNBI

While comparing WLI and NBI always involve some imprecision, comparing WLI and SNBI gives much more accurate results because the SNBI data is just enhanced WLI data.

One-class detection

From both experiments, the results show that detection improves with SNBI, especially SNBI2, versus the original WLI. In experiment 1 (Table 5.1), the two modalities have no large difference in results for most videos. There are some small exceptions, for instance, in video #13. Samples are shown in Figure 6.6.

In video #13, the WLI F1 was 0.6, while the SNBI1 F1 was 0.339. As Figure 6.6 indicates, the light reflections are enhanced in the transformation. This may indicate that NBI has more reflection problems, which consequently are amplified with the proposed SNBI transform. Although possibly being a problem in video #13, it does not seem to be a major problem for the detection performance in general.

From experiment 1, one can conclude that the SNBI1 transform, all over, performs slightly better for automatic detection. This is supported by the results from experiment 2 (Table 5.5), although this experiment compares other transforms (SNBI2 and SNBI4) against WLI.

Two-class detection

For two-class detection, there are some differences between the results in experiment 1 and experiment 2. While experiment 1 (Table 5.3) indicates that SNBI generally is better than WLI, the results from experiment 2 (Table 5.6) somewhat indicate the opposite. As pointed out earlier, the trend for all models in experiment 1 is that they do not have problems with detecting the hyperplastic polyps.

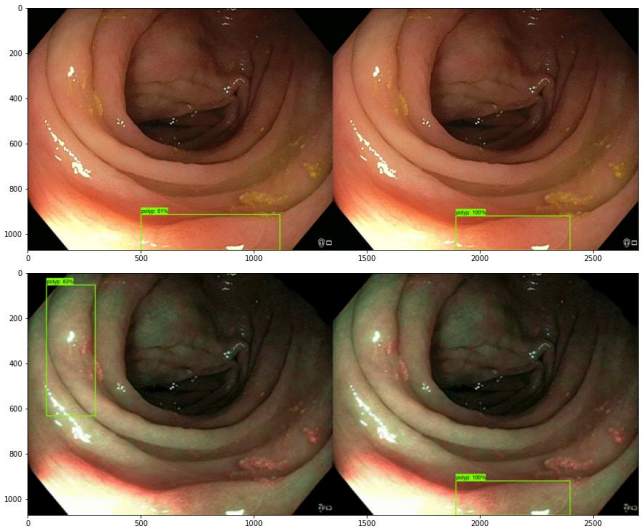


Figure 6.6: From video #13, showing the same frame for WLI and SNBI1. Predictions to the left and ground-truths to the right. Precision/recall was 0.343/0.336 for SNBI1 and 0.713/0.518 for WLI.

It is when classifying them that the models fail. Although being tested on a small dataset, the classification results from experiment 2 are therefore considered to be the most significant.

In experiment 2, SNBI2 shows a relative improvement in classifying hyperplastic polyps while being much worse in adenoma detection. These results could be explained by what is seen in Figure 6.7.

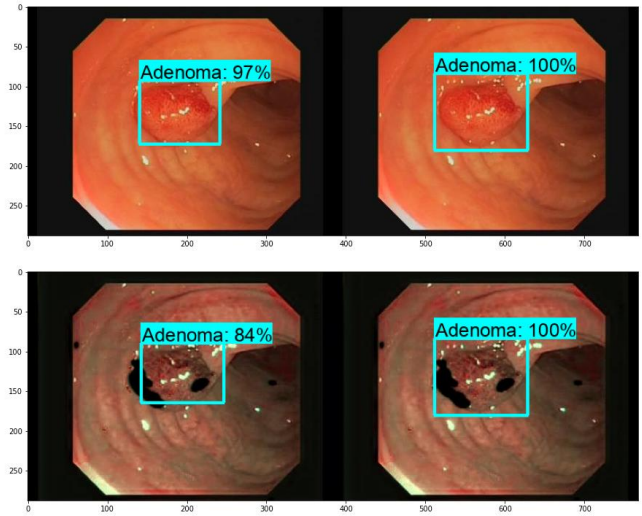


Figure 6.7: From the KUMC-based test set, showing the same frame for WLI and SNBI2. Predictions to the left and ground-truths to the right.

Some of the SNBI2 transformed images have black spots on them. It is not clear why this occurs. One reason might be that because the SNBI2-transformation was trained on high-quality images (the OUS-set) at the end, artifacts appear when trying to transform low-resolution images (the KUMC-based set). SNBI4, which has only been trained on quite low-quality data, supports this argument because it does not contain any such artifacts. If weighting the cycle-consistency loss as more important during training (see λ in (2.21)), the problem might vanish. These black dots do, however, not seem to be the reason for the poor adenoma detection, as most of the adenoma images with black spots were classified similarly to their original WLI frame (like in Figure 6.7). In a few cases, however, the detection seems to be disturbed by these artifacts, as shown in Figure 6.8.

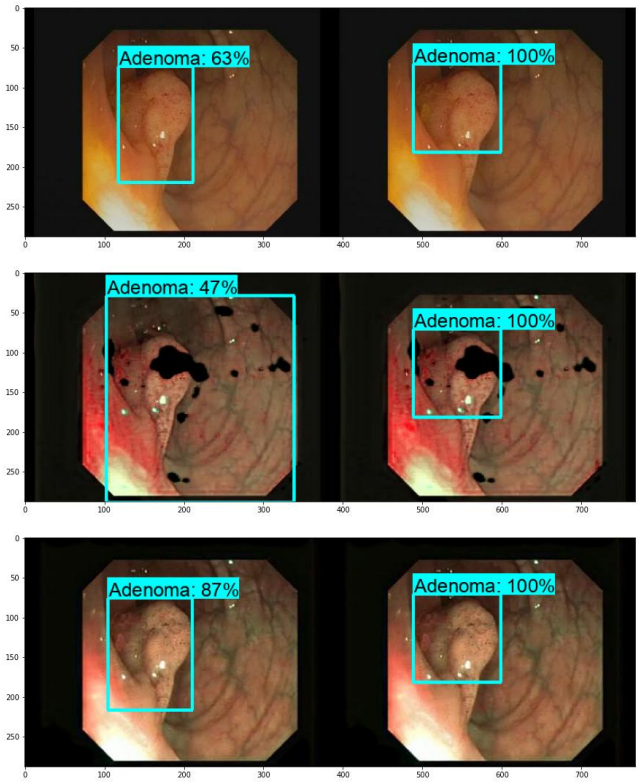


Figure 6.8: From the KUMC-based test set, showing the same frame for WLI (top), SNBI2 (middle) and SNBI4 (bottom). Predictions to the left and ground-truths to the right. As these images indicate, the black dots on SNBI2 can influence its detection performance.

By inspection, it looks like the SNBI2 model just guesses more polyps to be hyperplastic, which increases its F1 on hyperplastic polyps. The downside is that this causes many adenomas to be wrongly labeled.

A note on black spots for detection

If the black spots in the SNBI2 test set in general were concentrated around the polyp, one could also question whether these are the reason why SNBI2 beats WLI in (one-class) detection, i.e., that it helps the model to localize the polyp. Visual inspection indicates that this, however, is not the case. Moreover, the SNBI2 training data do not contain such artifacts, meaning that the model is not trained to "see" them. Therefore, one could argue that these black spots instead would be a disadvantage, making the SNBI2's performance over WLI in one-class detection even more remarkable.

6.3 NBI, SNBI, and SNBIx

SNBIx results from experiment 1 (in Table 5.1 and Table 5.3) gave an F1 of 0.441 for one-class and 0.402 for two-class detection. This is probably the best indication that the proposed SNBI actually resembles (and can resemble) real NBI. It is not known what type of imaging equipment has been used for making the different datasets. Since the NBI technology has been improved over the years, only images from the latest, best-performing technology should be used ideally. Such resources have not been available for this project. Despite this, when using different NBI datasets for the CycleGAN and the NBI detection/classification model, the SNBI can be both detected and classified by the NBI models.

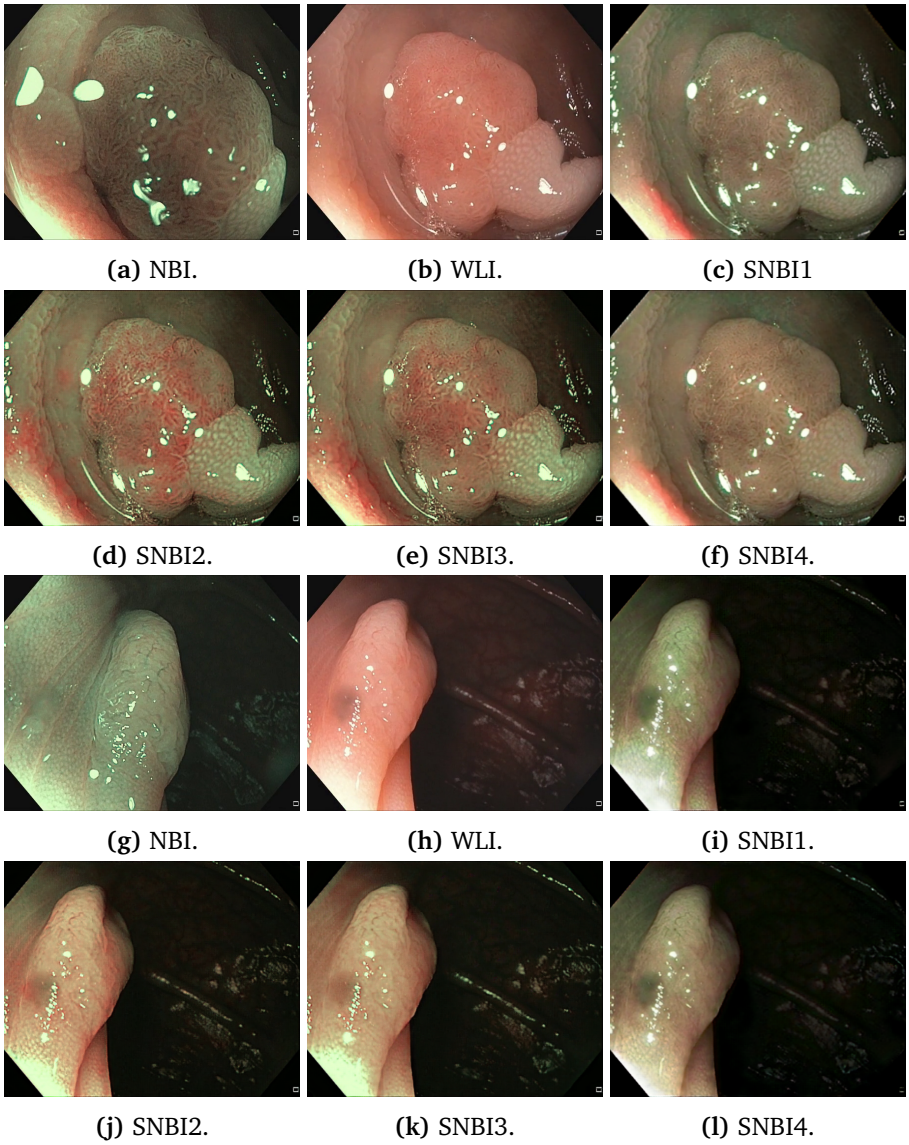
One thing worth mentioning is that even though a perfect comparison of NBI and WLI has not been made, one can argue that NBI at least will be able to perform as well as the SNBI does. This is simply because NBI data has been used to create the SNBI transforms. The features of SNBI that improve its performance compared to the original WLI come arguably from NBI. On the other hand, the features that impair SNBI performance may also come from NBI.

Among the proposed SNBI transforms, the one performing best is the SNBI2. Based on its performance and compared to how the other transforms/SNBIs were created (see subsection 4.3.2), it looks like the best performing SNBI benefits from being trained on different and diverse datasets.

6.4 Visual SNBI Evaluation

Figure 6.9 shows two close-up samples; an adenoma and a hyperplastic polyp of the different SNBI, as well as the original WLI and reference NBI image of the same polyp.

From visual inspection, the transformations that visually best resemble NBI color-wise are SNBI1 and SNBI4. This is arguably because these transformations were only trained on the Mesejo set, which contain very green NBI footage. The OUS-set, which was used to train the SNBI2 and SNBI3 transformations, contains a more red color in the NBI. More specifically, yellow colors appear to be red. The transformations based on this set thus seem to add more red color where the WLI



(a) NBI. **(b)** WLI. **(c)** SNBI1

(d) SNBI2. **(e)** SNBI3. **(f)** SNBI4.

(g) NBI. **(h)** WLI. **(i)** SNBI1.

(j) SNBI2. **(k)** SNBI3. **(l)** SNBI4.

Figure 6.9: From the experiment 2 training sets. On top (two rows) are adenomas and on bottom are hyperplastic polyps.

image contains some yellow color. Note, however, that the red appears slightly more dark in SNBI3. One can hypothesize that further training on the OUS-set would make these red-colored areas even darker, thus resembling the real NBI even better.

In terms of contrast, SNBI2 seems to be the best transformation. This is arguably because of the images' relatively strong red colors and a slightly more coarse resolution than the in other transformations. These features may have been an advantage and explain why this transformation gave the best detection and classification results. SNBI1 and SNBI4 have the poorest contrasts, not only be-

cause they mainly are green but maybe because they have only been trained on low-quality data.

Although they all have flaws and strengths, the different transformations seem to resemble NBI quite well. More training, first and foremost on the OUS-set, could improve the results, especially when tested on high-quality colonoscopy videos. In terms of manual detection, medical expertise is needed to evaluate each transformation's usefulness.

6.5 Brief Comparison with Related Work

The results show that even when using different compensation for class imbalance, adenomas are still easier to detect than hyperplastic polyps. This holds for both one- and two-class detection and matches the findings in [15] and [18] (as presented in Section 1.6). It also makes sense intuitively due to the nature of adenomas. The experiments in this thesis strongly indicate that hyperplastic polyps easier can be detected and classified with NBI. There are indications that NBI improves polyp classification, but as in [19], this cannot be fully confirmed.

Although the results also indicate wrong detections, classification errors are the most prominent. This does not, at first sight, match the results in [21]. However, as previously mentioned, the amount and class distribution of the training data plays an important role when it comes to this errors. The choice of model and model parameters also arguably make an impact.

6.6 Errors

6.6.1 Errors in the class-imbalance compensation

As experiment 1 indicates, the hyperplastic polyps are not classified well compared to the adenomas for any of the models. Although a class weighting was initialized when creating the tensors (.record files), it did not seem to have an effect. Focal loss is designed to account for class imbalance automatically. However, because the detection method uses anchor box generation, the focal loss may be doing its automatic class weighting between foreground and background (misclassified bounding boxes). Despite this, looking at the source code for the *weighted* sigmoid focal classification loss, it still seems possible to initialize class weights. One may suggest that there has been an error in the implementation, causing the weights to have no effect.

Using the pre-process augmentation in experiment 2 seemed to work better than the class weighting. Despite this, the classification performance of adenomas is still much larger than that of hyperplasias. This weakens the suggestion that the class-weight implementation in experiment 1 did not work. Instead, one could blame the lack of hyperplastic polyp diversity in the training set. The simpleness of the pre-process augmentation in experiment 2 did probably not improve the class imbalance enough because the diversity in the augmented set was still low.

The problem may have been possible to overcome by using more complicated augmentation; for instance, would GAN-based augmentation be interesting to try.

Another known measure that maybe could compensate for the class imbalance is the weighting of classification importance relative to detection. As the detection seemed to work better than the classification in general, weighting the classification as more important could teach the network to classify better, even with imbalanced class distribution in the training data.

6.6.2 Errors in the PICCOLO set

When conducting a project like this, one should be able to expect that publicly available datasets are correctly labeled. In addition to the already mentioned dataset errors, a few suspicious annotations were additionally encountered in the PICCOLO dataset.

The PICCOLO set, or parts of it, has been used to train the different detection models. A problem that was discovered when preparing the datasets for experiment 2 is shown in Figure 6.10.

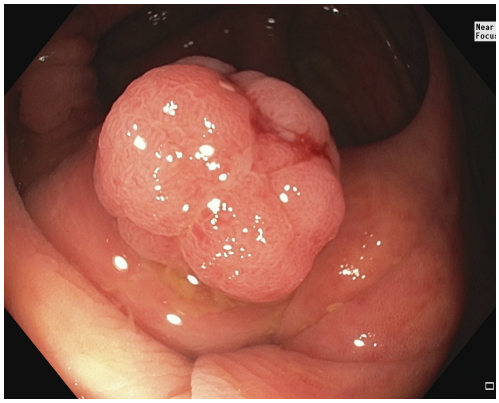


Figure 6.10: A WLI frame of the PICCOLO set, originally classified as type 1 (hyperplastic).

This polyp has been clinically annotated as being a hyperplastic polyp. As this was strongly suspected to be an adenomatous polyp, its label was changed when creating the training set for experiment 2. Some other polyps were suspected of being wrongly labeled but not changed due to unavailable medical expertise. Another clear error was found in the PICCOLO test set for experiment 2 and caused the test set not to be used (replaced by the KUMC-based test set). This is illustrated in Figure 6.11.

Regardless of whether the contour in the background should be labeled or not, it is clear that the images' annotations are inconsistent. This will impose errors in the model predictions.

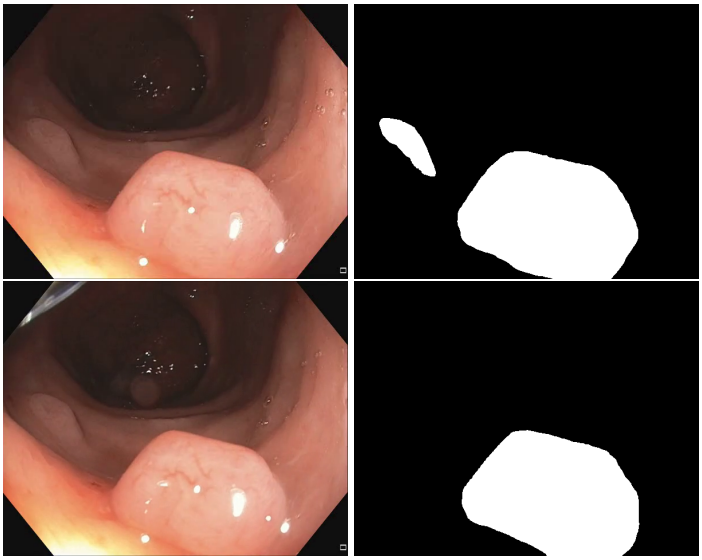


Figure 6.11: Two frames from the (cutted) PICCOLO set to the left, and their respective masks on the right. As can be seen, the mask annotations are inconsistent.

Chapter 7

Conclusion

The scope of this thesis was to find a way to create SNBI from colonoscopy images captured with WLI and then evaluate its usefulness. To create the SNBI, the DL-based method CycleGAN was used. Training data containing polyp images captured with both WLI and NBI were manually created from available datasets and used differently to create four transformation models. Using these, four SNBI datasets were generated from additional WLI data. The different SNBIs were evaluated by the DL-based state-of-the-art object detection network EfficientDet-D0. Independent models were trained, evaluated, and tested for the different (synthetic and real) modalities on both one-class and two-class detection (adenomas vs. hyperplastic polyps). Finally, their results were compared to see how the SNBIs performed against the original WLI and real NBI.

The experiments in this thesis show that it is possible to create a post-process modality transformation from WLI to NBI by using DL. The results show that the transformations ease the automatic detection of polyps, especially hyperplastic polyps. Regarding classification, the most reliable results indicate that SNBI(2) improves the classification of hyperplastic polyps, but is beaten by original WLI for adenomas. Although giving the best results, this transform suffers from black spot artifacts. More research is needed to find the reason for this, but it seems to be related to the CycleGAN training parameters. Clinical expertise will also need to determine the manual relevance of the proposed SNBI.

The experiments involving real NBI indicate that this modality beats traditional WLI imaging, generally in both detection and classification. Because the SNBI transformations are created by NBI images, its improvements over WLI should arguably also apply to NBI. These findings support similar work in general.

Using GAN-based models can be regarded as a novel approach for post-process enhancement of colonoscopy imaging and a novel approach for evaluating the advantage of NBI in colonoscopy. Despite different data flaws, the results show that this method has potential for practical use. Since having an inference time of 5.3ms, the CycleGAN's WLI-to-NBI generator can also be used in real-time applications. If more high-quality images for the CycleGAN are acquired, one may hypothesize that an even more real-looking NBI is possible to create.

Chapter 8

Future Work

The experiments in this thesis show that GAN-based methods can serve as a real-time post-process enhancement of colonoscopy imaging. Several aspects of this topic are still worth investigating.

First and foremost, more correctly annotated data should be used in the evaluation (detection model) of the SNBI[1]. Especially, more hyperplastic polyp images is expected to reduce problems with class imbalance. One could also try applying more advanced data augmentation, for instance GANs, to the gathered data. Moreover, to verify the results in this thesis, the proposed CycleGAN transformations should be further evaluated by other detection and segmentation networks. Different metrics could also be considered; for instance, varying the IoU threshold may give a clearer view of the nature of the different modalities.

In addition to further evaluation, different image-to-image translation methods could be explored for the generation of SNBI. There exists an arsenal of other novel GAN methods for this purpose. Using these might give even better synthetic data. Other DL methods may also be considered, for instance, methods based on *contrast learning*. Acquiring more high-quality data for training such transformations is also considered advantageous. Furthermore, NBI is used not only in colonoscopy but in several other endoscopic examination procedures as well. In lack of colonoscopy data, endoscopic images from other parts of the digestive system may also enable the training of a well-functioning WLI-to-NBI translation. However, the semi-pairing of white and narrow-band data should not be omitted.

Although not being invented for medical use, CycleGAN has proven its usefulness in this field. Based on the findings in this thesis, one can therefore hypothesize that DL-based methods for unpaired image-to-image translation can be applied to transform WLI content to other domains as well, for instance BLI. It might also be applicable for translating WCE content to regular colonoscopy imaging, which might ease both manual and automatic detection and classification of WCE imaging. These methods have great potential for medical use and are worthy of further research.

[1] An updated collection of available colonoscopy datasets are provided here: *https://github.com/sing-group/deep-learning-colonoscopy/blob/master/README.md* (7.6.22).

Bibliography

[1] M. R. Haugland, "Semantic segmentation of colorectal polyp images, and color transformation from WLI to NBI," Department of Electronic Systems, NTNU – Norwegian University of Science and Technology, Project report in TFE4580, Dec. 2021.

[2] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "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, Feb. 2021. DOI: 10.3322/caac.21660.

[3] H. A. Qadir, "Development of image processing algorithms for the automatic screening of colon cancer," Ph.D. dissertation, University of Oslo, 2020.

[4] Y. Xi and P. Xu, "Global colorectal cancer burden in 2020 and projections to 2040," *Translational Oncology*, vol. 14, no. 10, p. 101 174, Oct. 2021. DOI: 10.1016/j.tranon.2021.101174.

[5] C. Mangas-Sanjuan, E. Santana, J. Cubiella, E. Rodríguez-Camacho, A. Seoane, M. A. Alvarez-Gonzalez, A. Suárez, V. Álvarez-García, N. González, A. Luè, L. Cid-Gomez, M. Ponce, L. Bujanda, I. Portillo, M. Pellisé, P. Díez-Redondo, M. Herráiz, A. Ono, Á. Pizarro, P. Zapater, R. Jover, R. Jover, C. Mangas-Sanjuan, E. Santana, J. A. Casellas, F. A. Ruíz-Gómez, E. Serrano, C. Mira, A. Suárez, V. Álvarez-García, O. Castaño, L. Blanco, N. González, J. Lara, E. Quintero, H. Clínic, M. Pellisé, L. Rivero, J. Llach, H. Cordova, I. Araujo, A. Sánchez, I. Ordas, K. Lisette, H. del Mar, A. Seoane, M. A. Álvarez-González, L. Carot, I. A. Ibáñez, F. Riu, M. Pantaleón, J. M. Dedeu, L. E. Barranco, A. Ono, J. Cubiella, E. Rodríguez-Camacho, F. Baiocchi, C. Tejido, L. Bujanda, I. Portillo, I. Idígoras, I. Bilbao, M. Herráiz, C. Carretero, M. Betés, Á. Pizarro, M. Ponce, M. Bustamante, V. Pons, L. Argüello, C. Satorres, P. Díez-Redondo, H. Núñez, V. Busto, L. Cid-Gómez, V. Hernández, L. de Castro, N. Fernández-Fernández, A. Martínez-Turnes, B. Romero-Mosquera, R. Fernández-Poceiro, A. Lué, Á. Lanas, A. Ferrández, and P. Roncales, "Variation in colonoscopy performance measures according to procedure indication," *Clinical Gastroenterology and Hepatology*, vol. 18, no. 5, 1216–1223.e2, May 2020. DOI: 10.1016/j.cgh.2019.08.035.

[6] Y. Saito, S. Oka, T. Kawamura, R. Shimoda, M. Sekiguchi, N. Tamai, K. Hotta, T. Matsuda, M. Misawa, S. Tanaka, Y. Iriguchi, R. Nozaki, H. Yamamoto, M. Yoshida, K. Fujimoto, and H. Inoue, "Colonoscopy screening and surveillance guidelines," *Digestive Endoscopy*, vol. 33, no. 4, pp. 486–519, May 2021. DOI: 10.1111/den.13972.

[7] K. Gono, "Narrow band imaging: Technology basis and research and development history," *Clinical Endoscopy*, vol. 48, no. 6, pp. 476–480, Nov. 2015. DOI: 10.5946/ce.2015.48.6.476.

[8] T. Matsuda, A. Ono, M. Sekiguchi, T. Fujii, and Y. Saito, "Advances in image enhancement in colonoscopy for detection of adenomas," *Nature Reviews Gastroenterology & Hepatology*, vol. 14, no. 5, pp. 305–314, Mar. 2017. DOI: 10.1038/nrgastro.2017.18.

[9] L. F. Sánchez-Peralta, J. B. Pagador, A. Picón, Á. J. Calderón, F. Polo, N. Andraka, R. Bilbao, B. Glover, C. L. Saratxaga, and F. M. Sánchez-Margallo, "PICCOLO white-light and narrow-band imaging colonoscopic dataset: A performance comparative of models and datasets," *Applied Sciences*, vol. 10, no. 23, p. 8501, Nov. 2020. DOI: 10.3390/app10238501.

[10] M. Iwatate, D. Hirata, and Y. Sano, "NBI international colorectal endoscopic (NICE) classification," in *Endoscopy in Early Gastrointestinal Cancers, Volume 1*, Springer Singapore, Oct. 2020, pp. 69–74. DOI: 10.1007/978-981-10-6769-3_8.

[11] Y. Mori, H. Neumann, M. Misawa, S.-e. Kudo, and M. Bretthauer, "Artificial intelligence in colonoscopy - now on the market. what's next?" *Journal of Gastroenterology and Hepatology*, vol. 36, no. 1, pp. 7–11, Jan. 2021. DOI: 10.1111/jgh.15339.

[12] P. Swain, "Wireless capsule endoscopy," *Gut*, vol. 52, no. 90004, pp. 48iv–50, Jun. 2003. DOI: 10.1136/gut.52.suppl_4.iv48.

[13] J. Mi, X. Han, R. Wang, R. Ma, and D. Zhao, "Diagnostic accuracy of wireless capsule endoscopy in polyp recognition using deep learning: A meta-analysis," *International Journal of Clinical Practice*, vol. 2022, P. B. D., Ed., pp. 1–10, Mar. 2022. DOI: 10.1155/2022/9338139.

[14] N. S. Atkinson, S. Ket, P. Bassett, D. Aponte, S. D. Aguiar, N. Gupta, T. Horimatsu, H. Ikematsu, T. Inoue, T. Kaltenbach, W. K. Leung, T. Matsuda, S. Paggi, F. Radaelli, A. Rastogi, D. K. Rex, L. C. Sabbagh, Y. Saito, Y. Sano, G. M. Saracco, B. P. Saunders, C. Senore, R. Soetikno, K. C. Vemulapalli, V. Jairath, and J. E. East, "Narrow-band imaging for detection of neoplasia at colonoscopy: A meta-analysis of data from individual patients in randomized controlled trials," *Gastroenterology*, vol. 157, no. 2, pp. 462–471, Aug. 2019. DOI: 10.1053/j.gastro.2019.04.014.

[15] K. Li, M. I. Fathan, K. Patel, T. Zhang, C. Zhong, A. Bansal, A. Rastogi, J. S. Wang, and G. Wang, "Colonoscopy polyp detection and classification: Dataset creation and comparative evaluations," *PLOS ONE*, vol. 16, no. 8, G. Raja, Ed., e0255809, Aug. 2021. DOI: 10.1371/journal.pone.0255809.

[16] C.-T. Yen, Z.-W. Lai, Y.-T. Lin, and H.-C. Cheng, "Optical design with narrow-band imaging for a capsule endoscope," *Journal of Healthcare Engineering*, vol. 2018, pp. 1–11, 2018. DOI: 10.1155/2018/5830759.

[17] A. Nogueira-Rodríguez, R. Domínguez-Carbajales, H. López-Fernández, Á. Iglesias, J. Cubiella, F. Fdez-Riverola, M. Reboiro-Jato, and D. Glez-Peña, "Deep neural networks approaches for detecting and classifying colorectal polyps," *Neurocomputing*, vol. 423, pp. 721–734, Jan. 2021. DOI: 10.1016/j.neucom.2020.02.123.

[18] X. Liu, Y. Li, J. Yao, B. Chen, J. Song, and X. Yang, "Classification of polyps and adenomas using deep learning model in screening colonoscopy," in *2019 8th International Symposium on Next Generation Electronics (ISNE)*, IEEE, Oct. 2019. DOI: 10.1109/isne.2019.8896649.

[19] T. Ozawa, S. Ishihara, M. Fujishiro, Y. Kumagai, S. Shichijo, and T. Tada, "Automated endoscopic detection and classification of colorectal polyps using convolutional neural networks," *Therapeutic Advances in Gastroenterology*, vol. 13, Jan. 2020. DOI: 10.1177/1756284820910659.

[20] Y. Tian, L. Z. Pu, R. Singh, A. D. Burt, and G. Carneiro, "One-stage five-class polyp detection and classification," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, IEEE, Apr. 2019. DOI: 10.1109/isbi.2019.8759521.

[21] J. Sha, P. Wang, N. Sang, H. Zhang, A. Yang, L. Chen, Z. Gong, C. Li, Y. Qin, X. Li, Y. Ji, and F. Gao, "The value of three narrow-band imaging model in the diagnosis of small colorectal polyps," *Scientific Reports*, vol. 10, no. 1, Dec. 2020. DOI: 10.1038/s41598-020-78708-1.

[22] N. Yoshida, O. Dohi, K. Inoue, R. Yasuda, T. Murakami, R. Hirose, K. Inoue, Y. Naito, Y. Inada, K. Ogiso, Y. Morinaga, M. Kishimoto, R. A. Rani, and Y. Itoh, "Blue laser imaging, blue light imaging, and linked color imaging for the detection and characterization of colorectal tumors," *Gut and Liver*, vol. 13, no. 2, pp. 140–148, Mar. 2019. DOI: 10.5009/gnl18276.

[23] S. Hancock, E. Bowman, J. Prabakaran, M. Benson, R. Agni, P. Pfau, M. Reichelderfer, J. Weiss, and D. Gopal, "Use of i-scan endoscopic image enhancement technology in clinical practice to assist in diagnostic and therapeutic endoscopy: A case series and review of the literature," *Diagnostic and Therapeutic Endoscopy*, vol. 2012, pp. 1–9, Dec. 2012. DOI: 10.1155/2012/193570.

[24] C. Akarsu, N. A. Sahbaz, A. C. Dural, M. G. Unsal, O. Kones, A. Kocatas, I. Halicioglu, and H. Alis, "FICE vs narrow band imaging for in vivo histologic diagnosis of polyps," *JSLS : Journal of the Society of Laparoendoscopic Surgeons*, vol. 20, no. 4, e2016.00084, 2016. DOI: 10.4293/jsls.2016.00084.

[25] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, Jun. 2018. DOI: 10.1007/s13244-018-0639-9.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015. DOI: 10.1038/nature14539.

[27] A. Anaya-Isaza, L. Mera-Jiménez, and M. Zequera-Diaz, "An overview of deep learning in medical imaging," *Informatics in Medicine Unlocked*, vol. 26, p. 100 723, 2021. DOI: 10.1016/j.imu.2021.100723.

[28] M. A. Nielsen, *Neural networks and deep learning*, misc, 2018. [Online]. Available: http://neuralnetworksanddeeplearning.com/.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014. arXiv: 1412.6980 [cs.LG].

[30] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2020. DOI: 10.1109/cvpr42600.2020.01079.

[31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," Mar. 2017. arXiv: 1703.10593 [cs.CV].

[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Aug. 2017. arXiv: 1708.02002 [cs.CV].

[33] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," Nov. 2018. arXiv: 1811.03378 [cs.LG].

[34] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," May 2019. arXiv: 1905.05055 [cs.CV].

[35] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, IEEE, Aug. 2017. DOI: 10.1109/icengtechnol.2017.8308186.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Dec. 2015. arXiv: 1512.03385 [cs.CV].

[37] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520*, Jan. 2018. arXiv: 1801.04381 [cs.CV].

[38] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *International Conference on Machine Learning, 2019*, May 2019. arXiv: 1905.11946 [cs.LG].

[39] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," Nov. 2016. arXiv: 1611.10012 [cs.CV].

[40] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021. DOI: 10.3390/electronics10030279.

[41] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," May 2014. arXiv: 1405.0312 [cs.CV].

[42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Dec. 2016. arXiv: 1612.03144 [cs.CV].

[43] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," Jun. 2014. arXiv: 1406.2661 [stat.ML].

[44] R. Tang and K. Mao, "An improved GANs model for steel plate defect detection," *IOP Conference Series: Materials Science and Engineering*, vol. 790, no. 1, p. 012 110, Mar. 2020. DOI: 10.1088/1757-899x/790/1/012110.

[45] K. Kuznetsov, R. Lambert, and J.-F. Rey, "Narrow-band imaging: Potential and limitations," *Endoscopy*, vol. 38, no. 01, pp. 76–81, Jan. 2006. DOI: 10.1055/s-2005-921114.

[46] P. Mesejo, D. Pizarro, A. Abergel, O. Rouquette, S. Beorchia, L. Poincloux, and A. Bartoli, "Computer-aided classification of gastrointestinal lesions in regular colonoscopy," *IEEE Transactions on Medical Imaging*, vol. 35, no. 9, pp. 2051–2063, Sep. 2016. DOI: 10.1109/tmi.2016.2547947.

[47] T. Ahmed and N. H. N. Sabab, "Classification and understanding of cloud structures via satellite images with efficientunet," Sep. 2020. arXiv: 2009.12931 [eess.IV].

[48] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," Dec. 2019. arXiv: 1912.01703 [cs.LG].

Appendix A

Complete Results

A.1 Experiment 1

One-class detection test results

Test results from each video is given in Table A.1.

Two-class detection test results

Test results from each test video are given in Table A.2.

A.2 Experiment 2

One-class detection on KUMC-based set

Test results are given in Table A.3.

Two-class detection on KUMC-based set

Test results are given in Table A.4.

Table A.1: One-class detection results from experiment 1.

Video: polyp type		NBI	WLI	SNBI1	SNBI1x	Num. of images
V1: HP	Precision	0.989	1	0.985	0.985	274
	Recall	0.971	1	0.985	0.989	
V2: HP	Precision	0.235	0.007	0.22	0.166	297
	Recall	0.158	0.007	0.02	0.128	
V3: A	Precision	0.818	0.992	0.911	0.823	123
	Recall	0.732	0.992	0.911	0.829	
V4: A	Precision	0.879	0.907	0.929	0.325	177
	Recall	0.744	0.897	0.908	0.293	
V5: HP	Precision	0.699	0.648	0.926	0.05	173
	Recall	0.544	0.393	0.289	0.046	
V7: HP	Precision	0.887	0.13	0.314	0.309	285
	Recall	0.8	0.13	0.309	0.309	
V8: A	Precision	0.956	0.87	1	0.971	238
	Recall	0.799	0.87	1	0.971	
V9: A	Precision	0.796	0.861	0.912	0.792	216
	Recall	0.751	0.861	0.861	0.812	
V10: A	Precision	0.317	0.094	0.078	0.014	156
	Recall	0.314	0.099	0.053	0.015	
V12: HP	Precision	0.758	0.212	0.072	0	253
	Recall	0.676	0.21	0.055	0	
V13: HP	Precision	0.677	0.713	0.343	0.284	153
	Recall	0.3	0.518	0.336	0.245	
V14: HP	Precision	1	0.806	0.939	0.222	180
	Recall	0.95	0.753	0.837	0.217	
V15: HP	Precision	0.971	0.683	0.846	0.167	123
	Recall	0.347	0.483	0.371	0.124	
V16: A	Precision	0.073	0.421	0.337	0.355	107
	Recall	0.065	0.421	0.29	0.355	
V18: A	Precision	1	0.844	1	0.967	245
	Recall	0.559	0.841	0.992	0.963	
V20: HP	Precision	0.839	0.481	0.687	0.529	187
	Recall	0.834	0.377	0.493	0.391	
V24: A	Precision	0.081	0.223	0.528	0.111	216
	Recall	0.023	0.216	0.279	0.083	

Table A.2: Two-class detection test results from experiment 1.

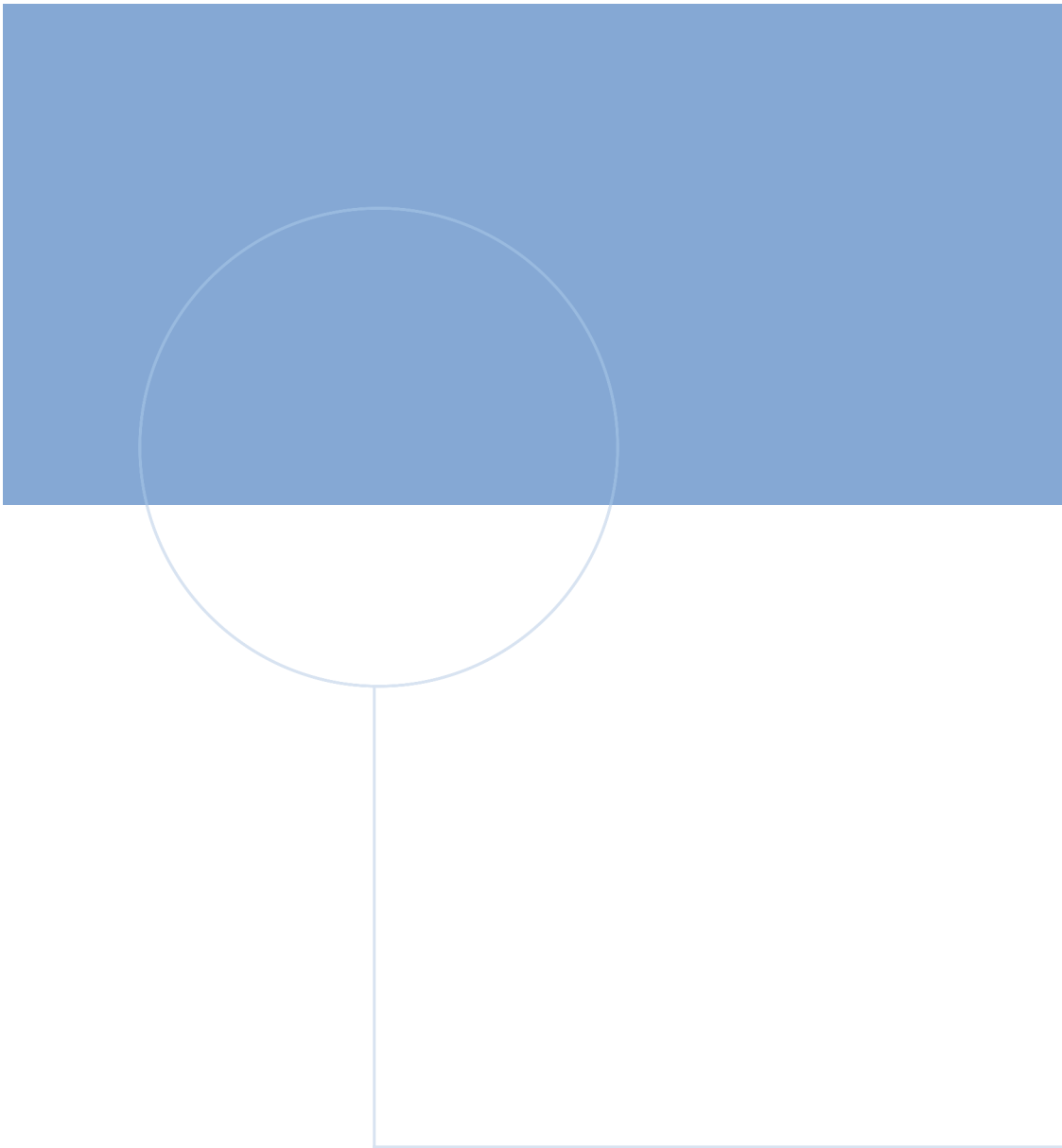
Video: polyp type		NBI	WLI	SNBI1	SNBI1x	Number of images
V1: HP	Precision	0.305	0	0.256	0.695	274
	Recall	0.285	0	0.252	0.657	
V2: HP	Precision	0	0	0	0	297
	Recall	0	0	0	0	
V3: A	Precision	0.899	0.852	0.818	0.823	123
	Recall	0.797	0.886	0.732	0.642	
V4: A	Precision	1	0.582	0.77	0.504	177
	Recall	0.782	0.569	0.713	0.328	
V5: HP	Precision	0	0.04	0.074	0.068	173
	Recall	0	0.006	0.029	0.029	
V7: HP	Precision	0.417	0.049	0.199	0.053	285
	Recall	0.088	0.049	0.193	0.042	
V8: A	Precision	0.995	1	0.903	0.964	238
	Recall	0.807	0.996	0.895	0.912	
V9: A	Precision	0.933	0.849	0.817	0.912	216
	Recall	0.723	0.701	0.743	0.792	
V10: A	Precision	0.98	0.095	0.633	0.482	156
	Recall	0.467	0.107	0.237	0.206	
V12: HP	Precision	0	0	0.006	0	253
	Recall	0	0	0.005	0	
V13: HP	Precision	0	0.543	0.031	0	153
	Recall	0	0.227	0.009	0	
V14: HP	Precision	1	0.231	0.441	0.277	180
	Recall	0.928	0.217	0.361	0.247	
V15: HP	Precision	0	0.02	0.04	0	123
	Recall	0	0.011	0.022	0	
V16: A	Precision	0.229	0.457	0.33	0.653	107
	Recall	0.15	0.402	0.28	0.458	
V18: A	Precision	1	0.964	0.951	1	245
	Recall	0.58	0.869	0.955	1	
V20: HP	Precision	0.671	0.05	0.102	0.07	187
	Recall	0.62	0.036	0.072	0.036	
V24: A	Precision	0.07	0.504	0.141	0.262	216
	Recall	0.014	0.294	0.069	0.108	

Table A.3: One-class detection test results from experiment 2.

	WLI	SNBI2	SNBI4	SNBI2x	SNBI4x	Number of images
Precision (all)	0.678	0.709	0.652	0.394	0.562	308
Recall (all)	0.691	0.711	0.698	0.429	0.607	
Precision (adenoma)	0.734	0.742	0.671	0.38	0.536	154
Recall (adenoma)	0.734	0.747	0.714	0.403	0.584	
Precision (hyperplasia)	0.648	0.675	0.633	0.407	0.588	154
Recall (hyperplasia)	0.669	0.675	0.682	0.454	0.63	

Table A.4: Two-class detection test results from experiment 2.

	WLI	SNBI2	SNBI4	Number of images
Precision (all)	0.446	0.381	0.412	308
Recall (all)	0.484	0.393	0.425	
Precision (adenoma)	0.676	0.478	0.631	154
Recall (adenoma)	0.747	0.494	0.643	
Precision (hyperplasia)	0.205	0.283	0.2	154
Recall (hyperplasia)	0.221	0.292	0.208	



NTNU

Norwegian University of
Science and Technology