UiO : Department of Informatics
University of Oslo

Polyp Detection using Neural Networks

Data Enhancement and Training Optimization

Fredrik Lund Henriksen, Rune Jensen

Master's Thesis Spring 2017

Polyp Detection using Neural Networks

Fredrik Lund Henriksen, Rune Jensen

May, 2017

Acknowledgments

We would like to express our gratitude to our supervisors, Pål Halvorsen, Michael Riegler and Konstantin Pogorelov, for the opportunity to work on this project and all the support given during the thesis. This thesis would not be possible without your aid.

Fredrik would like to thank his mother, Hege Lund, for her care, support and motivation during the thesis. In addition, Fredrik would like to thank his brother, both grandparents and the rest of his family for their support and advice.

Rune would like to thank his parents, Tor Jensen and Kinam Jensen, for their support and the many, many dinners during the thesis. In addition, Rune would like to thank Majkel van den Brink, for moral support and advice.

Abstract

Colorectal cancer is the third most common type of cancer diagnosed for men and the second most for women. Today's main methods of examination are expensive, time consuming and intrusive for the patient. Recent technologies, such as CAD and ACD, aims to increase automation in the screening and examination processes. CAD could aid medical professionals during examinations by providing a second opinion, while ACD could be used to screen entire populations, and thus relieving pressure on the health care system. In recent years, neural networks have gained traction among researchers in topics regarding recognition, and we believe it can be utilized in these automated systems.

In this thesis, we examine the performance of neural networks for polyp detection. We also explore how data enhancement affect the training and evaluation of the networks, and if it can be used to increase the polyp detection rate. Finally, we experiment with how various training techniques can be used to increase performance.

We conclude that neural networks are suitable for polyp detection. We show how data enhancement and training optimization can be used to increase different aspects of the performance. We discuss what aspects are suitable for different scenarios. At the end, we also discuss how our system can be used to detect polyps per frame, per sequence and per polyp, and what the results of our system look like using the different metrics. Detection per frame can be considered a computer science viewpoint, while detection per sequence or per polyp is more of a medical field viewpoint.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

There are a number of medical disorders that can occur in the GI tract, from annoyances to lethal diseases. One example is colorectal cancer, which is the third most common type of cancer diagnosed for men and the second most for women [41]. Today's main methods of examination and screening are colonoscopy, gastroscopy and computed tomography (CT) scan, all of which are both expensive, time consuming and intrusive for the patient. Endoscopy procedures may also involve some level of discomfort for the patient. They all require the use of expensive equipment and medical professionals, making it impossible to screen entire populations.

More recently, Computer Aided Diagnosis (CAD) and Automated Computer Diagnosis (ACD) have emerged, both of which could make the process more automated. CAD aims to help the doctors during examinations by having both the doctor and a detection system searching for diseases, producing a synergistic effect where the computer can provide a second opinion. ACD aims to automate the process in a way where a doctor is not required during the initial screening. This could make it possible for patients to perform the initial screening themselves, putting less strain on the health care system. This increases the scalability by lowering the cost, making it possible to screen a larger share of the population.

There have been conducted a lot of research on this topic, with one example being EIR [44], developed at Simula Research Laboratory. It is a complete pipeline for annotation, detection and visualization of diseases in the GI tract. It uses global image features to detect and categorize diseases, and has proven to produce a high detection rate. EIR can perform the function of both CAD and ACD, where the detection subsystem fulfils ACDs requirements, and the detection subsystem combined with the visualization subsystem fulfils CADs requirements.

EIR is based on global image features, but in recent years, machine learning, and especially neural networks, have gained traction among researchers in topics regarding recognition. For instance, Google has successfully used neural networks in Google Translate, search and more,

and released Tensorflow [15, 1], an open source neural network library.

Even with such popularity, there have been a relatively limited amount of research performed on the use of neural networks for polyp detection, and we have been unable to discover any using TensorFlow. Additionally, little research exist regarding how data enhancement and training optimization affect neural networks.

1.2 Problem Statement

As indicated in the background and motivation section, there have been a relatively limited amount of research completed on neural networks in combination with polyp detection. TensorFlow has recently been released, and is in its early stages of development. It is being used for many purposes, both internally at Google and externally, but has seen limited use in medical scenarios.

In this thesis, we will use an existing, but modified, object detection framework utilizing TensorFlow to investigate the following:

1. *Does neural networks work for polyp detection?*
   This is determined by a comparison with state-of-the-art systems, where we use the best results achieved by a combination of the next two questions.

2. *Can data enhancement methods improve the polyp detection rate?*
   We choose four different data enhancement methods, consisting of rotation, brightness variations, masking reflections and contrast enhancement. These represent a wide range of enhancements, where rotation and brightness variations increase the quantity of data in the dataset, while masking reflections and contrast enhancement increase the quality. To determine whether data enhancement methods can improve the polyp detection rate, a series of experiments are performed to test each method individually and in combinations. To examine their effect, the results are compared against each other and the results where no data enhancement methods have been used.

3. *Can the network architecture be modified to improve the polyp detection rate?*
   We examine four ways to improve the network architecture and analyze how they alter the characteristics of the network. Experiments are performed to analyze neural network architectures (RNN, Inception and Resnet), balances between positive and negative samples in the dataset, training techniques (Rezoom and LSTM), and training optimizers (RMS, SGD and Adam).

In addition, we discuss the findings from both a computer science and a medical viewpoint. Computer science focuses on per frame detection, while medical professionals focus on per polyp detection.

1.3 Limitations

We limit the focus of this thesis to the detection of polyps. There are a large number of possible diseases in the GI tract, but we only have access to a dataset containing polyps, making it hard for us to work with other diseases.

We also limit the thesis to a single object detection framework. Training of neural networks are very time consuming, making it difficult to include additional frameworks with the time constraint we have.

We perform two non-exhaustive experiments, one with different data enhancement methods and another with different training optimizations, where non-optimal solutions may be left out. This is also due to the time constraint of the thesis.

1.4 Research Method

The research presented in this thesis was done in accordance to the *Design paradigm* as described by ACM Task Force in *Computing as a discipline* [10]. We have stated requirements and specifications, and from these, designed and implemented a functional prototype. This prototype was evaluated and improved upon in an iterative manner, based on the results of previous iterations.

1.5 Main Contributions

We provide a deeper understanding of the potential in using neural networks for medical scenarios, especially for polyp detection. We use polyp detection as a scenario to explore how data enhancement methods affect the training and evaluation of neural networks, and what effect each method have on performance. We also explore how various training techniques, including different network models and optimizers, can be used to optimize performance of the overall system. Towards the end, we discuss interesting topics related to neural networks and polyp detection.

In order to achieve this, we create a pipeline to apply data enhancements, and prepare the training and evaluation sets. Each set, consisting of either a combination of data enhancement methods or training settings, will then be used to train a network, and later evaluated. We estimate that we need to train and evaluate approximately 120 sets, each requiring about 17 hours to complete for a total of around 2 000 hours, to be able to draw meaningful conclusions. These conclusions and results will then be the subjects in a comparison with other state-of-the-art systems and various discussions regarding how they can be seen in relation to real world scenarios, and how per frame, per sequence and per polyp detection can be visualized and their use in medical fields.

In section 1.2, we outlined three main questions that we are able to answer as follows:

1. *Does neural networks work for polyp detection?*
   Yes. Compared to state-of-the-art systems, neural networks produces good results, where depending on metrics and scenario, it produces comparable or better results.

2. *Can data enhancement methods improve the polyp detection rate?*
   Yes. Rotation increases the overall performance, and a combination of rotation and contrast enhancement results in the highest number of detected polyps. Additionally, both masking reflections and contrast enhancement show potential depending on the video. Brightness variations, on the other hand, seems unable to produce positive effects.

3. *Can the network architecture be modified to improve the polyp detection rate?*
   Yes. We found that using RNN as the network architecture, a dataset balance with a focus on positive samples, Rezoom as a training technique and SGD as a training optimizer, produce the best results where the detection is increased by up to 300%, while keeping the number of false positives relatively stable.

1.6 Outline

The thesis is structered as follows:

Chapter 2 — Background
We begin by describing the medical background and current screening methods. We then describe EIR, Polyp-Alert and other related work. At the end, we introduce machine learning and neural networks.

Chapter 3 — Polyp detection system and data enhancements
In this chapter, we introduce our polyp detection system, a pipeline from annotated videos to a trained network able to evaluate videos. We then describe the different subparts of the system, including TensorFlow and TensorBox.

Chapter 4 — Experiments
In chapter 4, we start by explaining how the experiments were conducted. Then we do the data enhancement experiment followed by the training optimization experiment. Results are presented per step, including a short discussion regarding the results. At the end, we discuss and summarize.

Chapter 5 — Conclusion
Finally, we conclude the thesis and summarize our findings as well as discuss future work.

Chapter 2

Background

2.1 Medical scenario

There are a number of medical disorders that can occur within the human digestive system, more specifically the gastrointestinal (GI) tract, ranging from annoyances to lethal diseases. One example is colorectal cancer, which is the third most common type of cancer diagnosed for men and the second most for women [41]. Of a population of 100 000, the number of incidences for men is 20.6 and 14.3 for women, whereas the mortality rate is 10.0 and 6.9, respectively, making the lethality of colorectal cancer close to 50%.

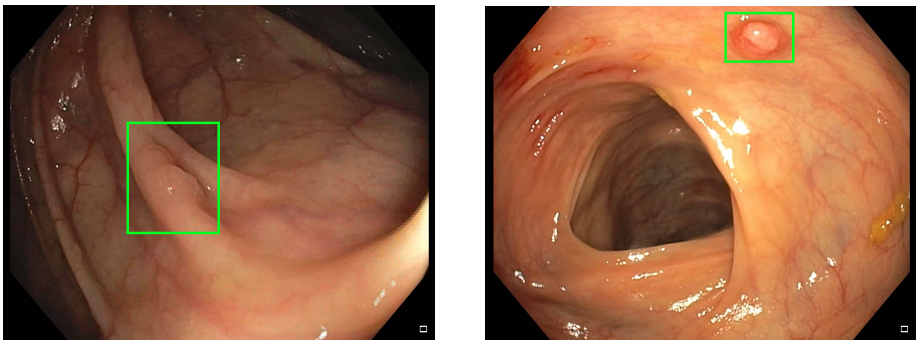


Figure 2.1: Example of polyps in the colon where the polyps are highlighted

A colon polyp, which can be seen in figure 2.1, is a cluster of cells that can develop on the inside of the colon, and often protrude out as a small hill like structure [51]. They are mostly harmless, but are a common precursor for colorectal cancer as, over time, some colon polyps can develop into cancer. A polyp can usually be removed if discovered in an early stage, minimizing the risk of cancer. If a polyp is not removed, the risk of developing cancer at a polyp site is 2.5% at 5 years, 8% at 10 years and 24% at 20 years after the polyp was diagnosed [52]. As there often are no symptoms related to polyps, it is important to have regular screenings. The U.S. Preventive Services Task Force (USPSTF) recommends screening of adults from the age of 50 until the age of 75 [37]. NORCCAP has similar recommendations [14] for Norway.

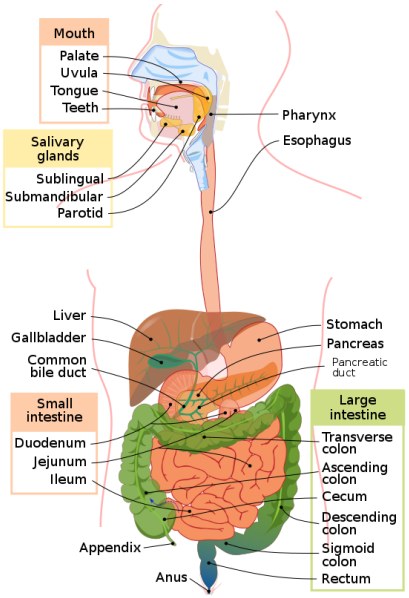


Figure 2.2: Overview of the digestive system[1]

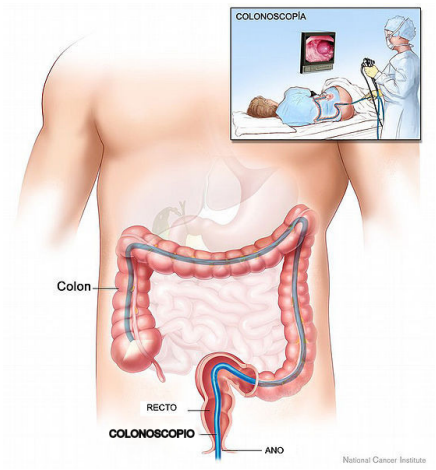


Figure 2.3: Colonoscopy illustration[2]

2.1.1 Endoscopy

Endoscopy is a procedure to look inside the body for medical reasons using an endoscope, a thin, flexible, hollow, lighted tube that has a tiny video camera at the end. Unlike other imaging tests, such as X-rays and CT scans, endoscopy involves inserting an endoscope directly into an organ or cavity. Endoscopes were initially used to look at parts of the body unreachable by other means. Today, endoscopy has many additional uses like prevention, early detection, diagnosis, staging, and treatment of cancer. There are different types of endoscopy procedures using customized endoscopes [50] for different areas of the body. For this thesis, only the endoscopy procedures within the GI tract are relevant, thus other variants of endoscopy are not explained.

2.1.2 Colonoscopy

The common method for performing screening for polyps today is colonoscopy, where an illustration can be seen in figure 2.3. Colonoscopy is a procedure where a doctor uses an endoscope designed for the colon called a colonoscope. The colonoscope is gently inserted into the rectum of the patient, where it transmits a live video feed from within the colon to a monitor. While most people do not find the examination painful,

[1]Figure created and released into the public domain by Mariana Ruiz Villarreal, `https://commons.wikimedia.org/wiki/File:Digestive_system_diagram_en.svg`

[2]Figure is in the public domain because it contains materials that originally came from the National Institutes of Health, `https://commons.wikimedia.org/wiki/File:Colonoscopia.jpg`

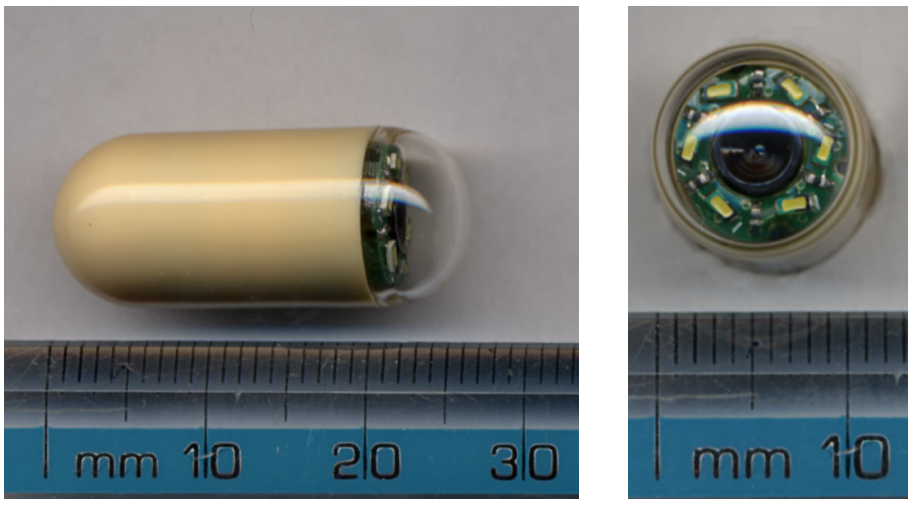
some may find it intrusive. The procedure also involves pumping air into the colon to keep it open in order for the doctor to get clear pictures, which can cause discomfort and cramping in the lower belly region. The examination takes around 30 minutes to complete and is usually performed by a gastroenterologist (a specialist on the gastrointestinal tract) or a surgeon. Around 8 minutes of the procedure is spent on inserting the colonoscope, and the rest is spent on slowly withdrawing the colonoscope while searching for polyps. The doctor can perform a polypectomy (removal of a polyp) during the procedure if the polyp is below a certain size, otherwise surgery may be required [58].

The average cost of a colonoscopy examination in the US in 2012 was $1,185 [47], and requires highly trained personnel, making it a challenging and expensive task to screen an entire population.

2.1.3 Gastroscopy

Gastroscopy is a procedure to look inside of the upper part of the gastrointestinal tract, more precisely the esophagus and the stomach, as can be seen in figure 2.2. During the procedure, an endoscope designed for the esophagus and the stomach is inserted through the mouth to look for symptoms such as inflammation, ulcers or cancer [34].

2.1.4 Wireless Capsule Endoscopy



(a) Side view[3] (b) Front view[4]

Figure 2.4: Endoscopy capsule

Wireless Capsule Endoscopy (WCE or CE) uses a small video camera located inside a pill-like capsule, called a capsule endoscope [28]. An

[3]Figure released into the public domain by Wikimedia user Euchiasmus, `https://en.wikipedia.org/wiki/File:CapsuleEndoscope.jpg`

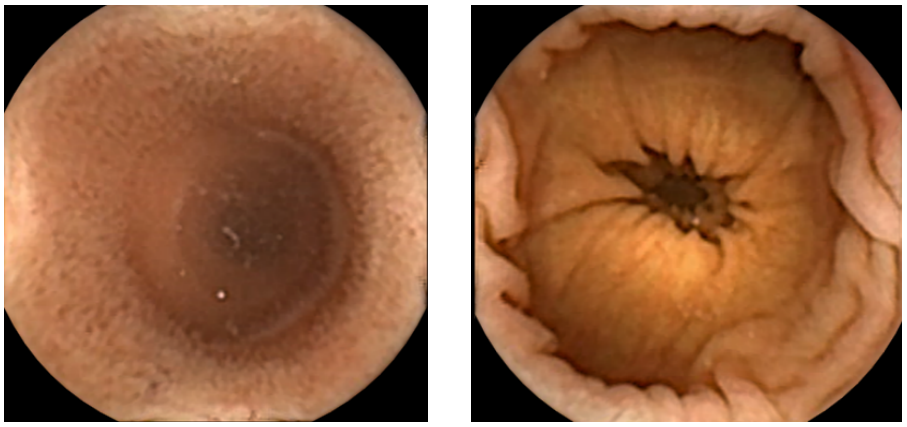
[4]Figure released into the public domain by Wikimedia user Euchiasmus, `https://en.wikipedia.org/wiki/File:CapsuleEndoscopeEnd.jpg`

example of a WCE is depicted in figure 2.4. The capsule is swallowed and travels through the digestive tract, taking pictures and transmitting them to a receiver, until the pill is excreted through the rectum. Example pictures from a WCE are shown in figure 2.5. WCE allows a doctor to see the complete digestive tract, including the small intestine, an area that traditional endoscopy procedures have trouble reaching. In the near future, WCE has the potential to become a cheap, automated, precise and extensive way to screen the whole digestive tract for multiple symptoms.

There are several limitations of WCE today. If the pill either moves too fast or too slow through the system, the pictures can be blurry or the battery can die before reaching the end. There is a small risk that the capsule can get stuck, which may require surgery or endoscopy in order to remove the pill.

A natural companion to WCE is CAD, which uses a computer program to assist the doctor. It may scan the video for symptoms and highlight areas the doctor should pay extra attention to. CAD can potentially increase precision and reduce the time required. Without CAD, a trained professional is required to manually review the approximately eight hours of footage produced by the capsule.

In the future, a goal of WCE is to let individuals buy an inexpensive capsule at a local store and use it at home. By uploading the video to a screening service with ACD, automated mass screening could be a possibility.



(a) The small instenstine[5] (b) The colon[6]

Figure 2.5: Images taken by a capsule

[5]Figure created and published under Creative Commons Attribution-Share Alike 3.0 Unported License by Dr.HH.Krause, `https://en.wikipedia.org/wiki/File:Dünndarm.PNG`

[6]Figure created and published under Creative Commons Attribution-Share Alike 3.0 Unported License by Dr.HH.Krause, `https://en.wikipedia.org/wiki/File:Normales_Colon.PNG`

2.1.5 Automated Computer Diagnosis

Doi [11] talks about ACD as opposed to CAD. CAD is a concept based on the equal roles of medical professionals and computers, where the computers provide a second opinion. The medical professionals make the final decisions, but can utilize the computers to increase their performance. The potential performance increase is due to the synergistic effect obtained by combining the medical professional's competence and the computer's capability.

ACD takes the concept one step further, by automating the process and thus removing the need for a medical professional. The performance level of the computer output needs to be very high, ideally equal to or higher than that of a medical professional. For example, if the computer has a lower detection rate for polyps, it would be hard to justify the use of ACD. The benefits of ACD could include patients being able to perform initial screening themselves without the need for an appointment, making the strain on the health care system lower. This makes it possible to screen an entire population, as the health care system only needs to treat those with positive initial screenings.

2.2 Related work / Polyp detection research

In this section, we will talk about related work in regards to polyp detection; the methods used, how the experiments were performed and the results. EIR, a complete pipeline for disease detection aimed to assist the medical professionals during annotations and examinations, is discussed in section 2.2.1. A list of state-of-the-art systems, gathered from Riegler's PhD Thesis [43], are shown in table 2.1. We will describe some of them briefly, and compare these results to those of our system after all experiments have been completed.

Publication/ System	Positive Recall	Positive Precision	Negative Recall	Negative Precision	Dataset size
Wang et al. [60]	97.7%*	-	95.7%	-	1 800 000 images
Wang et al. [61]	81.40%	-	-	-	1 513 images
Mamonov et al. [31]	47%	-	90%	-	18 968 images
Hwang et al. [20]	96%	83%	-	-	8 621 images
Li et al. [27]	95.07%	-	93.33%	94.20%	300 images
Li and Meng [29]	88.60%	-	96.20%	92.40%	-
Zhou et al. [66]	75%	-	95.92%	90.77%	-
Alexandre et al. [3]	93.69%	-	76.89%	-	35 images
Cheng et al. [8]	86.20%	-	-	-	74 images
Ameling et al. [5]	AUC=95%**	-	-	-	1 736 images
EIR [45, 46]	98.50%	93.88%	72.49%	87.70%	18 781 images

* The sensitivity is based on the number of detected polyps. Other papers use per frame detection.
** Reported only area under the curve (AUC) instead of sensitivity.

Table 2.1: Performance comparison of state-of-the-art systems for polyp detection

Wang et al. [60] introduce Polyp-Alert, a fast polyp detection system using their previous edge-cross section visual features and rule-based classifier [61]. It is able to run on off-the-shelf computers, and is used to assist during colonoscopy procedures. Polyp-Alert calculates the detection rate per polyp, rather than per frame, which is more important in the eyes of medical professionals. In a dataset consisting of 53 videos, it was able to detect 42 of 43 polyps (97.7%), where object tracking was used to track the polyp in preceding and subsequent frames. It is able to achieve a negative recall of 95.7%, which means 4.3% false positives.

Polyp-Alert is a relevant system for us to compare against, as it is designed for a specific real world scenario. It is also interesting as it focuses on per polyp, rather than per frame detection, which we will also discuss using our system in section 4.8.5.

Li et al. [27] propose a new scheme for polop detection in CE images using color and shape features. For color features, HSI color space are used, where only the hue and saturation channels are used to differentiate the colors. For shape features, Zernike, a region-based shape descriptor, is used on the intensity channel to gain an understanding of the different shapes. A dataset of 300 images selected by GI tract experts, where 150 samples contained polyps and 150 did not, were used for evaluation. Multi-Layer Perceptron (MLP) and Support Vector Machines (SVM) were used as classifiers in order to make comparisons between neural networks and other forms of machine learning. They concluded that MLP produced the best results, with an accuracy of 94.20%, a specificity of 93.33% and a sensitivity of 95.07%.

This paper is relevant for us since it shows the potential of neural networks in polyp detection scenarios.

Mamonov et al. [31] propose an algorithm for polyp detection based on extraction of geometric information from the images. This creates the basis for a binary classifier that categorizes the images as either positive or negative samples. For geometric features, protrusion is calculated. If the protrusion is of a sufficent size, the image is classified as a positive sample. A dataset of 18 968 images, where 230 samples contained polyps and 18 738 did not, were used for evaluation. They calculate the polyp detection rate per polyp, rather than per frame, in the same way as Polyp-Alert does. The dataset contains 16 polyps, where a polyp is defined as detected if found in at least one frame. They find 13 of 16 polyps, giving them a polyp detection rate of 81.25%. However, if we calculate the detection rate per frame, only a detection rate of 47% is achieved. In average, they have a false positive rate of 9.8%.

The results of this paper is relevant for us, as we want to see if object detection based on neural networks are able to produce better results.

Zhou et al. [66] assume that a polyp, due to its shape and texture, reflects more light than its surroundings. They present a method to automatically detect and determine the polyps radius in CE frames. A SVM is used for classification. A datset of 359 images were used, 294 for training and 65 for evaluation, with no cross-validation. Of those 65 used for evaluation, 16 contained polyps and 49 did not. They achieved an accuracy of 90.77%, a sensitivity of 75% and a specificity of 95.92%.

In our dataset [56], we can not see the same correlation between reflections and polyps. We have therefore decided to mask the reflections in order to eliminate their effect as opposed to exploiting it, which is further discussed in section 3.1.3.

2.2.1 EIR

Riegler [43] argues that there is a need for improved tools in order to optimize the workflow for medical professionals. EIR [45, 46] is a system developed at Simula Research Laboratory to detect diseases in the GI tract. It is based on the idea of using global features to classify and detect diseases in images. Global image features are features which can describe the content of an image in a single feature, such as color distribution or texture.

EIR consists of the annotation, detection and visualization subsystems. The annotation subsystems main purpose is to gather high quality data for the detection subsystem by giving the medical professionals tools to efficiently annotate videos. A polyp only needs to be annotated once, and the system will try to track the polyp in the previous and subsequent frames automatically. The subsystem is also capable of creating annotation clusters, where each cluster is based on visual global features in the image. This has two main advantages; giving the doctors the possibility to investigate and analyze vast amounts of data, and making this information available for the other subsystems to use.

The detection subsystem uses global image features to automatically classify diseases in images. It is a modular system where it is easy to add support for detection of additional diseases. The detection in itself does not determine the location of the disease within the images, a seperate localization subsystem is used to locate the disease using the output of the detection subsystem.

The visualization subsystems purpose is to visualize the results from the detection. This can be utilized in multiple scenarios, such as aiding the medical professional during a colonoscopy procedure by scanning the live video feed, and thus increasing the combined performance. Another scenario is to share data among researchers and medical professionals.

EIR has been proven to produce high detection rates [43]. Global image features seem to work well for detection and categorization of diseases, but there are still room for improvements. Neural networks is a new trend within recognition, showing great promise. We are curious if such methods could further improve detection rates.

2.3 Machine Learning

Machine learning is the concept where computers gain the ability to learn without being explicitly programmed. It has evolved from artificial intelligence research, and has been one of the hotest topics among researchers in recent years [23]. It learns by making data driven decisions or predictions instead of following static instructions. It alters its own understanding in an iterativ manner by evaluating its current understanding against past understandings, creating a new and improved understanding where the improvements are kept and changes for the worse are discarded. After many iterations, it will have gained a general understanding of the concept.

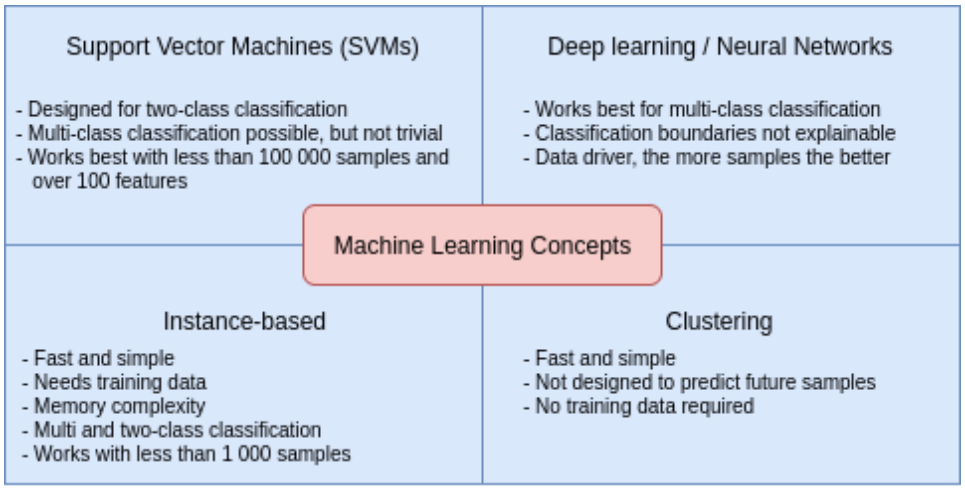


Figure 2.6: Four of the most popular machine learning approaches, based on Figure 2.5, page 24, in Rieglers PhD Thesis [43]

Machine learning is categorized into two categories; supervised and unsupervised learning. In supervised learning, labeled data and a feedback loop is needed. For example, if you train a network for face recognition, the labeled data tells the system which parts of the images are faces. From this data, it tries to gain an understanding of what constitues a face by looking at what the faces have in common. The system uses the feedback loop to improve itself, where it gets a score based on its current performance.

In unsupervised learning, there are no labeling of data and no explicit feedback loop. Instead, it clusters the data into categories. For example, it cannot tell what a face is, but it can differentiate between faces and chairs, placing them into different clusters. This is done by finding similarities in images containing similar items, and using this to classify different concepts into different categories.

The four most popular machine learning approaches, Support Vector Machines (SVMs), Deep Learning/Neural Networks, Instance-based and clustering, are summarized in figure 2.6. In our thesis, neural networks are used, and therefore the only one further described.

2.4 Neural Networks

Neural Networks [49] is a type of machine learning which loosly mimicks how a biological brain learns. Deep neural networks, or deep learning, is the common meaning of neural networks today. They contain multiple layers, making them deep. Each layer can learn different abstraction levels of the data using the input of previous layers until a final layer, which is the final understandig. The information travels through different routes in the network depending on each layers understanding in the same way as a brain works using neurons. The route ends up in a terminal, which is the output of the final layer and the estimation made by the network.

In recent years, neural networks has rapidly gained popularity among researchers in areas of recognition, due to them being able to learn general concepts from concrete examples. Areas that are proven to be well suited for neural networks are speech [18, 2, 16], handwriting [63, 33, 59], and object recognition [35, 22, 36], among others [13, 57, 64, 6, 21].

Nevertheless, neural networks come with several challenges. Firstly, training a neural networks is complicated. It can be seen as a blackbox approach, a concept of using a system without understanding what happens between the input and output, which could be a problem in medical scenarios. Because the system could decide between life and death, the decisions leading to the output should be verifiable and fully understood, in addition to the output itself. If decisions are not fully understood, it is harder to make sure future predictions are correct. Secondly, neural networks require a high amount of training data of sufficient quality and with ground truth. This is especially hard in the medical field since collecting such data requires the time of experts. Additionally, there are many legal and ethical issues. Finally, neural networks are computationally heavy, especially to properly train. While CPUs can be used, the time required to complete training could be months or even years. Due to the advent of GPU computation, the time requirements to train neural networks have become feasible [26].

2.5 Summary

In this chapter, we have discussed diseases in the GI tract, and today's examination and screening methods. Methods such as colonoscopy and gastroscopy, where the doctor uses a camera attached to a tube, are the most common today. We then introduced methods such as CAD, ACD and WCE, which are modern methods where the computer plays a bigger role. Then, we presented modern research in this field where we briefly described some of the state-of-the-art systems such as EIR, a complete pipeline for annotation, detection and visualization of diseases in the GI tract, using global image features for detection and classification. Finally, we explained machine learning, with a focus on neural networks, and discussed its usage in a polyp detection scenario.

Chapter 3

Polyp detection system and data enhancements

In this chapter, we describe our polyp detection system. It is divided into two main parts, training of a neural network and evaluation against the trained network. An overview can be seen in figure 3.1, where training is on the left side and evaluation is on the right.

The training system can be seen as a 5-part pipeline. The first part is input in the form of annotated videos. The second part is the model creation, where videos are split into images, metadata with polyp location are created, and data enhancements may be applied in any combination. The third part is a modified version of TensorBox, which is the neural network framework for object detection we have chosen to use. The fourth part is TensorFlow, an open-source neural network library developed by Google, that TensorBox uses. The fifth and last part is the output in the form of trained weights, the state of the neural network at a given time, which are used for evaluation.

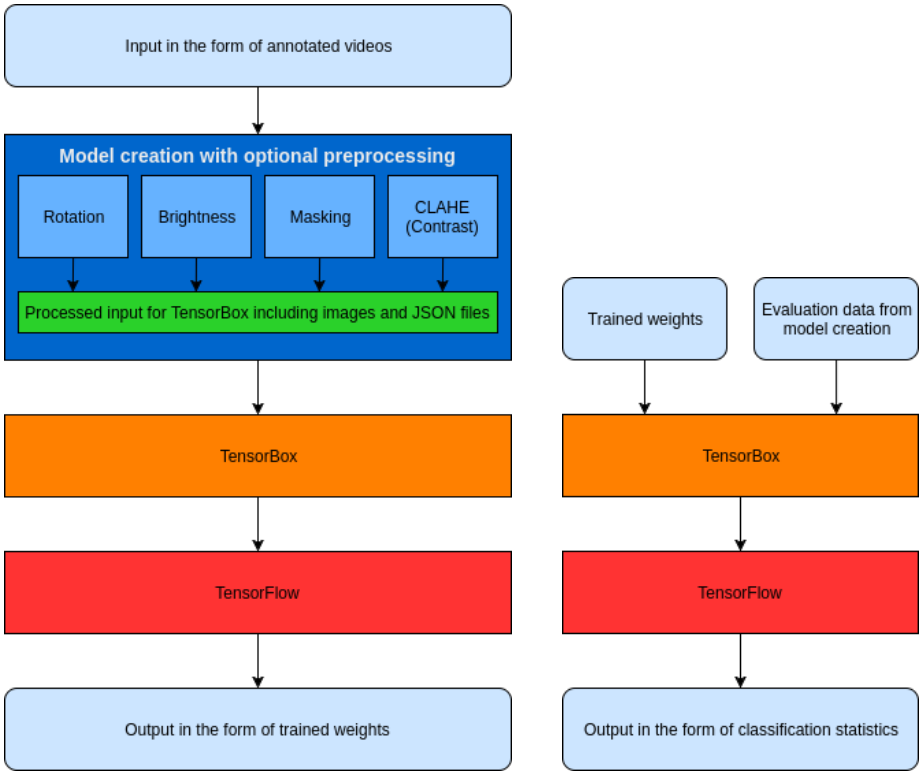
The evaluation system can be seen as a 4-part pipeline. The first part is input, which is the trained weights and evaluation data created as part of the training pipeline. The second is the modified TensorBox. The third part is TensorFlow. The fourth and final part is the output, which is the statistics of correct and wrong classifications.

In the following sections, we will first discuss our choices of data enhancement methods, and then describe each individual step of the training pipeline in detail.

3.1 Data enhancement

In our polyp detection system, as for all computer vision systems, the input data greatly affects the end results. The quality and/or quantity of the input data can be increased by using data enhancement. A higher quality could make the polyps easier to detect, while a higher quantity gives the system more samples to learn from.

In this section, we explain the methods we have tested during the thesis.



(a) Training system (b) Evaluation system

Figure 3.1: Polyp detection system overview, consisting of a training and evaluation subsystem

3.1.1 Artificially increasing the dataset size

One can artificially increase the size of a dataset by transforming the existing dataset in various ways, such as rotation, translation, scaling, flipping, shearing and stretching [65].

If the amount of input data to a neural network is too small or too narrow, it can result in overfitting the network. Overfitting is when the neural network learns details from a specific dataset that are not considered generic. An example would be if a neural network was used to detect bicycles in images and all bicycles in the input data were blue, the neural network could then mistakenly think a bicycle has to be blue.

Another benefit of artificially increasing the size of the dataset is the ability to introduce more variability in the existing dataset, showing the neural network that objects can be in different states. The variations should be done in a way that could be considered logical. For instance, an upside down house may not be considered logical for image recognition since houses are generally never upside down, but brightness variations, scaling and mirroring would result in logical results.

3.1.1.1 Image rotation

Amaral et al. [4] has performed experiments using rotated images, and were able to increase the detection rate by between 8% and 42%. They had a low number of images as input, and argues that the main benefit of rotation is to increase the amount of input data. We, on the other hand, have a large amount of input data, so it may not be as beneficial, but we still want to test if the detection rate can be further increased.

Polyps have no logical up or down as they can be found anywhere inside the colon. If the neural network sees a polyp on the bottom of the colon, growing upwards, we want to show the network that the polyp could just as well be on the right wall, growing leftwards, or have any other rotation. Because of this, we believe that rotation of images could benefit polyp detection.

3.1.1.2 Brightness

While we have not found any experiments which explicitly uses brightness variations to increase the dataset size, we believe it could be a way to augment the dataset in the same way as for rotation.

To be able to capture video inside the colon, one needs to have a light source. Depending on the light source, there could be differences in the brightness levels in different parts of the image. Since a polyp can be found anywhere in the image, we believe that showing the neural network polyp images with different brightness levels could lead to improved detection.

3.1.2 Contrast enhancement

In our dataset, it can be a challenge to distinguish the polyps from the surrounding areas. A possible way to improve polyp detection is to enhance the contrast in the images. Yadav et al. [62] were able to increase the number of detectable edges in images with heavy fog by enhancing the contrast. While we have no images with fog, contrast enhancement could be beneficial by increasing the detail level in low-contrast areas.

The Adaptive Histogram Equalization (AHE) [38] is a technique to perform contrast enhancement. AHE, in contrast to ordinary histogram equalizations, uses the neighbouring regions to derive a transformation function. The benefit of this is that dark and light regions within the image are also sufficiently enhanced, since it adapts the function to local areas in the image.

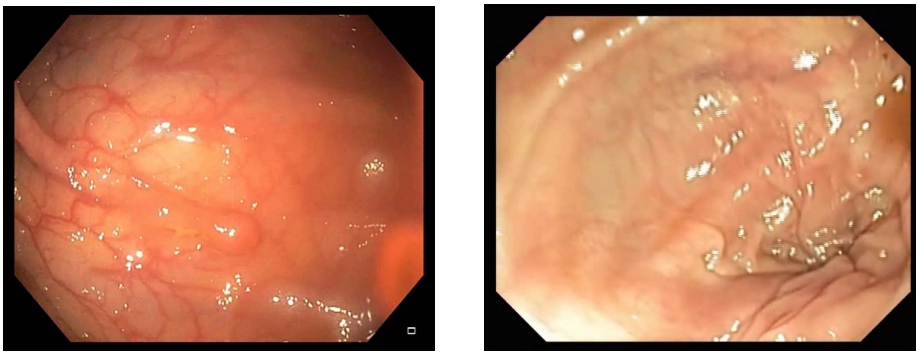
AHE can result in overamplification of noise. Contrast Limited AHE (CLAHE) [68] is an optimization which avoids this problem by limiting the amplification. It clips the histogram at a predefined limit, and distributes the clipped part among surrounding areas, preserving the clipped part while limiting the amplification.

In theory, CLAHE should be able to improve the detection by enhancing the edges of the polyps.

3.1.3 Masking reflections

The light source, which is needed to capture video inside the colon, can potentially create sharp reflections since the colon surface can be uneven and contain fluids. Zhou et al. [67] use these reflections for detection and polyp measurements. In the dataset we use, we cannot see such a correlation between reflections and polyps.

As can be seen in figure 3.2, both the image with a polyp and the image without contains similar types of reflections. In addition, the image with the polyp has similar reflections both on the polyp and the surrounding area. As such, it could be beneficial for the polyp detection to remove the reflections, letting the neural network focus on other features of the polyp.



(a) Image containing a polyp (b) Image not containing a polyp

Figure 3.2: Reflections in colonoscopy images

3.2 Model creation

In this section, we elaborate upon how we convert the input data, from annotated videos to a usable data format for TensorBox. This step is represented as the second block in figure 3.1a, Model creation.

Model creation is handled through a series of tools that preprocess and produce data in a format required for the next step. The source code for all the tools are available on github[1].

3.2.1 Model Creator

TensorBox, further explained in section 3.4, requires input in the form of json files containing the path and polyp coordinates for each image, in addition to the images themselves. JSON, JavaScript Object Notation, is a human readable open standard data interchange format [7, 12].

Model Creator is a python script we created to automate the process of generating the json files and prepare the images listed in them. It traverses the input folders looking for videos with tiff images, extracts the individual

[1] https://github.com/FredrikAndRuneMaster/MasterThesis

```
{
  "image_path": "Model1_rbm_train_images/ShortVD_wp_49-497_r90.png",
  "rects": [
    {
      "x1": 370.0,
      "x2": 409.0,
      "y1": 230.0,
      "y2": 292.0
    }
  ]
},
{
  "image_path": "Model1_rbm_train_images/ShortVD_wp_49-497_r180.png",
  "rects": [
    {
      "x1": 230.0,
      "x2": 292.0,
      "y1": 71.0,
      "y2": 110.0
    }
  ]
}
```


Figure 3.3: Snippet from a training file for Split 1, in json format

frames using ffmpeg[2] from the videos, and scans each corresponding tiff image for the polyp location. The tiff files are the annotation of each individual frame, denoting the ground truth of the polyp, and are used as binary classifiers. White areas denote polyps while the rest is black. An example can be seen in figure 4.1.

At the end, it stores the information in separate json files, one for training and one for evaluation. A short snippet of such a json file can be seen in figure 3.3.

The scanning is performed by iterating over the pixel values in the corresponding tiff image looking for white pixels. If any white pixels are found, the highest and lowest coordinates in both axes are saved, forming a rectangle around the polyp. If no white pixels are found, the coordinate attribute for the image will be empty.

Model Creator may also execute tools we have made for data enhancement on each extracted frame, depending on arguments given. These tools are described in the following sections.

The steps are as follows:

- Extract the frames
- Scan for the polyp location
- Mask the reflections in the image (optional)
- Contrast enhance the image details (optional)
- Generate rotated variants of the image (optional)
- Generate variants of the image with different brightness levels (optional)

[2]https://ffmpeg.org/

By using this tool, we are able to generate datasets in an automated way where all possible combinations can be made with a single command.

3.2.2 Masking reflections

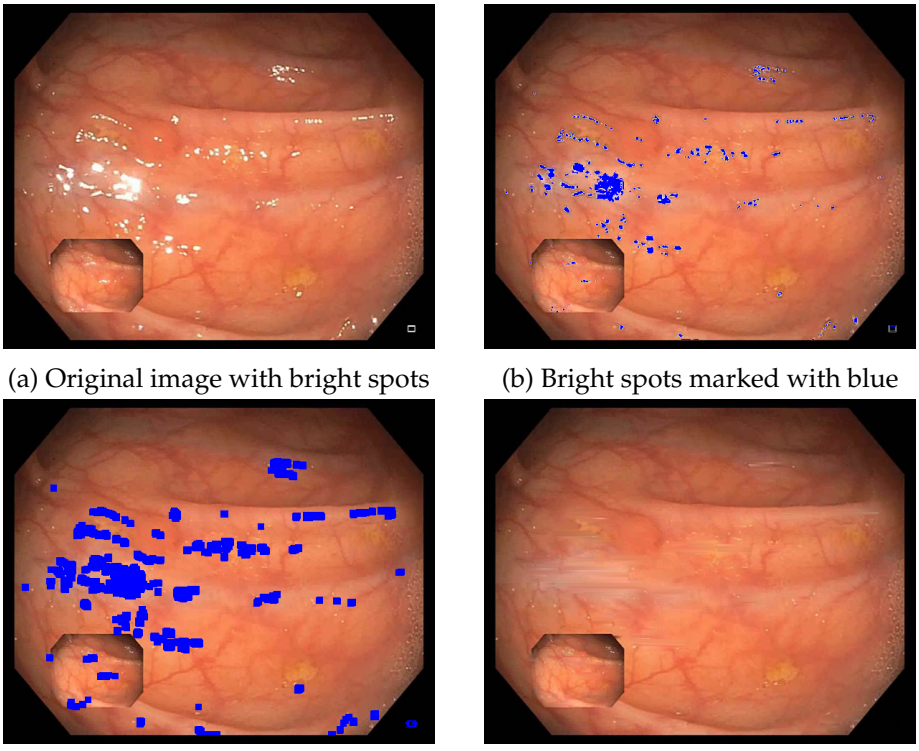
To perform the masking of reflections in the images, we have written our own tool called `masking_reflections.py`. It consists of three steps; marking bright areas, padding marked areas, and filling marked areas with surrounding colors.

Marking bright areas is done by iterating over the pixels in the image, and for each pixel, check if any of the RGB channels is over a certain limit. If it is, the pixel is marked by coloring it blue.

Padding the marked areas is done by iterating over the image again. Each pixel within a given radius from a marked pixel will also be marked.

Filling the marked areas is done by iterating over the image one last time. For each marked area, we find the color to the left and right, and color the pixels as a gradient color between the left and right color. If no valid color is found in either direction, we try the pixel above instead.

Each individual step is shown in figure 3.4, where the RGB limit is (240, 150, 150) and the padding radius is 5 pixels.



(a) Original image with bright spots

(b) Bright spots marked with blue

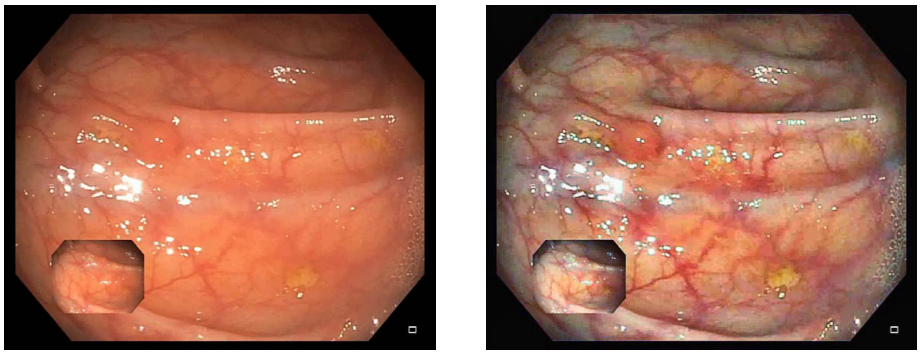
(c) Marked areas are padded

(d) Bright spots filled

Figure 3.4: The steps taken during masking reflections

3.2.3 Contrast enhancement

To perform the contrast enhancement, we have written a small C++ program called `clahe_filter.cc` that uses OpenCVs Histogram Calculation module[3]. We use `createCLAHE()` to create the CLAHE and apply it for each RGB channel in the image. At the end, the original image is replaced with the enhanced version, making updating file names unnecessary and avoids having duplicate images. An example of this can be seen in figure 3.5



(a) Original image (b) Enhanced image

Figure 3.5: Example of contrast enhancement

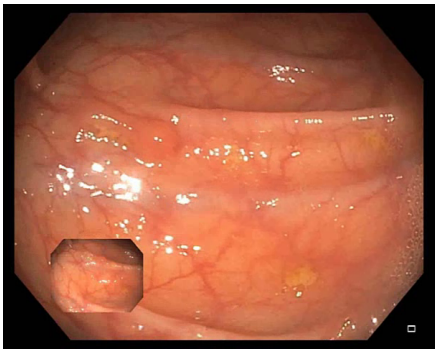
3.2.4 Rotation of images

Rotation of images are performed on each image that contains a polyp and is part of the training data. The original image is duplicated three times, with 90°, 180° and 270° rotated variants, in addition to the original. Choosing random degrees could have been an option, but we wanted to ensure that the variants would be distinctly different. To perform the rotation, PILs rotate functionality[4] is utilized. All rotations of the image are added to the set of images used in the next step. The result of a rotation can be seen in figure 3.6.

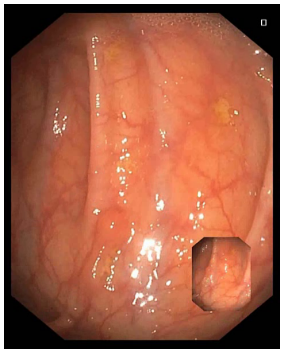
3.2.5 Brightness variations

As with rotation, only images that contain a polyp and are part of the training data are brightness adjusted. The original image is duplicated three times, with 33%, 66% and 133% brightness level variants, in addition to the original. Choosing random percentages could have been an option, but we wanted to ensure that the variants would be distinctly different. To alter the brightness, PILs ImageEnhance module with its brightness functionality[5] is utilized. The result of a brightness alteration can be seen in figure 3.7.

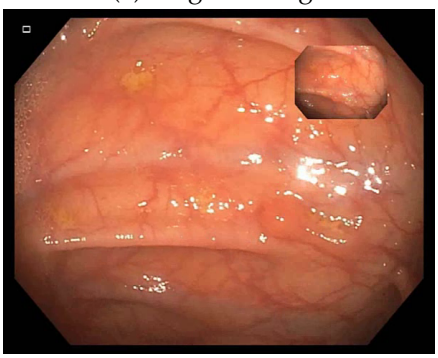
[3]http://docs.opencv.org/3.0-beta/modules/cudaimgproc/doc/histogram.html
[4]https://pillow.readthedocs.io/en/4.0.x/reference/Image.html
[5]https://pillow.readthedocs.io/en/4.0.x/reference/ImageEnhance.html



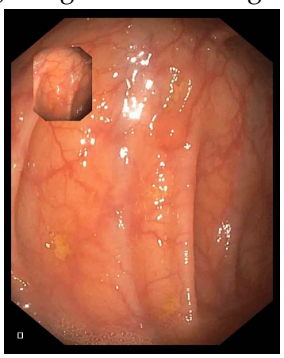
(a) Original image



(b) Image rotated 90 degrees



(c) Image rotated 180 degrees



(d) Image rotated 270 degrees

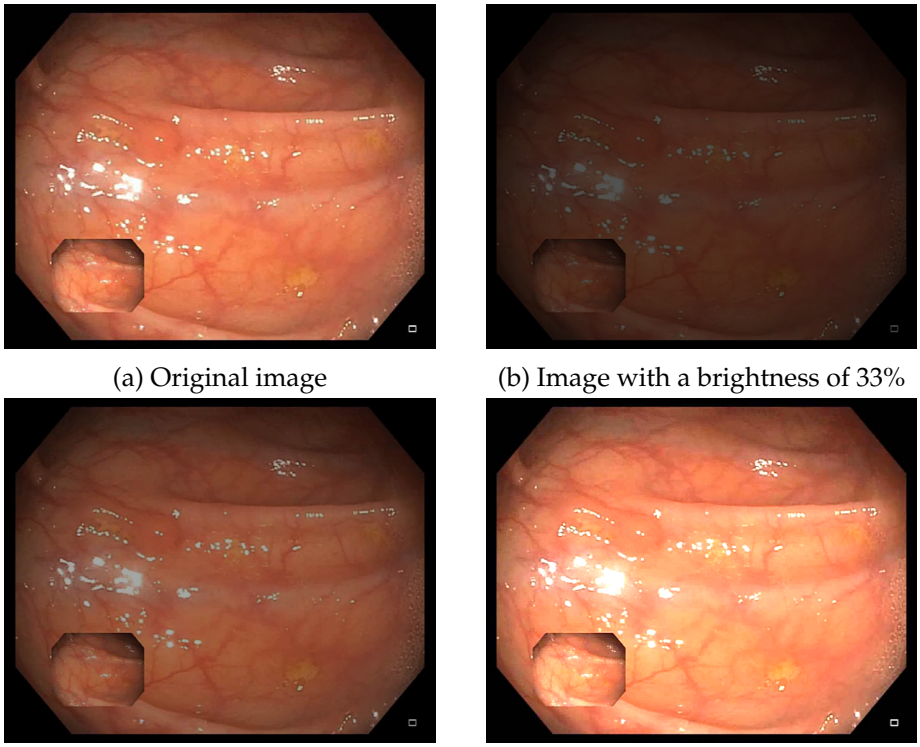
Figure 3.6: Rotation of an image counter-clockwise

3.3 TensorFlow

TensorFlow [1] is the continuation of DistBelief, Google Brains first machine learning system developed in 2011, and has been used internally at Google in products like Google Search, Google Photos, Google Maps, Google Translate, and many others. From the knowledge gained by DistBelief, Google developed TensorFlow, their second generation machine learning system, built for large-scale machine learning models. It supports an arbitrary number of GPUs, and can both be run locally and distributed, making it able to run on anything from a phone to a data center.

In TensorFlow, a computation is described by a direct graph which represents a dataflow computation. Each node represents an operation with one or more inputs and a name, for example "add" or "divide". A tensor is a multidimensional array, a datatype within TensorFlow. It is also the source of TensorFlows name. TensorFlow has become a popular neural network library, with over 7000 TensorFlow-related repositories on GitHub, and has been adopted by several large scale companies like Intel, eBay and Twitter [15]. It is available on Linux, Mac OS X and Windows, and provides a documented Python API.

TensorBoard is provided when installing TensorFlow. It is a visualization tool where it is possible to see detailed graphs and information about the model, making it easier to track problems and optimization possibilities. An example of such graphs can be seen in figure 3.8.



(a) Original image

(b) Image with a brightness of 33%

(c) Image with a brightness of 66%

(d) Image with a brightness of 133%

Figure 3.7: Images with different variations

3.4 TensorBox

The third block in figure 3.1a is TensorBox[6], developed by Russell Stewart. TensorBox is a framework on top of TensorFlow for object detection in images and has built-in support for several techniques for training neural networks.

We use a slightly modified version of TensorBox[7], where the training has been slightly modified to support TensorFlow 0.12.1. The evaluation was originally tied to a specific confidence percentage, but we wanted to be able to evaluate against multiple confidences in a single evaluation. We also wanted to retrieve the actual confidence percentages instead of just images with the polyp location annotated. We therefore modified it to produce additional classifications (true positives, false positives, true negatives and false negatives) and perform classification with multiple confidences simultaneously.

The reason we have used TensorBox is because it gave us the possibility to easily test various training optimizations and compare them seamlessly. It uses json files for both settings and inputs, which make modifications easy. This gives us the ability to create experiments with various data enhancement methods and parameters, train and evaluate them with

[6]https://github.com/TensorBox/TensorBox

[7]commit hash 7162368e204de8277f66a1224dc70c419986a64b

almost no modifications of the source code between experiments.

TensorBox is used for both training and evaluation. The training script sets up the graph in TensorFlow, with the techniques and choices specified in the settings file. The training consists of learning by iterating over the graph until the specified number of iterations have been reached, each time receiving feedback and adjusting weights accordingly. It saves the weights in the form of a checkpoint every X iterations, and a final checkpoint when it completes the training. Any of these checkpoints can be used to evaluate, giving us the ability to measure how the training evolves over time.

The evaluation script uses one of these checkpoints and the evaluation image set generated during model creation, classifying images into true positives, true negatives, false positives, and false negatives. It can also optionally produce the annoted images in seperate folders, one for each classification

TensorBox comes with support for different training techniques, neural networks and optimizers, some of which we will describe below.

Long short-term memory (LSTM)

LSTM [19] is a variant of Recurrent Neural Networks (RNN). RNNs are a type of neural networks that preserve earlier knowledge, but have issues when the amount of earlier knowledge increases.

LSTM, on the other hand, uses another technique where it decides the degree of information that is forgotten and gained for each node. This is done by having three or four gates on each node, calculating a number between 0 and 1. The number represents the degree of how much to remember, where 0 is to discard all information and 1 is to remember all information. By doing this, LSTM is suited for tasks where previous knowledge is important.

Inception

The most common method of increasing a neural networks performance is to increase its size, both in depth and width [54]. This is an easy way to improve performance, especially when given a high amount of quality training data. There are however two drawbacks to this, where the first is that increasing the size of the network often leads to an increase in the number of parameters. This makes the network more prone to overfitting, which can cause a major bottleneck, as manual intervention is likely to be required. The second is an increase in the required amount of computational resources, as the computational budget is always finite in practice.

Inception [54] was developed to counter these problems, and with a goal to improve performance. This was done by optimizing the neural network, instead of purely adding additional layers. The name Inception derives from a paper by Lin et al. called "Network In Network" [30], combined with a famous meme from the movie Inception.

Inception has been able to produce good results, being significantly more accurate even with 12 times fewer parameters as the ISLVRC 2012

winner, Krizhevsky et al. [24]. It is continually improved, where various versions have been released [55, 53].

Google has used an Inception based network to generate trained weights, which are published as checkpoints. TensorBox comes with support for the Inception architecture, which are able to use these checkpoints for further training.

Residual Networks (Resnet)
Deep neural networks have led to breakthroughs in image classifications, among others [17]. The depth of a neural network is of crucial importance, where the networks which have achieved the best results have had from 16 to 30 layers. In recent times, researchers have wondered if the future of neural networks consists of stacking ever more layers, but reaching higher depths have led to problems. One problem is that the accuracy will reach a point where it will degrade rapidly, which is not caused by overfitting. Adding more layers to a deep model leads to higher training error. This problem is called a degradation problem.

Resnet [17], developed by researchers at Microsoft, is a proposed solution to such problems. It is not unlike LSTMs, in that it is able to preserve knowledge, but uses a convolution processing layer instead of gates. It has been proven to produce good results [48, 9], and it won the ILSRCV 2015 image classification comptetition using 152 layers.

Rezoom
Rezoom is a training technique in TensorBox, which is explained in the source code of TensorBox as "Rezoom into a feature map at multiple interpolation points in a grid". There are no further explanations as to how Rezoom works.

Optimizers
An optimizer[8] in TensorFlow is a class providing support for computing gradients for losses and applying gradients to variables. TensorBox provides built-in funcionality for RMS, SGD and Adam.

Root Mean Square Propagation (RMS) is a method to adapt the learning rate for each of the parameters. This is done by dividing the learning rate for a weight by a running average on its recent gradients.

Adaptive Moment Estimation (Adam) is based on RMS. The main difference between them is that Adam includes both the gradients and their magnitude in the running average.

Stochastic Gradient Descent (SGD) is an implementation of the gradient descent algorithm. SGD tries to find minima or maxima by iteration. The use of SGD in neural networks is motivated by a high cost of back propagation over a full training set.

[8]https://www.tensorflow.org/api_guides/python/train

3.5 Summary

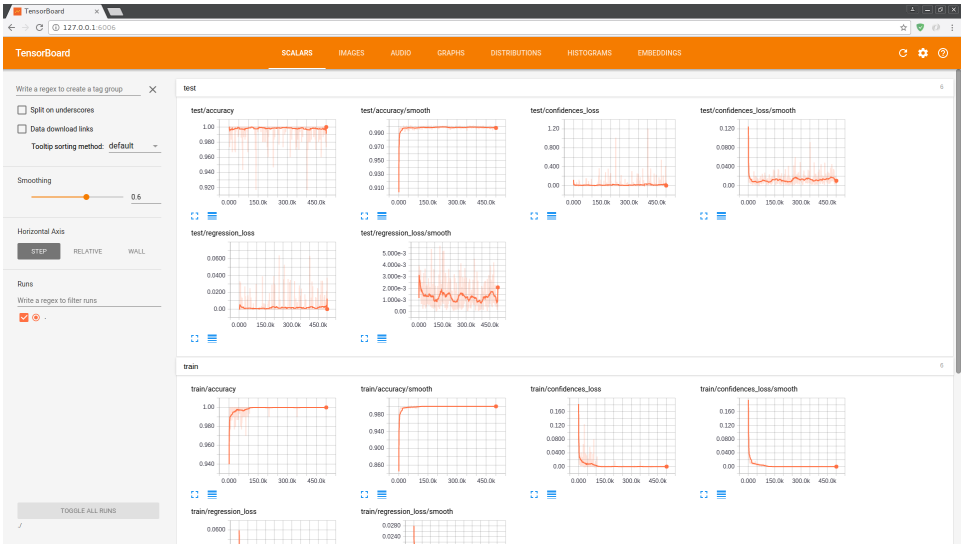
In this chapter, we have presented our polyp detection system consisting of a pipeline from annotated videos, extraction and data enhancement of frames, and training, to evaluation of videos.

The annotated videos, which is the ASU Mayo Dataset [56], is in the format of videos and annotation information. The videos needs to be extracted into frames and the polyp coordinates retrieved from the annotation information. This is done by the model creator, which extracts the frames using ffmpeg, scans corresponding tiff images for the polyp locations, and generates the json files for training and evaluation. This is also the step where data enhancement, consisting of any combination of contrast enhancement, masking reflections, rotation and different brightness variations, is applied.

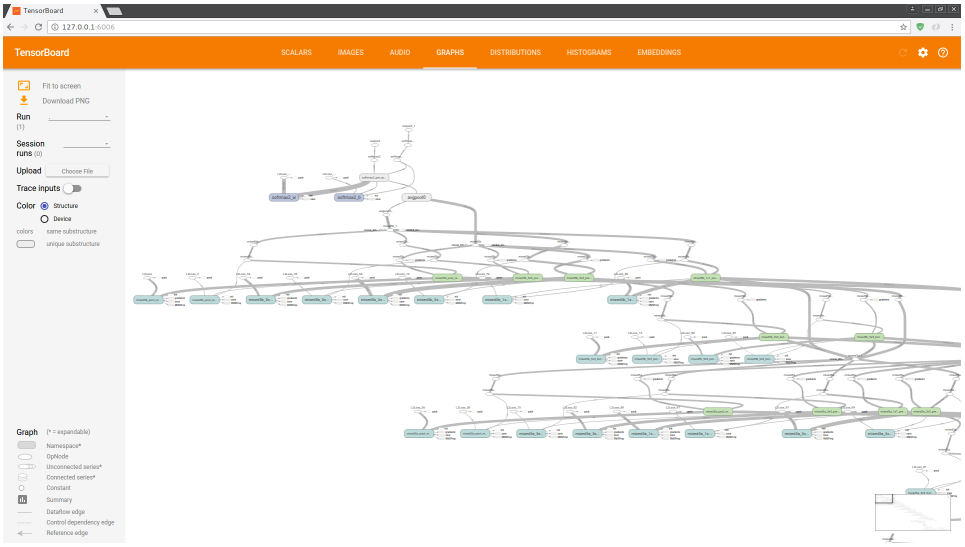
We have decided to use TensorBox, which is a neural network framework for object recognition using TensorFlow. It has support for various training techniques, neural networks, and optimizers out of the box, such as Rezoom, Inception, SGD, among others, and a json settings file for easy modifications. It is created for generic object detection, but we think it could also work well for polyp detection.

TensorFlow is a neural network library developed by Google and has gained popularity in the last couple of years among both researchers, developers and companies. It has over 7000 GitHub-related repositories, and has been adopted by companies such as Intel, eBay and Twitter. It can run on more or less any hardware configuration, from mobile devices to large data centers, has support for multiple GPUs, and can be run both locally and distributed. It outputs trained weights in the form of checkpoints, making it very easy to evaluate.

We have now described our polyp detection system. In the next chapter, we evaluate the performance of our approach.



(a) Graph with training and test information



(b) Snippet of the modell

Figure 3.8: Graphs generated in TensorBoard

Chapter 4

Experiments

We begin by describing our testbeds, data and evaluation method. We then conduct a data enhancement experiment, divided into sets of data enhancement methods, with a discussion based on the results for each step and a summary for each set. Then an experiment to optimize the training is performed in a similar manner as the previous experiment. Two smaller scale experiments follow, where the first is to determine the effect of additional training iterations and the second is an evaluation of our pre-trained system against a completely different dataset. We then discuss topics related to neural networks and our system. Finally, we summarize our findings.

4.1 Testbeds

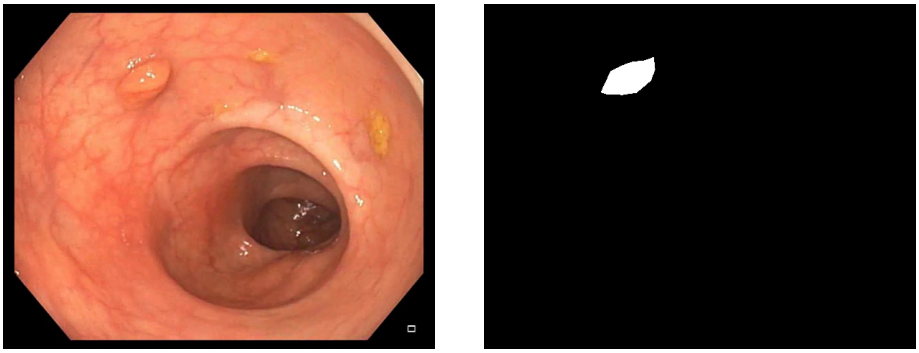
We use two different machines in order to process everything within the given time limit of the thesis. The complete list of software and hardware can be found in table 4.1. Machine 1 performs training and evaluation for split 1 through 4, while machine 2 does the same for split 5. Machine 1 was upgraded during the thesis from a NVIDIA GTX 1080 using driver version 357.26 to a NVIDIA GTX 1080 TI using driver version 378.13. Training and evaluation were performed using both cards and drivers on the same data, to confirm that the upgrade does not affect results in any way.

4.2 Data

We use the ASU Mayo Clinic polyp dataset [56], which is a publicly available annotated dataset of polyp videos as training and evaluation data. It consists of 20 videos, 10 of which contain polyps and 10 that do not. An overview of the videos that constitute the dataset can be found in tables 4.2 and 4.3. The videos are of various resolutions and durations, and come in the wmv format [32]. For each frame in a video, there is an associated tiff file containing the location of the polyp. The tiff files are used as a binary map, where a white area denotes a polyp. An example can seen in figure 4.1.

Category	Machine 1	Machine 2
Operating System	Ubuntu 14.04.5 LTS	Ubuntu 14.04.5 LTS
CUDA	8.0.61	8.0.61
cuDNN	5.1	5.1
NVIDIA Driver	357.26 / 378.13	357.26
TensorFlow	0.12.1	0.12.1
TensorBox	Modified version	Modified version
Python	2.7.6	2.7.6
OpenCV	3.2.0	3.2.0
CPU	Intel i5-4590 @ 3.30GHz	Intel i7-2600 @ 3.40GHz
Memory	16GB DDR3	16GB DDR3
GPU	NVIDIA GTX 1080 / NVIDIA GTX 1080 TI	NVIDIA GTX TITAN

Table 4.1: Software and hardware configuration of the testbeds



(a) The video frame containing a polyp

(b) The tiff file showing the ground truth

Figure 4.1: Polyp annotation example [56]

4.3 Evaluation method and metrics

We use 5-fold cross validation to assess how our results will perform on an independent dataset. If a common partitioning had been used, where 70% of the videos are used for training and 30% for evaluation, there would be no guarantees that a different partitioning would produce similar results.

K-fold cross validation [25] divides the dataset into k number of equal sized partitions where the cross validation have to be performed k number of times. Each partition is then used as validation in a single cross validation, while the other partitions are used as training data. The result from each cross validation is then averaged to produce a single estimation. This ensures that the results are not based on a single partitioning, which could be a deviation, but rather an average over all partitionings.

In our case, we divide the 10 videos into 5 seperate partitions, so each cross validation we run will use 8 videos for training and 2 for evaluation. A benefit of this is that all data is part of both training and validation, while being part of validation exactly one time. In addition, all 10 videos not

Video name	Duration in minutes	Number of frames	Resolution
wp_2	00:10	324	1920 x 1080
wp_4	00:30	910	1920 x 1080
wp_24	00:17	519	720 x 480
wp_49	00:16	501	856 x 480
wp_52	00:36	1106	856 x 480
wp_61	00:11	339	1920 x 1080
wp_66	00:13	418	856 x 480
wp_68	00:08	259	1920 x 1080
wp_69	00:20	616	1920 x 1080
wp_70	00:13	410	856 x 480

Table 4.2: Overview of videos containing polyps

Video name	Duration in minutes	Number of frames	Resolution
np_5	00:22	682	720 x 480
np_6	00:27	838	720 x 480
np_7	00:25	769	720 x 480
np_8	00:23	712	720 x 480
np_9	01:01	1843	720 x 480
np_10	01:04	1925	720 x 480
np_11	00:51	1550	720 x 480
np_12	00:58	1740	720 x 480
np_13	01:00	1802	720 x 480
np_14	00:54	1639	720 x 480

Table 4.3: Overview of videos not containing polyps

containing polyps are used as part of the evaluation set for every cross validation. Each combination of one partition for evaluation, and the other four for training, constitutes a split. An overview of our splits is given in figure 4.2.

Figure 4.2: Overview of the splits

Short name	**Full name**	**Description**
TP	True Positives	Correctly classified as containing polyps
TN	True Negative	Correctly classified as not containing polyps
FP	False Positive	Wrongly classified as containing polyps
FN	False Negative	Wrongly classified as not containing polyps

Table 4.4: Short name, full name and description of each classification

The output of the evaluation are the number of TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives), which are explained in table 4.4. These are further used to calculate recall, precision and F1-score.

Recall is the probability of correct classification of a sample, either positive or negative. Positive recall, also called sensitivity or true positive rate, is the percentage of correctly detected polyps. Negative recall, also called specificity or true negative rate, is the percentage of correctly classified negative samples.

$$Positive\ recall\ (Sensitivity) = \frac{TP}{TP + FN}$$

$$Negative\ recall\ (Specificity) = \frac{TN}{TN + FP}$$

Precision is the precision of the detection. The higher precision, the more precise the detection is. Positive precision, also called positive predictive value, is the percentage of positive classifications that are correct. Negative precision, also called negative predictive value, is the percentage of negative classifications that are correct.

$$Positive\ precision = \frac{TP}{TP + FP}$$

$$Negative\ precision = \frac{TN}{TN + FN}$$

F_1-score is a weighted average between the precision and recall which gives an idea of the overall performance of the system.

$$Positive\ F_1 = 2 \times \frac{Positive\ precision \times Positive\ recall}{Positive\ precision + Positive\ recall}$$

$$Negative\ F_1 = 2 \times \frac{Negative\ precision \times Negative\ recall}{Negative\ precision + Negative\ recall}$$

To get the datasets total scores, the different metrics are weighted based on the number of positive and negative samples. In the following formulas, PR is the positive recall, NR is the negative recall, PP is the positive precision, NP is the negative precision, PF1 is the positive F1-score, and NF1-score is the negative F1-score.

$$Weighted\ recall = \frac{(PR \times Positive\ samples) + (NR \times Negative\ samples)}{Total\ samples}$$

$$Weighted\ precision = \frac{(PP \times Positive\ samples) + (NP \times Negative\ samples)}{Total\ samples}$$

$$Weighted\ F_1 - score = \frac{(PF1 \times Positive\ samples) + (NF1 \times Negative\ samples)}{Total\ samples}$$

4.4 Data enhancement

In this section, we will present and discuss the results from the data enhancement experiment where we use various data enhancement methods. The experiments are grouped into four sets, where each set consists of all combinations of that sets data enhancement methods. The first two sets are different data enhancement methods, artificially increasing the dataset and image preprocessing, while the last two are combinations of the first ones. In each experiment, we present the results in the form of a table and a graph, and finish with a discussion. The results we present is an average of the 5 splits in the 5-fold cross-validation, where each data enhancement method have been applied to each split.

First, a non-preprocessed version is compared against the majority class baseline to determine that the system is able to make good decisions. The majority class baseline is defined as all classifications being either positive or negative, depending on the balance of the dataset. In our case, we have a negative majority, which means the baseline has 0 TPs and 0 FPs. This gives our baseline a weighted recall of 94.74, a weighted precision of 95.04 and a weighted F1-score of 92.19. The baseline distribution between the negative and positive samples in the individual splits can be seen in figure 4.3, and the baseline values for the individual splits, as well as the average, can be seen in figure 4.4. Since the evaluation set always contains the 10 non-polyp videos, and only 2 polyp-videos, the number of negative samples dominate the number of positive samples, making the baseline difficult to best. Another possible baseline is a random baseline, where the classifications are 50-50 true and false, which would produce lower scores due to our distribution between negative and positive samples.

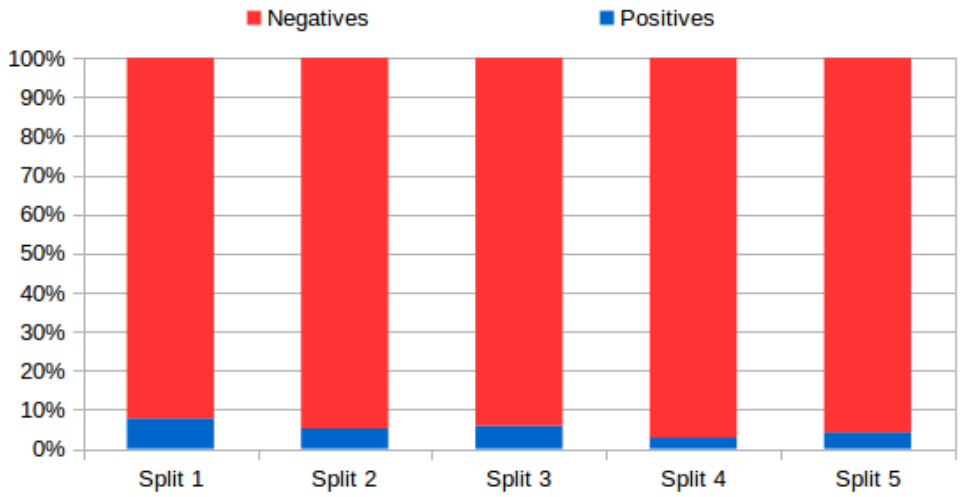


Figure 4.3: Distribution between positive and negative samples in the individual splits

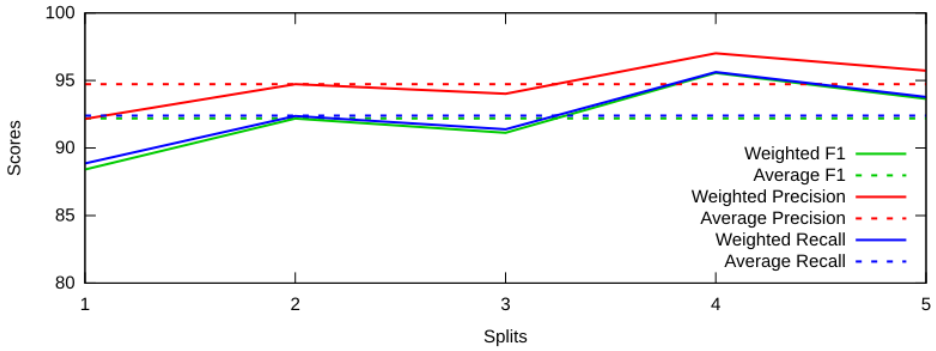


Figure 4.4: Majority class baseline overview for the individual splits

The following experiments are conducted to examine the effects the different data enhancement methods have on the polyp detection rate. The different data enhancement methods are compared against the non-preprocessed version, and previous methods, to examine their effect. We also want to find out if combining the methods can result in collective benefits, where the effect of one method is magnified by the effect of another. If this is the case, methods could be dependent on other methods to produce optimal results, and as such, some combinations could be greater than the sum of their parts. All experiments are performed with the same set of parameters, shown in figure 4.5, where the only difference being the training and evaluation data.

Each experiment is trained for 500 000 training iterations, where a training iteration is defined as one round of a feedback and adjustment loop. For evaluation, 90% confidence is used, which means that the system has to have a 90% or higher certainty that the image contains a polyp, before it is classified as positive. Training iterations and confidences are further discussed in section 4.8.3

Table 4.5 contains an overview of the data enhancement methods full names and their short names which will be used when multiple data enhancement methods are combined, for instance RBM which denotes the combination of rotation, brightness variations and masking reflections. At the end, we will summarize our findings and discuss the effects of the different methods.

Short name	**Full name**
NP	Non-preprocessed
R	Rotation
B	Brightness variations
M	Masking reflections
C	Contrast enhancement

Table 4.5: Short names and full names for all data enhancement methods

```
"logging": {
    "display_iter": 50,
    "save_iter": 100000
},
"solver": {
    "opt": "RMS",
    "use_jitter": false,
    "rnd_seed": 1,
    "epsilon": 0.00001,
    "learning_rate": 0.001,
    "learning_rate_step": 33000,
    "hungarian_iou": 0.25,
    "weights": "",
    "head_weights": [1.0, 0.1],
    "max_iter": 501000
},
"use_lstm": false,
"use_rezoom": true,
"biggest_box_px": 10000,
"rezoom_change_loss": "center",
"rezoom_w_coords": [-0.25, 0.25],
"rezoom_h_coords": [-0.25, 0.25],
"reregress": true,
"focus_size": 1.8,
"early_feat_channels": 256,
"later_feat_channels": 832,
"avg_pool_size": 5,
"num_lstm_layers": 2,
"image_width": 640,
"image_height": 480,
"grid_height": 15,
"grid_width": 20,
"batch_size": 1,
"region_size": 32,
"clip_norm": 1.0,
"lstm_size": 500,
"deconv": false,
"num_classes": 2,
"rnn_len": 1
```

Figure 4.5: The settings file used during experiments

4.4.1 Non-preprocessed data

We started by using non-preprocessed data to see how the system performed with no data enhancement method applied, giving us a basis for comparison with the data enhancement methods. The results can be seen in table 4.6 and figure 4.6.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
NP											
100k	29.08%	28.50	29.08	27.52	96.76%	96.19	96.76	96.46	**92.95**	**92.77**	**93.27**
200k	26.60%	31.27	26.60	28.03	98.02%	96.12	98.01	97.05	**93.51**	**92.80**	**94.36**
300k	26.16%	31.50	26.16	27.73	98.15%	96.10	98.15	97.11	**93.56**	**92.81**	**94.47**
400k	26.31%	31.64	26.31	27.85	98.15%	96.11	98.15	97.11	**93.57**	**92.83**	**94.47**
500k	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
Base	00.00%	00.00	00.00	00.00	100.0%	94.73	100.0	97.29	**92.19**	**95.04**	**94.74**

Table 4.6: Results using NP

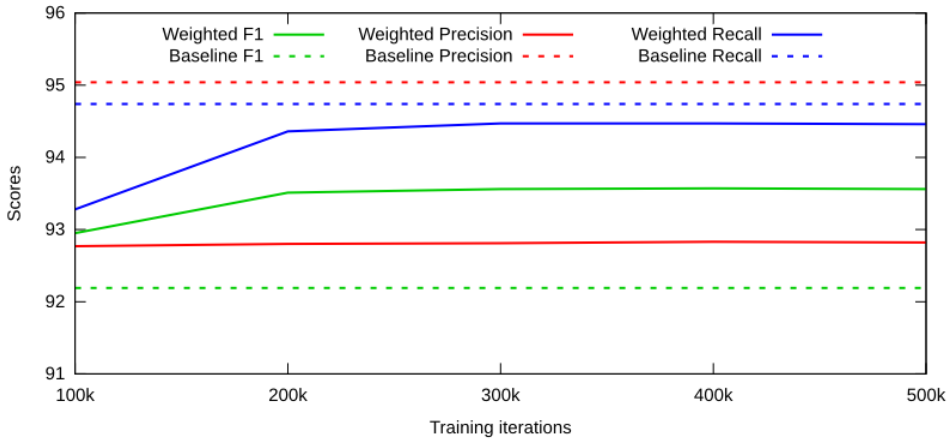


Figure 4.6: Weighted scores compared to the majority class baseline using NP

From the results, we can see that we get an increase in scores from 100k to 300k training iterations, but after 300k there are close to no gains. Even though the results stabilizes after 300k, there are minor variations up until 500k iterations. This is the reason why we use 500k iterations as the basis for comparisons with the baseline and the different data enhancement methods.

The system with non-preprocessed data achieves a weighted recall of 94.46, which means that 94.46% of the systems classifications were correct. The achieved weighted precision was 92.82, which means that 92.82% of the images the system classified as positive or negative were correct. The weighted F1-score is 93.56, which is the overall score of the non-preprocessed data.

Compared to the baseline, the system is able to achieve a higher F1-score, but a lower precision and recall. To beat the precision and recall of the baseline, the combined number of FPs and FNs needs to be lower than the number of positive samples. By increasing the number of detected polyps, additional FPs are also found as a natural consequence [40]. Since the number of positive samples in our dataset is so low, detecting a few TPs often also leads to detecting many FPs. With a F1-score 1.37 better than the baseline, the system with no data enhancements or modifications has an overall better performance.

The results in this section are used as a comparison in the following sections, where we experiment with the different data enhancement methods. For each experiment, we focus on the increase or decrease in positive and negative recall, as well as the weighted F1-score.

4.4.2 Rotation and brightness variations

The first set of the data enhancement methods is about artificially increasing the dataset, discussed in section 3.1.1, utilizing rotation and brightness variations. This is done to increase the number of polyp samples the system can learn from, while also producing additional variants of the polyps that can occur in the GI tract. By doing this, we hope to see an increase in detected polyps while keeping the number of FPs relatively stable.

4.4.2.1 Rotation

For this experiment, all polyp images for training has been rotated 90°, 180° and 270°, in addition to the original, as outlined in section 3.2.4. This was done to see if adding additional samples of polyps in different angles would increase the polyp detection rate. The results can be seen in table 4.7 and figure 4.7.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
					R						
100k	36.91%	34.48	36.91	34.02	96.87%	96.63	96.87	96.73	**93.46**	**93.35**	**93.82**
200k	32.29%	40.04	32.29	35.29	97.80%	96.39	97.80	97.08	**93.88**	**93.41**	**94.44**
300k	32.21%	37.32	32.21	34.25	97.44%	96.37	97.44	96.90	**93.66**	**93.28**	**94.10**
400k	31.82%	36.55	31.81	33.70	97.43%	96.35	97.43	96.66	**93.62**	**93.23**	**94.08**
500k	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**

Table 4.7: Results using R

As can be seen in the graph and table, the results are unstable until 300k training iterations have been completed, similar to the non-preprocessed version in the previous section. However, unlike the non-preprocessed

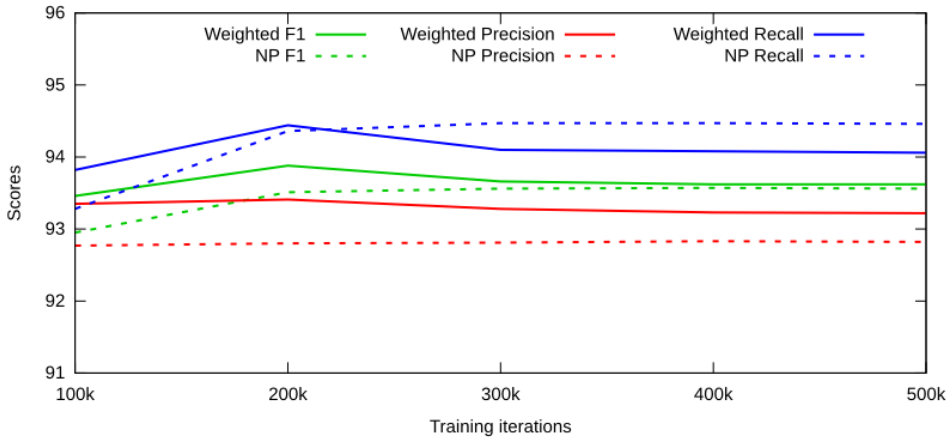


Figure 4.7: Weighted scores compared to NP using R

version, the results do not continually increase as additional training iterations are performed, but instead drops from 200k to 300k training iterations before stabilizing. Regardless if the stabilizing phase increases or decreases the results, it is unstable and thus not consistent enough for comparison. The results support our decision from the previous section to use 500k training iterations as the basis for comparison.

The results show a decrease in weighted recall of 0.40%, an increase in weighted precision of 0.40% and an increase in weighted F1-score of 0.06% compared to the non-preprocessed version. The positive recall and positive precision both have an increase of about 5%. From this, we can see that the effect of rotation is an increase in detected polyps, and of the images classified as containing a polyp, a higher percentage do indeed contain a polyp. On the other hand, negative recall has decreased by 0.73%, which means more FPs as well. The reason the number of FPs can increase at the same time as the positive precision increases, is because of the increase in positive recall. A higher percentage of the positive classifications can be correct, but because of the higher number of positive classifications, there can also be more FPs. The decrease of only 0.73% in negative recall is nearly enough to negate the increase of 5.51% in positive recall, confirming the issue of the high number of negative samples discussed in previous sections.

Overall, rotation is able to increase the F1-score by 0.06%. Even though the overall increase is small, the positive recall and precision saw a noticable improvement. The main effect of rotation is thus the increase in detected polyps, making rotation valuable in itself and also interesting for future combinations.

4.4.2.2 Brightness variations

For this experiment, all polyp images for training has had their brightness level altered to 33%, 66% and 133%, in addition to the original, as outlined in section 3.2.5. This was done to see if adding samples of polyps with different brightness levels would increase the polyp detection rate. The results can be seen in table 4.8 and figure 4.8.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
B											
100k	22.39%	16.31	22.39	18.27	94.33%	95.70	94.34	94.99	**91.06**	**91.66**	**90.60**
200k	20.16%	19.75	20.16	19.43	95.93%	95.60	95.93	95.75	**91.79**	**91.69**	**91.96**
300k	20.66%	20.08	20.66	19.80	96.00%	95.62	96.00	95.80	**91.85**	**91.72**	**92.05**
400k	20.68%	20.18	20.68	19.83	96.00%	95.62	96.00	95.80	**91.84**	**91.72**	**92.05**
500k	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**

Table 4.8: Results using B

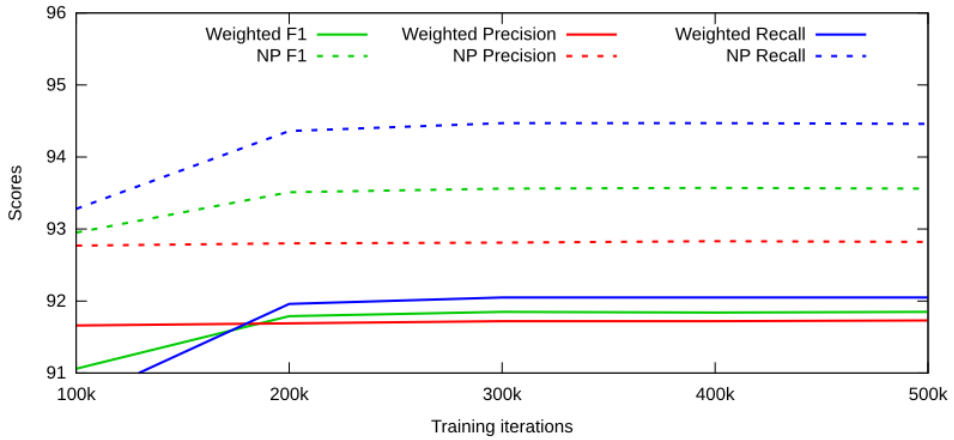


Figure 4.8: Weighted scores compared to NP using B

From the results, we can see a decrease in weighted recall of 2.41%, a decrease in weighted precision of 1.19% and a decrease in weighted F1-score of 1.71% compared to the non-preprocessed version. It is unable to improve a single metric compared to both rotation and non-preprocessed. Both negative and positive recall have decreased, which means a lower number of detected polyps and more FPs as well. The F1-score decreased by 1.71%, making the overall performance noticable lower.

We believe the negative effect of brightness variations is due to some confusion created during the training, where the different brightness levels impair the systems ability to gain correct knowledge. When the systems knowledge of what characterizes a polyp is poor, it will be unable to

recognize polyps, and also wrongly detect other objects as polyps. We believe this is the main reason for the decrease in both positive and negative recall.

With a F1-score 1.71% lower and unable to increase a single metric, there is no benefit in using brightness variations by itself. We will still include it in further experiments to investigate if it is able to lead to a positive effect when combined with other data enhancement methods.

4.4.2.3 Rotation and brightness variations

For this experiment, we combined rotation and brightness variations, which results in 16 different versions of each image that contains a polyp in the training data. This was done to see the effect combining the two different data enhancement methods have, and if they are able to produce a better score when combined than each of them individually. The results can be seen in table 4.9 and figure 4.9.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RB											
100k	26.90%	27.70	26.90	26.78	96.69%	96.01	96.69	96.34	**92.77**	**92.57**	**93.06**
200k	28.33%	26.06	28.33	26.55	96.66%	96.18	96.66	96.41	**92.85**	**92.58**	**93.19**
300k	27.80%	25.17	27.80	25.85	96.62%	96.18	96.62	96.39	**92.80**	**92.53**	**93.16**
400k	27.63%	24.99	27.63	25.73	96.61%	96.16	96.60	96.37	**92.78**	**92.51**	**93.13**
500k	27.65%	25.09	27.66	25.79	96.60%	96.17	96.60	96.38	**92.78**	**92.52**	**93.13**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**

Table 4.9: Results using RB

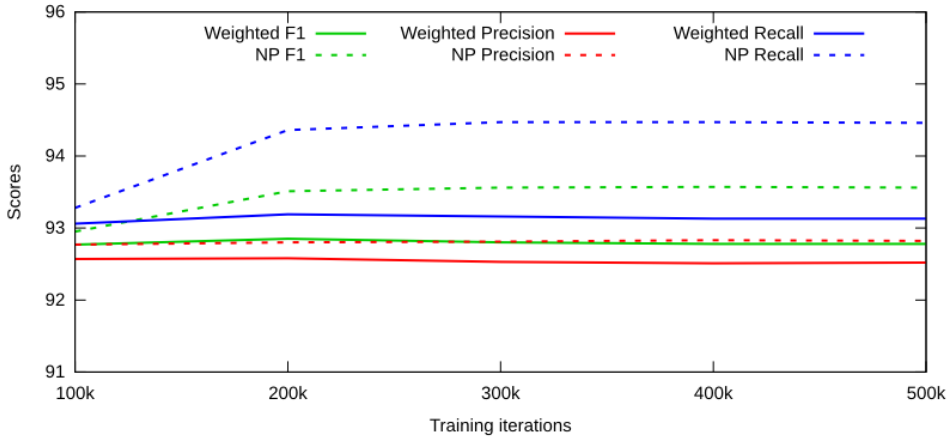


Figure 4.9: Weighted scores compared to NP using RB

We can see a decrease in weighted recall of 1.33%, a decrease in weighted precision of 0.30% and a decrease in weighted F1-score of 0.78% compared to the non-preprocessed version. If we compare the combined results against the individual results of rotation and brightness variations, RBs results can be seen as being the average of the two or close to it. The effect of both methods are visible, but they are unable to affect each other in a way that improves or aggravates the quality of the polyp detection. From this, we can see that combining rotation and brightness does not have a greater sum than the sum of its parts. Overall, the decrease in F1-score of 0.78% indicates that the performance is lower than the non-preprocessed version, as brightness variations impairs the performance more than rotation improves it.

4.4.2.4 Summary of rotation and brightness variations

We found that rotation improves the systems ability to find additional polyps, but also results in more FPs. The 5.51% increase in positive recall outweighs the decrease of 0.73% in negative recall, producing a slightly higher weighted F1-score of 93.62%.

Brightness variations on the other hand lowers the number of detected polyps, and also increases the number of FPs. With a weighted F1-score of 91.85%, and improvements in neither positive nor negative classifications, brightness is the only data enhancement method unable to improve any metric thus far.

The combination of rotation and brightness variations produces scores close to an average of the two. With a weighted F1-score 0.72% lower than the non-preprocessed version, it seems that rotation and brightness variations are unable to improve each other, instead their effects are averaged when combined.

4.4.3 Masking reflections and contrast enhancement

The second set of the data enhancement methods is about improving the existing data by masking reflections and contrast enhancement, as discussed in sections 3.1.2 and 3.1.3, respectively. The reason for doing this is to remove weaknesses in the dataset, such as reflections of light and low visibility of polyps in low-contrast areas, which is a common problem in colonoscopy videos. By doing this, we hope to lower the number of FPs while also increase the number of detected polyps.

4.4.3.1 Masking reflections

For this experiment, all images for both training and evaluation have had their reflections masked. This was done to see if the system is able to reduce the number of FPs, as the reflections can be mistaken as polyps, as discussed in section 3.2.2. The results can be seen in table 4.10 and figure 4.10.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
					M						
100k	25.80%	22.42	25.80	23.10	95.43%	96.04	95.43	95.72	**92.15**	**92.48**	**91.91**
200k	22.19%	25.10	22.19	22.98	97.81%	96.00	97.81	96.88	**93.22**	**92.52**	**94.03**
300k	22.39%	24.88	22.39	22.98	97.78%	96.01	97.78	96.88	**93.22**	**92.52**	**94.01**
400k	22.29%	24.76	22.29	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**
500k	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**

Table 4.10: Results using M

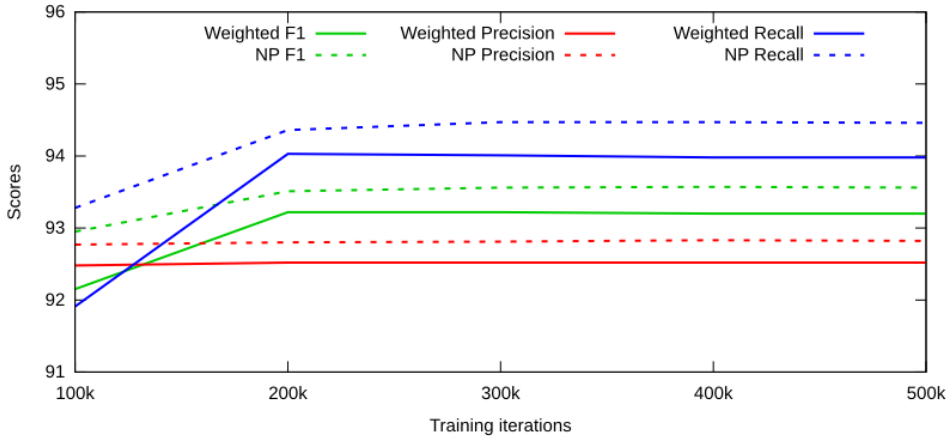


Figure 4.10: Weighted scores compared to NP using M

The results show a decrease in weighted recall of 0.48%, a decrease in weighted precision of 0.30% and a decrease in weighted F1-score of 0.36% compared to the non-preprocessed version. Similar to when using brightness variations, every metric has decreased. However, when comparing the results from individual splits, masking reflections have radically varying effects. The best split shows an increase in positive recall of 5% and negative recall of 1.5%, while the worst split shows a decrease in positive recall of 19% and negative recall of 1.3%. While brightness variations have an equal negative effect on each split, masking reflections is more of a hit and miss, where it can result in everything between a good improvement to a great loss in performance.

Masking of reflections is a challenge as the characteristics of images taken inside the GI tract changes radically from image to image. Reflections come in many forms, sizes and colors, and correctly masking every one of them is complicated. Depending on the reflection in the image, the results vary, which could be the reason for our results. A more advanced version of masking reflections could improve results in difficult images and maintain the improvements we see in the best split.

Masking reflections is unable to produce consistent results across the splits, but by combining it with other data enhancement methods, the video characteristics may change enough for masking reflections to overall have a positive effect.

4.4.3.2 Contrast enhancement

For this experiment, all images for training and evaluation have had their contrast enhanced by using CLAHE [68], as discussed in section 3.2.3. This was done to see if by enhancing the contrast, additional polyps could be detected in low-contrast areas. The results can be seen in table 4.11 and figure 4.11.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
C											
100k	24.67%	25.36	24.67	24.39	97.19%	96.12	97.19	96.64	**93.04**	**92.57**	**93.60**
200k	24.42%	26.95	24.42	24.89	97.57%	96.13	97.57	96.84	**93.26**	**92.66**	**93.96**
300k	24.44%	27.41	24.44	25.11	97.66%	96.14	97.66	96.88	**93.31**	**92.69**	**94.04**
400k	24.60%	27.26	24.60	25.14	97.60%	96.14	97.60	96.86	**93.28**	**92.69**	**94.00**
500k	24.62%	27.26	24.62	25.14	97.60%	96.15	97.60	96.86	**93.28**	**92.69**	**93.99**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
M	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**

Table 4.11: Results using C

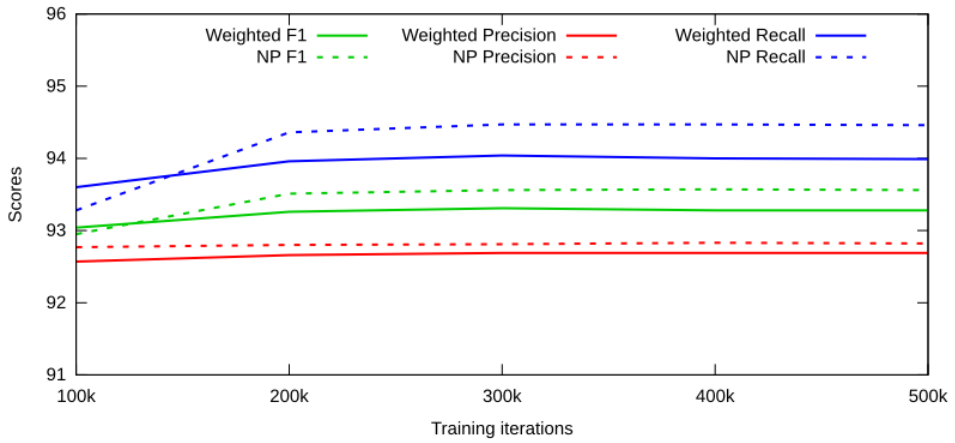


Figure 4.11: Weighted scores compared to NP using C

We can see a decrease in weighted recall of 0.47%, a decrease in weighted precision of 0.13% and a decrease in weighted F1-score of 0.28% compared to the non-preprocessed version. The effect of contrast enhancement varies slightly, not unlike what we have seen for previous data enhancement methods, with minor gains and losses for different splits. However, unlike brightness variations, the effect is usually a

decrease in one, and an increase in the other for TP and TN, with the decrease generally being bigger than the increase. This results in an overall decrease of both positive and negative recall, making the weighted F1-score slightly lower.

Like in masking reflections, the effect of contrast enhancement depends on the characteristics of the image. For polyps with a less defined outline, contrast enhancement has a limited effect as there are no outline to enhance. For more defined polyps, contrast enhancement seems to work well, with a gain in positive recall in most splits. In general, other structures in the image are contrast enhanced as well, and if their shape resembles that of a polyp, it becomes more likely that they will be mistaken as such, producing added FPs.

Compared to brightness variations and masking reflections, contrast enhancement has a higher overall performance, but is unable to beat the non-preprocessed version by itself.

4.4.3.3 Masking reflections and contrast enhancement

In this experiment, we combined masking reflections and contrast enhancement. The purpose of both data enhancement methods are to improve the quality of the input data, and with both being applied on the same images, there could be mutual gains. The results can be seen in table 4.12 and figure 4.12.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
MC											
100k	27.42%	27.46	27.42	26.72	96.72%	96.13	96.72	96.41	**92.91**	**92.73**	**93.18**
200k	24.62%	28.09	24.61	25.95	97.87%	96.11	97.87	96.97	**93.44**	**92.74**	**94.21**
300k	25.17%	27.87	25.17	26.13	97.83%	96.13	97.83	96.96	**93.44**	**92.76**	**94.19**
400k	24.85%	27.89	24.85	25.96	97.88%	96.12	97.88	96.98	**93.45**	**92.75**	**94.22**
500k	24.82%	27.84	24.82	25.92	97.87%	96.12	97.88	96.98	**93.45**	**92.74**	**94.22**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
M	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**
C	24.62%	27.26	24.62	25.14	97.60%	96.15	97.60	96.86	**93.28**	**92.69**	**93.99**

Table 4.12: Results using MC

We can see a decrease in weighted recall of 0.24%, a decrease in weighted precision of 0.08% and a decrease in weighted F1-score of 0.11% compared to the non-preprocessed version. The combination produces better scores than each of the methods individually, unlike the combination of rotation and brightness. We assume that the reason behind the improvements when combined is because of the reflections in the images being masked before contrast enhancement is applied. With reflections in the images, contrast enhancement will also highlight the reflections while enhancing the image. By masking the reflections beforehand, there are a much lower number of reflections that the contrast enhancement

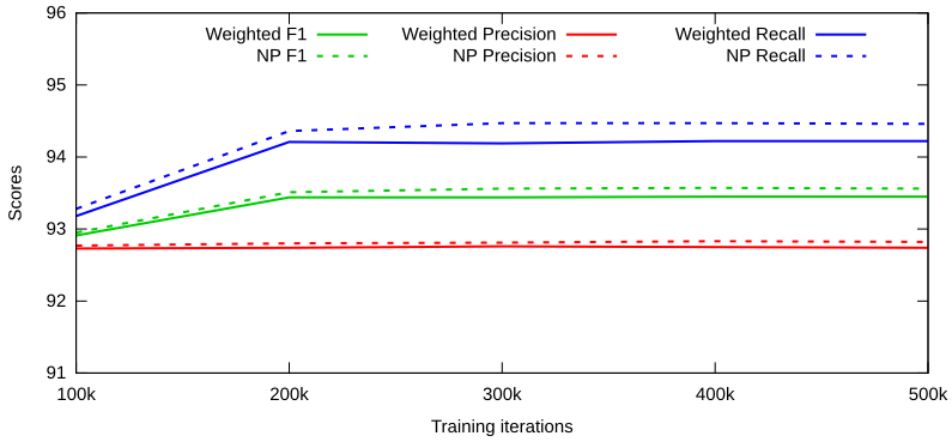


Figure 4.12: Weighted scores compared to NP version using MC

will highlight, increasing the resulting quality of the data enhancement method. Thus, it can be said that the combination of masking reflections and contrast enhancement has a greater sum than the sum of its parts.

4.4.3.4 Summary of masking reflections and contrast enhancement

We found that masking reflections had a small negative impact on the system, with a weighted F1-score of 93.20, 0.36% lower than non-preprocessed. The negative impact is quite small, but the performance is worse in both positive and negative recall and precision. Masking reflections produces varying results depending on the video, from a good improvement to a major loss in performance. As such, masking reflections can be said to be hit and miss.

Contrast enhancement has less of a negative impact on the system, with a weighted F1-score of 93.28, 0.28% lower than non-preprocessed. Similar to masking reflections, contrast enhancement is dependent on the videos, leading to some variations in the results. However, the variations are less extreme, making the results more predictable.

The results of the combination have a weighted F1-score of 93.45, 0.11% lower than the non-preprocessed. Thus, the combination of masking reflections and contrast enhancement is able to produce better results combined than on their own.

4.4.4 Rotation, brightness variations, masking reflections and contrast enhancement

The third set of the data enhancement methods is about combining the first two sets. Each of the two sets have had a different purpose, where the first was to artificially increase the dataset size while the second was to improve the image quality. By combining these two sets, we hope to improve the results by combining an increase in both quantity and quality.

4.4.4.1 Rotation, brightness variations and masking reflections

For this experiment, rotation, brightness variations and masking reflections have been combined. The results can be seen in table 4.13 and figure 4.13.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	Weighted F1	Weighted Precision	Weighted Recall
RBM											
100k	29.53%	27.62	29.53	28.21	96.91%	96.26	96.91	96.58	**93.11**	**92.76**	**93.50**
200k	29.87%	28.68	29.87	29.13	96.03%	96.22	96.03	96.12	**92.70**	**92.75**	**92.68**
300k	29.84%	25.42	29.84	27.20	95.43%	96.21	95.43	95.81	**92.31**	**92.57**	**92.12**
400k	30.31%	24.96	30.31	27.15	95.30%	96.24	95.15	95.76	**92.26**	**92.57**	**92.03**
500k	30.24%	24.86	30.24	27.06	95.29%	96.23	95.29	95.75	**92.25**	**92.56**	**92.01**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
RB	27.65%	25.09	27.66	25.79	96.60%	96.17	96.60	96.38	**92.78**	**92.52**	**93.13**
M	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**

Table 4.13: Results using RBM

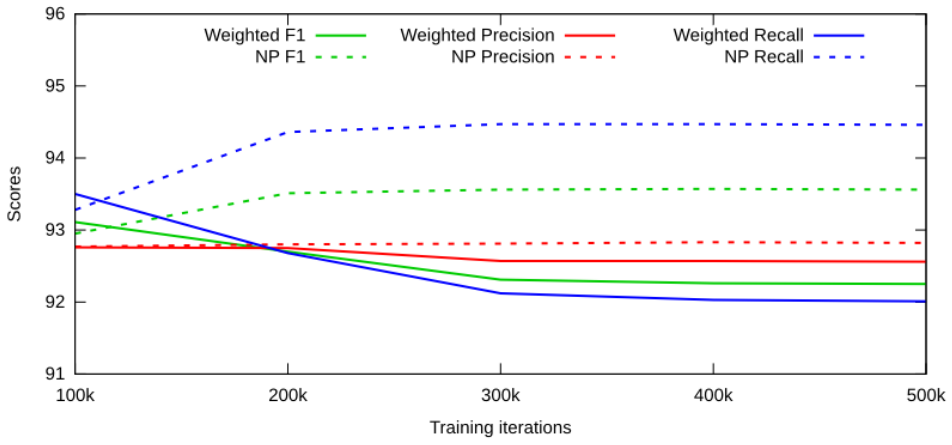


Figure 4.13: Weighted scores compared to NP using RBM

From the results, we can see a decrease in weighted recall of 2.45%, a decrease in weighted precision of 0.26% and a decrease in weighted F1-score of 1.31% compared to the non-preprocessed version. The positive recall is 30.24%, up 7.98% from masking reflections and up 2.59% from RB, which RBM is the combination of. On the other hand, the negative recall is 95.29%, down 2.47% from masking reflections and down 1.31% from RB. From this, we can see that RBM produces results that are both higher and lower than the data enhancement methods' individual results. These results are hard to explain only by looking at the individual results, but we assume they are due to mutual effects of the combination.

While the percentwise increase in positive recall is higher than the decrease in negative recall, the increase in TPs are lower than the decrease in FPs. The effect of this is a lower overall performance with a F1-score 0.95% lower than masking reflections and 0.53% lower than RB. In a scenario where the positive recall is focused, the higher positive recall may be worth the trade-off in negative recall.

4.4.4.2 Rotation, brightness variations and contrast enhancement

For this experiment, rotation, brightness variations and contrast enhancement have been combined. The results can be seen in table 4.14 and figure 4.14.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RBC											
100k	31.84%	22.79	31.84	24.56	94.47%	96.41	94.47	95.37	**91.73**	**92.54**	**91.41**
200k	31.74%	23.96	31.74	26.79	94.44%	96.34	94.44	95.37	**91.87**	**92.59**	**91.35**
300k	31.33%	21.07	31.33	24.68	93.83%	96.29	93.82	95.03	**91.46**	**92.42**	**90.74**
400k	31.20%	20.95	31.20	24.50	93.83%	96.28	93.83	95.02	**91.45**	**92.41**	**90.73**
500k	31.12%	20.86	31.12	24.40	93.80%	96.28	93.80	95.01	**91.43**	**92.40**	**90.70**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
RB	27.65%	25.09	27.66	25.79	96.60%	96.17	96.60	96.38	**92.78**	**92.52**	**93.13**
M	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**
C	24.62%	27.26	24.62	25.14	97.60%	96.15	97.60	96.86	**93.28**	**92.69**	**93.99**
RBM	30.24%	24.86	30.24	27.06	95.29%	96.23	95.29	95.75	**92.25**	**92.56**	**92.01**

Table 4.14: Results using RBC

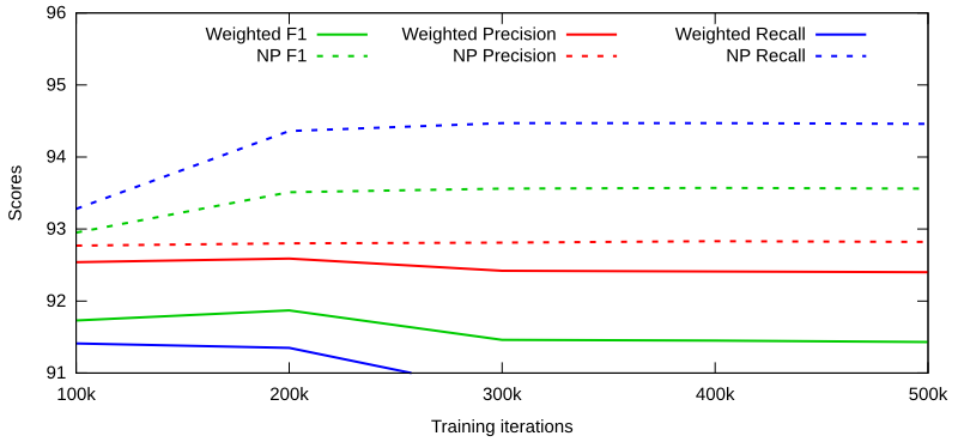


Figure 4.14: Weighted scores compared to NP using RBC

The results show a decrease in weighted recall of 3.76%, a decrease in weighted precision of 0.42% and a decrease in weighted F1-score of 2.13%

compared to the non-preprocessed version. This combination follows a similar pattern as RBM, only to an even larger extent. From RBM, the positive recall has increased by 0.88% to 31.12% and the negative recall decreased by 1.49% to 93.80%. We assume that the increase and decrease are due to contrast enhancement having similar increases and decreases when compared to masking reflections, making the combinations that include them exhibit the same type of behaviour.

With the decrease in negative recall being almost twice the increase in positive recall, RBC detects more polyps when compared to RBM, but the trade-off is not likely to be worth it in most scenarios. This is also indicated by the 0.82% reduction in weighted F1-score.

4.4.4.3 Rotation, brightness variations, masking reflections and contrast enhancement

For this experiment, rotation, brightness variations, masking reflections and contrast enhancement have been combined. The results can be seen in table 4.15 and figure 4.15.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RBMC											
100k	23.46%	21.64	23.46	21.30	95.60%	95.96	95.60	95.75	**91.94**	**92.05**	**92.04**
200k	26.18%	21.49	26.18	22.63	95.44%	96.06	95.44	95.73	**92.01**	**92.23**	**91.98**
300k	25.23%	19.70	25.23	21.31	95.22%	96.00	95.22	95.59	**91.80**	**92.06**	**91.72**
400k	25.06%	19.48	25.06	21.11	95.20%	95.98	95.20	95.57	**91.77**	**92.03**	**91.69**
500k	25.00%	19.42	25.00	21.05	95.20%	95.98	95.20	95.57	**91.77**	**92.03**	**91.69**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
RBM	30.24%	24.86	30.24	27.06	95.29%	96.23	95.29	95.75	**92.25**	**92.56**	**92.01**
RBC	31.12%	20.86	31.12	24.40	93.80%	96.28	93.80	95.01	**91.43**	**92.40**	**90.70**

Table 4.15: Results using RBMC

Compared to the non-preprocessed version, we can see a decrease in weighted recall of 2.77%, a decrease in weighted precision of 0.79% and a decrease in weighted F1-score of 1.53%. RBMC shows a big decrease in positive recall from RBM and RBC, and unlike the others, it is also lower than non-preprocessed. The negative recall is between the previous two combinations, which is natural.

Like RBC, this combination has many drawbacks, a low positive recall and similar negative recall, compared to RBM. It also lacks any unique positive properties, making RBMC an unoptimal combination.

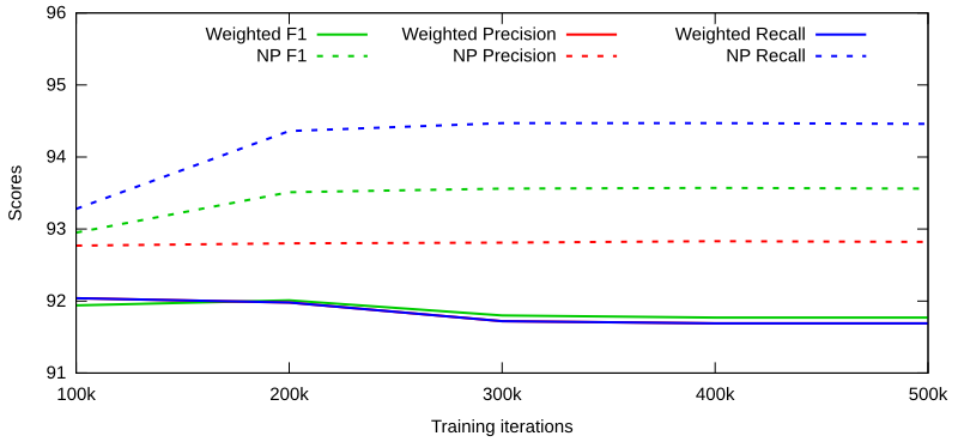


Figure 4.15: Weighted scores compared to NP using RBMC

4.4.4.4 Summary of rotation, brightness variations, masking reflections and contrast enhancement

In this section, we combined RB with masking reflections and contrast enhancement, performing experiments with RBM, RBC and RBMC. RBM produced the most promising results, with an increase of 3.95% in positive recall and a decrease of 2.85% in negative recall compared to the non-preprocessed version. This trade-off may be worth it in scenarios where the positive recall is focused. RBC increased the positive recall by 0.88% compared to RBM, but the negative recall decreased by 1.49%. This creates an uneven balance, where the decrease is almost twice the size of the increase. Due to the unbalanced dataset, the increase in FPs is substatially higher than that of TPs. RBMC decreased the positive recall by 6.12% from RBC and 5.24% from RBM, to 25.00%, which is the lowest in this set. The negative recall is roughly equal to RBM.

From these results, the only combination with a trade-off between positive and negative recall that could be worth it is RBM. Both RBC and RBMC has too big of a decrease in either positive or negative recall.

4.4.5 Rotation, masking reflections and contrast enhancement

In this final set of the data enhancement methods, we will conduct the same experiments as in the previous set, but leaving out brightness variations. This is because brightness variations seems to only produce adverse effects, and we want to see the results the previous set can achieve when it is removed. This will also give us a better idea as to what degree brightness variations lowers the performance.

4.4.5.1 Rotation and masking reflections

For this experiment, rotation and masking reflections have been combined. The results can be seen in table 4.16 and figure 4.16.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RM											
100k	28.92%	30.29	28.92	28.60	96.04%	96.24	96.05	96.13	**92.72**	**92.85**	**92.74**
200k	31.75%	35.04	31.75	32.61	97.17%	96.49	97.17	96.78	**93.49**	**93.19**	**93.89**
300k	31.99%	31.31	31.99	31.12	96.86%	96.39	96.86	96.62	**93.28**	**93.04**	**93.59**
400k	31.51%	30.73	31.50	30.56	96.84%	96.37	96.84	96.60	**93.23**	**93.00**	**93.55**
500k	31.62%	30.84	31.61	30.67	96.85%	96.37	96.85	96.60	**93.24**	**93.01**	**93.56**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
RBM	30.24%	24.86	30.24	27.06	95.29%	96.23	95.29	95.75	**92.25**	**92.56**	**92.01**

Table 4.16: Results using RM

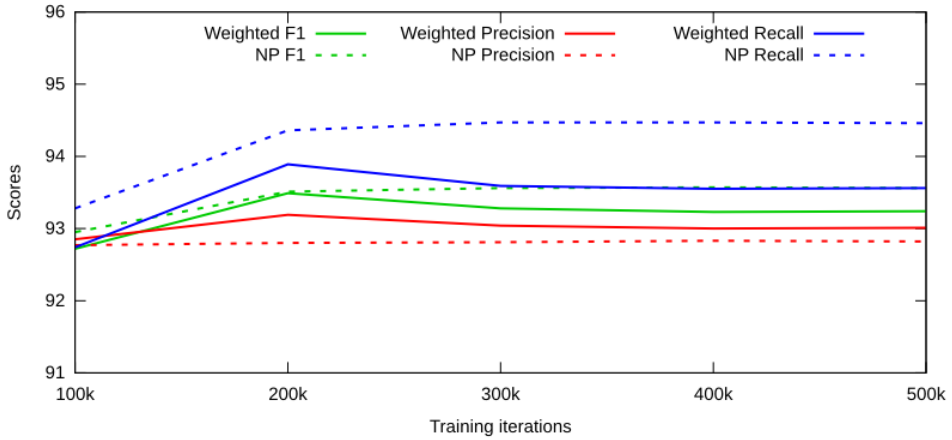


Figure 4.16: Weighted scores compared to NP using RM

The results show an increase in weighted recall of 1.55%, an increase in weighted precision of 0.45% and an increase in weighted F1-score of 0.99% compared to RBM. In addition, the positive recall has increased by 1.38% and the negative recall has increased by 1.56%. RM is able to increase the number of TPs while also decreasing the number of FPs. These numbers indicate that brightness variations is unable to yield positive effects both when used individually and in combinations with other data enhancement methods.

4.4.5.2 Rotation and contrast enhancement

For this experiment, rotation and contrast enhancement have been combined. The results can be seen in table 4.17 and figure 4.17.

From the results, we can see an increase in weighted recall of 2.27%, an increase in weighted precision of 0.88% and an increase in weighted F1-score of 1.65% compared to RBC. We can also see an increase of 6.67% in positive recall and an increase of 2.13% in negative recall. The positive recall of 37.79% is the highest result achieved as of yet, even beating

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RC											
100k	32.82%	36.80	32.82	33.13	97.27%	96.46	97.27	96.86	**93.53**	**93.42**	**94.03**
200k	37.71%	31.42	37.71	33.73	96.13%	96.65	96.13	96.39	**93.21**	**93.33**	**93.17**
300k	37.92%	31.05	37.92	33.69	96.01%	96.64	96.01	96.32	**93.14**	**93.30**	**93.05**
400k	37.80%	30.76	37.80	33.43	95.92%	96.63	95.92	96.27	**93.08**	**93.28**	**92.96**
500k	37.79%	30.80	37.79	33.44	95.93%	96.63	95.93	96.27	**93.08**	**93.28**	**92.97**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
RM	31.62%	30.84	31.61	30.67	96.85%	96.37	96.85	96.60	**93.24**	**93.01**	**93.56**
RBC	31.12%	20.86	31.12	24.40	93.80%	96.28	93.80	95.01	**91.43**	**92.40**	**90.70**

Table 4.17: Results using RC

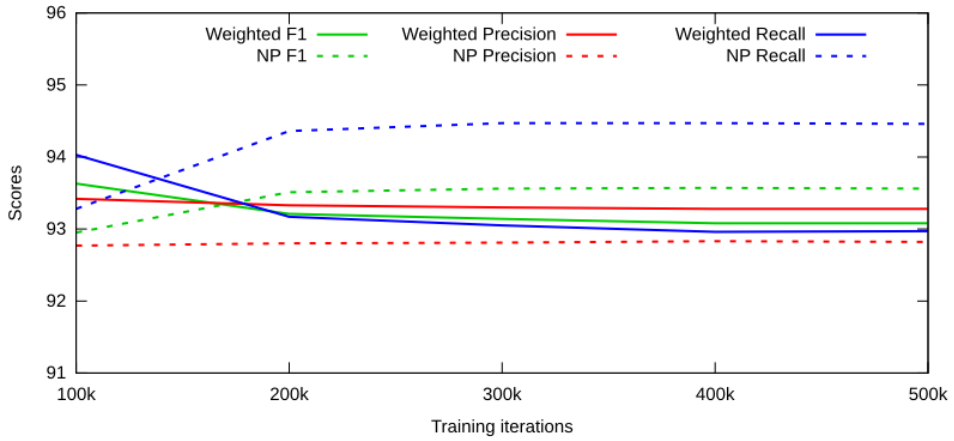


Figure 4.17: Weighted scores compared to NP using RC

rotation by 5.99%, which was the highest result before RC. The negative recall is 1.48% lower than rotation, which is why rotation has the highest weighted F1-score. These results support our belief from the previous section that brightness variations is unable to yield positive effects.

In scenarios with a high focus on detected polyps, RC is a good candidate as it produces the highest positive recall while still maintaining a high negative recall. If the focus is on overall performance, or on finding few FPs, rotation would be a better candidate.

4.4.5.3 Rotation, masking reflections and contrast enhancement

For this experiment, rotation, masking reflections and contrast enhancement have been combined. The results can be seen in table 4.18 and figure 4.18.

Compared to RBMC, the weighted recall is increased by 0.71%, the weighted recall is increased by 0.58% and the weighted F1-score is

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
RMC											
100k	32.83%	27.71	32.83	28.76	96.06%	96.39	96.06	96.21	**92.79**	**92.89**	**92.84**
200k	29.32%	25.01	29.32	25.70	96.00%	96.23	96.00	96.10	**92.51**	**92.58**	**92.64**
300k	29.82%	24.95	29.82	25.93	95.73%	96.24	95.73	95.97	**92.41**	**92.60**	**92.40**
400k	29.99%	24.79	29.99	25.84	95.69%	96.24	95.69	95.95	**92.39**	**92.60**	**92.37**
500k	29.99%	24.97	29.99	25.95	95.72%	96.24	95.73	95.97	**92.41**	**92.61**	**92.40**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
RM	31.62%	30.84	31.61	30.67	96.85%	96.37	96.85	96.60	**93.24**	**93.01**	**93.56**
RC	37.79%	30.80	37.79	33.44	95.93%	96.63	95.93	96.27	**93.08**	**93.28**	**92.97**
RBMC	25.00%	19.42	25.00	21.05	95.20%	95.98	95.20	95.57	**91.77**	**92.03**	**91.69**

Table 4.18: Results of using RMC

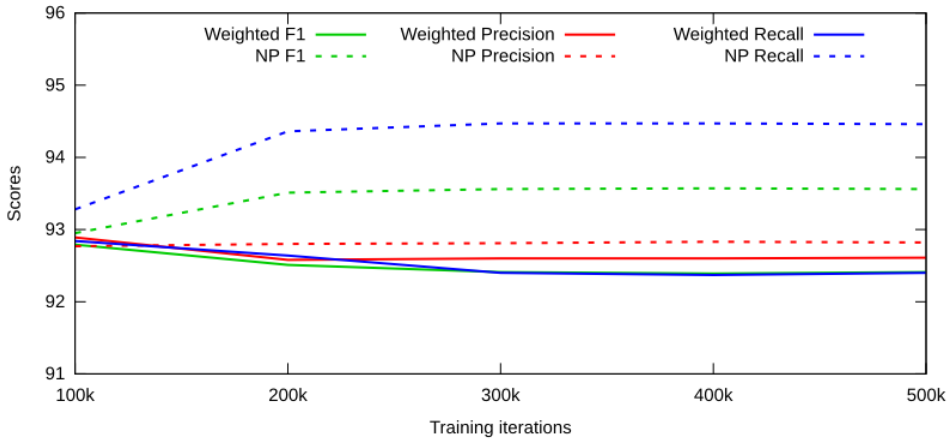


Figure 4.18: Weighted scores compared to NP using RMC

increased by 0.64%. RMC follows a similar trend as RBMC, where the positive recall has decreased and the negative recall is relatively consistent. These are the lowest results within this set. Both positive and negative recall and precision has decreased when compared to RM and RC.

We have now conducted three experiments where brightness variations have been left out, where all show an increase in every metric. This proves that our system, when using our implementation of brightness variations, is unable to produce any positive effects.

4.4.5.4 Summary of rotation, masking reflections and contrast enhancement

From the first and third set, we have seen that brightness variations only brings a detrimental effect. We therefore repeated the same experiments as in the third set, but with brightness variations excluded.

In both RM, RC and RMC, we see that all metrics have increased by excluding brightness variations. In positive and negative recall, the

combinations follow the same pattern as RBM, RBC and RBMC. RC achieves a positive recall of 37.79%, which is the highest achieved for all combinations regardless of set, while also maintaing a relatively high negative recall of 95.93%. This is therefore an exciting combination, as it could be a good candidate when polyp detection rate is more important than limiting the number of FPs.

For all three experiments, excluding brightness variations has improved the results. This proves that our system, when using our implementation of brightness variations, is unable to produce any positive effects.

4.4.6 Summary

In this section, we have performed experiments with various data enhancement methods, both individually and in various combinations. We began by experimenting with a non-preprocessed version in order to get a basis for comparing the effects of each data enhancement method. We divided the data enhancement methods into four sets, where the first two are with different types of data enhancements, while the last two are combinations of the first two sets. Table 4.19 shows the results from the experiments, where all combinations in a set are grouped.

Combination	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
NP	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**
R	31.80%	36.51	31.80	33.67	97.41%	96.35	97.41	96.88	**93.62**	**93.22**	**94.06**
B	20.70%	20.24	20.70	19.87	96.01%	95.62	96.01	95.80	**91.85**	**91.73**	**92.05**
RB	27.65%	25.09	27.66	25.79	96.60%	96.17	96.60	96.38	**92.78**	**92.52**	**93.13**
M	22.26%	24.74	22.26	22.85	97.76%	96.00	97.76	96.86	**93.20**	**92.52**	**93.98**
C	24.62%	27.26	24.62	25.14	97.60%	96.15	97.60	96.86	**93.28**	**92.69**	**93.99**
MC	24.82%	27.84	24.82	25.92	97.87%	96.12	97.88	96.98	**93.45**	**92.74**	**94.22**
RBM	30.24%	24.86	30.24	27.06	95.29%	96.23	95.29	95.75	**92.25**	**92.56**	**92.01**
RBC	31.12%	20.86	31.12	24.40	93.80%	96.28	93.80	95.01	**91.43**	**92.40**	**90.70**
RBMC	25.00%	19.42	25.00	21.05	95.20%	95.98	95.20	95.57	**91.77**	**92.03**	**91.69**
RM	31.62%	30.84	31.61	30.67	96.85%	96.37	96.85	96.60	**93.24**	**93.01**	**93.56**
RC	37.79%	30.80	37.79	33.44	95.93%	96.63	95.93	96.27	**93.08**	**93.28**	**92.97**
RMC	29.99%	24.97	29.99	25.95	95.72%	96.24	95.73	95.97	**92.41**	**92.61**	**92.40**

Table 4.19: Summary of all the results from the different data enhancement methods, where the sets are seperated, given 90% confidence and 500k training iterations

The first set of the data enhancement methods is about artificially increasing the dataset. It consists of rotation and brightness variations and the combination of the two. The goal of this is to give the system additional samples to learn from, by adding polyps in different angles and brightness levels. The second set is about improving the quality of the dataset. It consists of masking reflections and contrast enhancement and the combination of the two. The goal is to remove weaknesses in the existing dataset, such as reflections and polyps in low-contrast areas, which

is a common problem in colonoscopy videos. The third set combines the first two in an effort to increase both the quantity and the in the dataset. The fourth set was created because of a detrimental effect caused by brightness variations in set one and three. This set repeats the experiments from set three, only without brightness variations.

Rotation shows an ability to improve positive recall, but also produces a slight decrease in negative recall. However, the improvement in positive recall is enough to offset the decrease in negative recall, resulting in rotation being the only data enhancement method able to beat the F1-score of NP. Brightness variations shows an inability to improve any aspect of the results, both when used independently and in combinations. It lowers both positive and negative recall, and thus affects weighted F1-score negatively. Masking reflections has a varying effect on the results, depending on the videos. The best experienced result was an increase in positive recall of 5% and negative recall of 1.5%, while the worst experienced result was a decrease in positive recall of 19% and negative recall of 1.3%. Due to this, masking reflections had an overall negative effect on the F1-score. A more advanced implementation of masking reflections that are able to handle reflections of many shapes and colors, may be able to increase the results in additional videos, making masking reflections able to improve overall performance. Contrast enhancement has a varying effect on the results, depending on the polyps in the videos. If there are polyps with defined outlines, it is able to enhance the polyp, making it more detectable. If there are no outlines, for instance if the polyp is part of other structures in the colon, the polyp will not be more detectable, but instead other structures that resemble that of a polyp may be mistaken as such, producing additional FPs.

Rotation can be combined with any data enhancement method, where it is able to increases the positive recall while only slightly lowering the negative recall, making the addition of rotation an overall improvement. When masking reflections and contrast enhancement are combined, they improve each others results by providing mutual gains, improving the performance from their individual results. By introducing rotation to MC, the mutual gains are not present, making RMC produce worse results than RM and RC. When introducing rotation to either masking reflections or contrast enhancement, we see the highest increase in positive recall, where RMs 31.62% is up from 22.26% and RCs 37.79% is up from 24.62%, which makes RCs positive recall an increase of 11.50% compared to NP. The combinations that include contrast enhancement tend to have a higher positive recall, while those that include masking reflections tend to have a higher negative recall. Because of the imbalance in the dataset, masking reflections achieves a higher F1-score than contrast enhancement. We still view RC as the most interesting combination because of its high positive recall and relatively high negative recall, and is thus a suitable combination for various scenarios.

4.5 Training optimization

We have previously conducted experiments in section 4.4, where the focus was to examine the effects of the different data enhancement methods. The results there show that when we are able to increase the positive recall, the negative recall was lowered as a consequence.

In this section, we modify the training by experimenting with different neural networks, dataset balances, training techniques and optimizers. This is done in an effort to improve positive recall, negative recall or achieve a better balance between the two. Non-preprocessed data will be used for all experiments unless otherwise stated.

In the previous experiment, we included training iterations from 100k to 500k in the results. There we noticed that the results stabilizes around 300k, with only minor changes up to 500k, which is why 500k were used when discussing the results. This pattern exists in this experiment as well, which is why 500k training iterations is used without displaying the others. Different confidences were excluded from the results and discussions in the previous experiment, as the different confidences only produced minor shifts between positive and negative recall. This applied to all data enhancement methods, both when used individually and in combinations. As an example, the non-preprocessed version from the previous experiment can be seen in table 4.22, where the confidences only show a minor spread. A confidence of 90% always produced the highest weighted F1-score, and was thus used for comparison. By modifying the training, we hope to see a larger shift when comparing different confidences, and as such confidences are included in the results and discussions in this experiment.

4.5.1 Different types of neural networks

In this section, we experiment with different types of neural networks. In recent years other types of neural networks have been developed, which have proven to produce promising results. We experiment with Inception and Resnet to see what effect they have on polyp detection.

4.5.1.1 Inception

We began with Inception [54] because it could give us a good foundation in which we can build upon for polyp detection. By using Inception, we hope to see an increase in the detection of polyps, while keeping the negative percentage consistent.

In this experiment, TensorFlow 1.0 and an updated TensorBox version were used. Support for Resnet was added in a later version of TensorBox, which required TensorFlow 1.0. In our case, there are no differences between TensorFlow 0.12.1 and 1.0, except that it was required by the updated TensorBox version. The experiment with Inception was performed on the same machine as for Resnet, which is why the updated versions are used for Inception as well.

Inception was used by specifying the Inception checkpoint in the TensorBox settings file. An Inception checkpoint is a file that contains the weights of a pre-trained network, which is used as a general basis on which to train the network for a specific task. The checkpoint is `inception_v1.ckpt`, which TensorBox downloads during setup.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Inception											
10%	94.20%	09.11	94.20	16.61	20.05%	97.60	20.05	33.27	**31.96**	**90.66**	**25.86**
20%	93.07%	09.54	93.07	17.31	24.94%	97.69	24.94	39.74	**37.98**	**90.78**	**30.28**
30%	91.69%	09.89	91.69	17.85	28.90%	97.61	28.90	44.60	**42.50**	**90.73**	**33.82**
40%	91.26%	10.26	91.26	18.45	31.12%	97.74	32.12	48.35	**46.01**	**90.88**	**36.76**
50%	90.74%	10.69	90.74	19.13	35.54%	97.83	35.54	52.14	**49.55**	**91.00**	**39.87**
60%	90.22%	11.14	90.22	19.83	38.81%	97.90	38.81	55.58	**52.78**	**91.10**	**42.84**
70%	89.61%	11.72	89.61	20.73	42.58%	97.97	42.58	59.36	**56.33**	**91.21**	**46.27**
80%	88.66%	12.67	88.66	22.17	48.03%	98.03	48.03	64.47	**61.15**	**91.34**	**51.21**
90%	86.67%	14.52	86.67	24.87	56.62%	98.04	56.62	71.78	**68.10**	**91.49**	**58.98**
RNN (Default)											
10%	70.04%	63.40	70.04	66.55	96.56%	97.43	96.56	96.99	**94.60**	**94.76**	**94.48**
20%	69.44%	64.57	69.44	66.92	96.76%	97.38	96.76	97.07	**94.71**	**94.81**	**94.62**
30%	69.09%	65.14	69.09	67.06	96.86%	97.36	96.86	97.11	**94.75**	**94.83**	**94.68**
40%	68.83%	65.92	68.83	67.34	96.97%	97.34	96.97	97.15	**94.81**	**94.88**	**94.76**
50%	68.57%	66.72	68.57	67.63	97.09%	97.32	97.09	97.20	**94.88**	**94.92**	**94.85**
60%	68.31%	67.61	68.31	67.96	97.22%	97.30	97.22	97.26	**94.96**	**94.97**	**94.95**
70%	67.97%	68.38	67.97	68.17	97.33%	97.28	97.33	97.30	**95.02**	**95.01**	**95.03**
80%	67.45%	69.43	67.45	68.43	97.47%	97.24	97.47	97.35	**95.08**	**95.06**	**95.12**
90%	66.32%	70.66	66.32	68.42	97.66%	97.15	97.66	97.40	**95.13**	**95.07**	**95.20**

Table 4.20: The results of using Inception and RNN with split 1

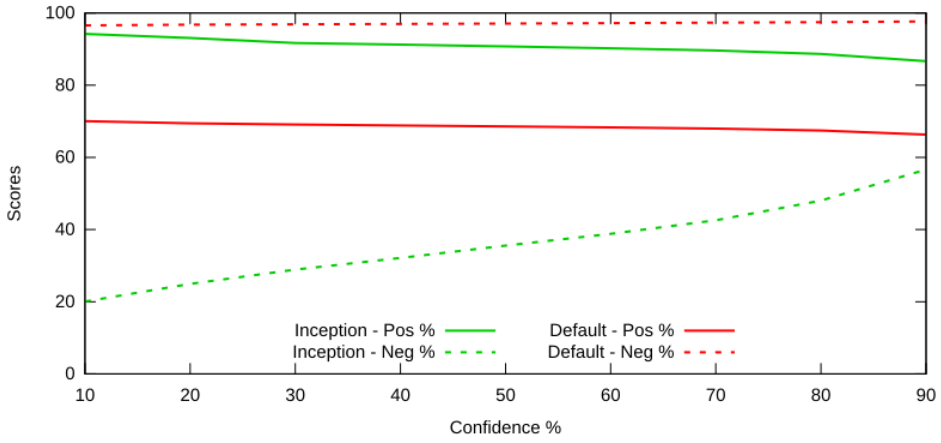


Figure 4.19: Graph of the results using Inception and RNN with split 1

As can be seen in table 4.20 and figure 4.19, Inception was able to detect a higher number of TPs, but at the same time a much lower number of TNs. The positive recall for Inception showed a spread of about 8% depending on the confidence, while negative recall showed a spread of about 36%.

Already at 90% confidence, Inception has almost 6 000 FPs, which further increases to almost 11 000 at 10% confidence. RNN, on the other hand, showed a smaller spread, where positive recalls spread is about 4% and the negative recalls spread is only about 1%. This is visible in the weighted F1-scores, where Inception results in a spread of 36%, while RNN results in a spread of 0.5%.

While Inception has a very high positive recall, the negative recall makes it less suitable for many scenarios. With a confidence of 90%, for every image it classified as a polyp, only 1 in 7 truly contained a polyp. As a comparison, with the default setup, 2 of 3 images classified as containing polyps truly contained a polyp.

From this, we conclude that the way TensorBox has implemented Inception, it is unoptimal for most scenarios in polyp detection. Due to the low negative detection rate, we decided to not persue Inception further, looking at other possibilities instead.

4.5.1.2 Resnet

Resnet [17] is another form of neural network which differs from RNN in the way it transfers earlier knowledge. It is a very deep neural network, and have proven to produce good results in various fields. These good results are the reason that we chose Resnet as the second type of neural network to inspect.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Resnet											
10%	89.35%	10.01	89.35	18.00	31.71%	97.22	31.71	47.82	**45.48**	**90.38**	**36.23**
20%	88.66%	10.59	88.66	18.92	36.35%	97.41	36.35	52.94	**50.27**	**90.60**	**40.45**
30%	87.79%	11.01	87.79	19.57	39.65%	97.45	39.65	56.37	**53.49**	**90.67**	**43.42**
40%	87.53%	11.50	87.53	20.33	42.71%	97.58	42.71	59.41	**56.35**	**90.83**	**46.22**
50%	87.36%	11.97	87.36	21.06	45.36%	97.68	45.36	61.95	**58.74**	**90.96**	**48.65**
60%	86.75%	12.48	86.75	21.88	48.24%	97.72	48.24	64.59	**61.24**	**91.04**	**51.26**
70%	86.06%	13.09	86.06	22.77	51.40%	97.75	51.40	67.37	**63.87**	**91.11**	**54.12**
80%	85.45%	13.99	85.45	24.04	55.33%	97.81	55.33	70.68	**67.02**	**91.24**	**57.69**
90%	84.68%	15.44	84.68	26.12	60.55%	97.89	60.66	74.82	**71.00**	**91.43**	**62.44**
RNN (Default)											
10%	70.04%	63.40	70.04	66.55	96.56%	97.43	96.56	96.99	**94.60**	**94.76**	**94.48**
20%	69.44%	64.57	69.44	66.92	96.76%	97.38	96.76	97.07	**94.71**	**94.81**	**94.62**
30%	69.09%	65.14	69.09	67.06	96.86%	97.36	96.86	97.11	**94.75**	**94.83**	**94.68**
40%	68.83%	65.92	68.83	67.34	96.97%	97.34	96.97	97.15	**94.81**	**94.88**	**94.76**
50%	68.57%	66.72	68.57	67.63	97.09%	97.32	97.09	97.20	**94.88**	**94.92**	**94.85**
60%	68.31%	67.61	68.31	67.96	97.22%	97.30	97.22	97.26	**94.96**	**94.97**	**94.95**
70%	67.97%	68.38	67.97	68.17	97.33%	97.28	97.33	97.30	**95.02**	**95.01**	**95.03**
80%	67.45%	69.43	67.45	68.43	97.47%	97.24	97.47	97.35	**95.08**	**95.06**	**95.12**
90%	66.32%	70.66	66.32	68.42	97.66%	97.15	97.66	97.40	**95.13**	**95.07**	**95.20**

Table 4.21: The results of using Resnet and RNN with split 1

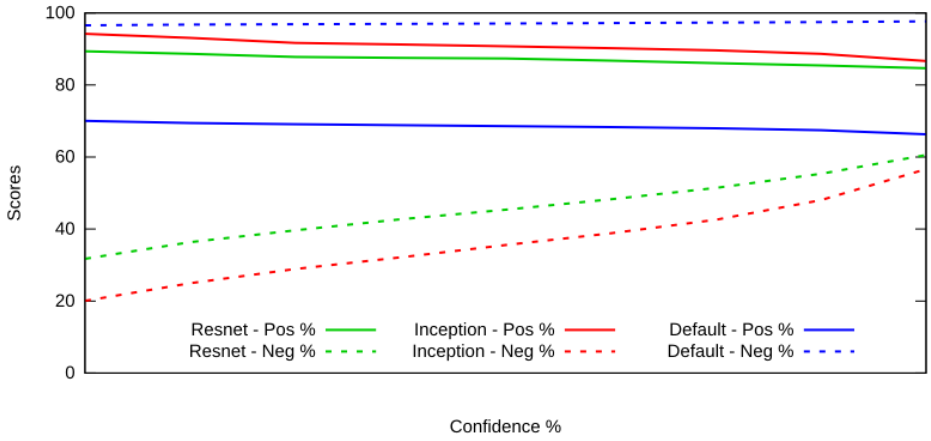


Figure 4.20: Graph of the results using Resnet and RNN with split 1

TensorFlow 1.0 and an updated version of TensorBox were used as support for Resnet was only added in later versions, as explained in the previous section. Resnet was used by specifying the use of Resnet in the settings file.

The results in table 4.21 and figure 4.20 show a similar pattern as Inception with a high positive recall, but it struggles with a low negative recall. The positive and negative recall also have a larger spread than RNN, but to a lesser degree than Inception. Overall, Resnet suffers from the same issues as Inception, and in addition, each training iteration took twice the amount of time to complete compared to the other networks. Due to the low negative recall of Inception and Resnet, we chose to keep RNN as part of the following sections.

4.5.2 Dataset balance optimizations

In our dataset, the negative samples dominate the total number of samples, which creates a focus on correct classification of negative samples. The result is that positive samples become secondary citizens, in that it becomes less important to classify them correctly. Dataset balancing can be done with a goal of altering the focus of the neural network. By balancing the dataset, the focus can be shifted towards the positive samples, or somewhere in between.

For evaluation, the full dataset, including every negative sample, is used to be able to compare against all other versions. The results from the full dataset is gathered from section 4.4.1, and repeated in table 4.22, but where training iterations have been replaced by confidences.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Full dataset											
10%	29.01%	27.07	29.01	27.41	96.97%	96.20	96.97	96.57	**93.04**	**92.68**	**93.48**
20%	28.54%	27.71	28.54	27.49	97.16%	96.18	97.16	96.66	**93.13**	**92.70**	**93.64**
30%	28.02%	27.85	28.02	27.31	97.29%	96.17	97.29	96.71	**93.17**	**92.69**	**93.74**
40%	27.71%	28.14	27.71	27.32	97.40%	96.16	97.40	96.76	**93.22**	**92.70**	**93.83**
50%	27.59%	28.63	27.59	27.46	97.51%	96.15	97.51	96.82	**93.27**	**92.72**	**93.93**
60%	27.42%	29.06	27.42	27.54	97.63%	96.15	97.63	96.88	**93.34**	**92.74**	**94.03**
70%	27.25%	29.88	27.25	27.80	97.78%	96.14	97.78	96.94	**93.41**	**92.77**	**94.16**
80%	26.83%	30.44	26.83	27.78	97.92%	96.13	97.92	97.01	**93.47**	**92.79**	**94.28**
90%	26.29%	31.60	26.29	27.85	98.14%	96.11	98.14	97.10	**93.56**	**92.82**	**94.46**

Table 4.22: The results of using a full dataset

4.5.2.1 Balanced dataset

For this experiment, the dataset was roughly balanced 50-50 between positive and negative samples to avoid a dominant side. This was done by removing all non-polyp videos except `np_5` from the evaluation set used during training. In theory, this should even out the focus.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Balanced dataset											
10%	31.44%	27.88	31.44	28.88	96.87%	96.31	96.87	96.57	**93.11**	**92.84**	**93.49**
20%	30.93%	28.62	30.93	29.05	97.07%	96.30	97.07	96.67	**93.21**	**92.85**	**93.66**
30%	30.67%	28.83	30.67	29.08	97.19%	96.29	97.19	96.72	**93.26**	**92.86**	**93.76**
40%	30.47%	29.16	30.47	29.17	97.28%	96.28	97.28	96.77	**93.30**	**92.86**	**93.84**
50%	30.34%	29.60	30.35	29.34	97.37%	96.28	97.37	96.81	**93.35**	**92.88**	**93.92**
60%	30.12%	29.93	30.12	29.40	97.45%	96.27	97.45	96.84	**93.38**	**92.88**	**93.98**
70%	29.75%	30.38	29.75	29.43	97.57%	96.25	97.57	96.89	**93.43**	**92.88**	**94.07**
80%	29.54%	31.33	29.54	29.75	97.72%	96.25	97.73	96.97	**93.51**	**92.92**	**94.21**
90%	28.71%	32.08	28.71	29.66	97.97%	96.21	97.97	97.07	**93.61**	**92.93**	**94.40**

Table 4.23: The results of using a balanced dataset

From the results in table 4.23 and figure 4.21, we can see that the balanced dataset, compared to the full dataset, is basically an offset by an increase of around 2.5% in positive recall and a decrease of around 0.15% in negative recall. The increase in positive recall in itself is not major, but even with our dataset, the increase in TPs is higher than the decrease in TNs, positively affecting the F1-score by about 0.06%.

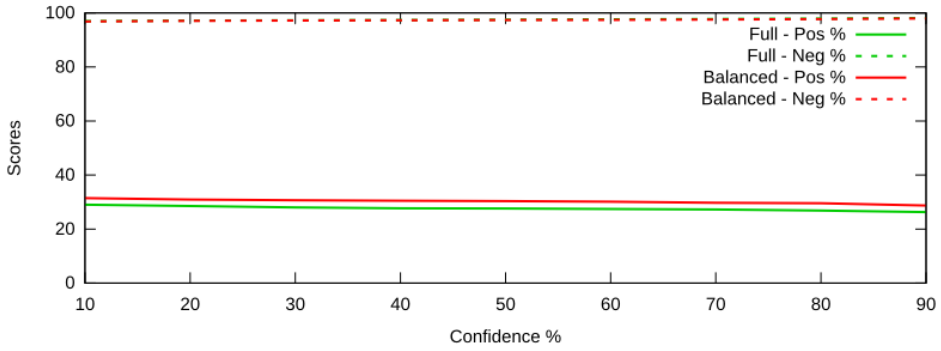


Figure 4.21: Graph of the results using a full and balanced dataset

4.5.2.2 Low negatives dataset

In the previous section, we saw a positive effect on the polyp detection rate when lowering the number of negative samples. In this section, we remove any explicit negative samples, which is done by removing all non-polyp videos from the evaluation set used during training. The only negative samples remaining are the images not containing a polyp, but are part of the polyp videos. This should shift the focus towards the positive samples.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Low negative dataset											
10%	32.38%	28.82	32.38	29.54	97.07%	96.39	97.07	96.71	**93.28**	**92.96**	**93.74**
20%	31.65%	29.12	31.65	29.46	97.27%	96.36	97.27	96.79	**93.36**	**92.94**	**93.90**
30%	31.40%	29.62	31.40	29.58	97.38%	96.35	97.38	96.85	**93.42**	**92.96**	**94.00**
40%	31.03%	29.91	31.04	29.54	97.47%	96.35	97.48	96.89	**93.46**	**92.97**	**94.08**
50%	30.69%	30.13	30.69	29.57	97.58%	96.33	97.58	96.93	**93.49**	**92.96**	**94.16**
60%	30.32%	30.55	30.32	29.59	97.69%	96.32	97.69	96.98	**93.54**	**92.97**	**94.24**
70%	29.90%	31.09	29.90	29.61	97.80%	96.30	97.80	97.03	**93.59**	**92.98**	**94.33**
80%	29.37%	31.70	29.38	29.59	97.94%	96.29	97.94	97.09	**93.65**	**93.00**	**94.44**
90%	28.55%	33.00	28.54	29.74	98.19%	96.26	98.19	97.20	**93.77**	**93.05**	**94.64**

Table 4.24: The results of using a low negative dataset

Looking at the results in table 4.24 and figure 4.22, most confidences show a slight increase in positive recall compared to the balanced dataset. Additionally, all confidences show improvements in negative recall compared to both the balanced and full dataset, which results in the higest F1-scores.

The results are somewhat surprising, as the assumption was an increase in positive recall and a slight decrease in negative recall, while the results show an increase in both. Due to the good results, this dataset is used in following sections.

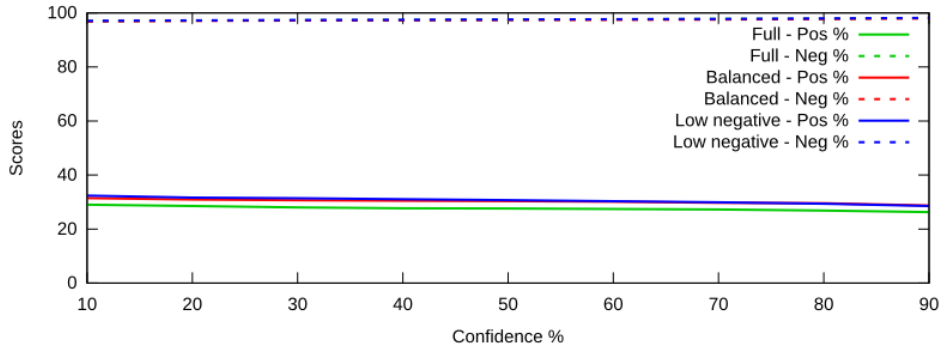


Figure 4.22: Graph of the results using a full, balanced and low negative dataset

4.5.3 Comparing training techniques

TensorBox comes with support for LSTM, in addition to Rezoom which was used during the data enhancement experiments, and also in this section up until now. After having only used Rezoom this far, we wanted to see what effect LSTM would have, and if it could improve the results. Since LSTM can be used both seperately and in combination with Rezoom, there are four possible combinations which are Rezoom, LSTM, both and none. To determine which combination to examine closer, we conducted a limited experiment with all combinations on split 5, which can be found in table 4.25.

Version	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
None	8.23%	13.01	8.23	10.08	97.55%	95.97	97.55	96.75	**93.05**	**92.43**	**93.74**
LSTM	22.42%	18.61	22.42	20.34	95.63%	96.51	95.63	96.07	**92.84**	**93.19**	**92.51**
Rezoom	12.26%	16.24	12.26	13.97	97.18%	96.13	97.18	96.65	**93.12**	**92.72**	**93.56**
Both	17.10%	17.52	17.10	17.31	96.41%	96.31	96.41	96.36	**92.99**	**92.95**	**93.02**

Table 4.25: Results of using Rezoom and LSTM and their combinations with 90% as confidence

LSTM produces the best positive recall and precision, while also the lowest negative recall. Rezoom on the other hand, produces lower positive recall and precision, but higher negative recall, giving it the highest weighted F1-score. With both Rezoom and LSTM enabled, scores seem to be more or less the average between the two. This indicates that each of them bring their own effect, which seem to average each other out with no visible common gains.

As we have already experimented with Rezoom, we chose to take a closer look at LSTM, as it seemed promising due to its high positive recall and still relatively high negative recall. Even though the F1-score is slightly

lower than the rest, we consider LSTM to have a high potential in real world scenarios.

4.5.3.1 LSTM

Here we take a closer look at LSTM with Rezoom disabled, and unlike the last section we now use cross validation as normal.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
LSTM											
10%	36.03%	21.14	36.03	26.19	94.12%	96.48	94.12	95.27	**91.77**	**92.66**	**91.14**
20%	35.54%	21.97	35.54	26.70	94.57%	96.47	94.57	95.49	**92.01**	**92.70**	**91.54**
30%	35.14%	22.52	35.14	27.01	94.85%	96.46	94.85	95.63	**92.16**	**92.72**	**91.79**
40%	34.75%	22.88	34.75	27.19	95.05%	96.45	95.04	95.73	**92.25**	**92.73**	**91.95**
50%	34.55%	23.33	34.56	27.48	95.25%	96.45	95.25	95.83	**92.36**	**92.75**	**92.14**
60%	34.04%	23.68	34.04	27.59	95.45%	96.43	95.45	95.93	**92.46**	**92.74**	**92.31**
70%	33.73%	24.23	33.73	27.88	95.68%	96.42	95.68	96.03	**92.58**	**92.76**	**92.51**
80%	33.39%	24.78	33.39	28.16	95.94%	96.41	95.94	96.16	**92.71**	**92.79**	**92.74**
90%	32.67%	25.81	32.67	28.58	96.33%	96.40	96.33	96.35	**92.91**	**92.82**	**93.08**

Table 4.26: The results of using LSTM

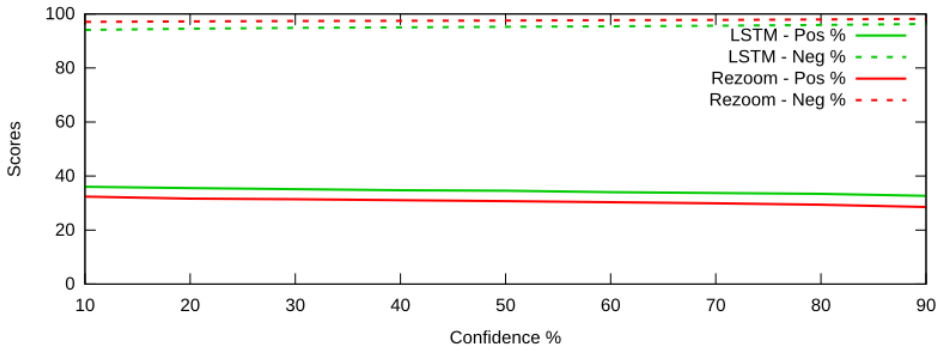


Figure 4.23: Graph of the results using LSTM

The results in table 4.26 and figure 4.23 show an improvement in positive recall of around 4% from when Rezoom was used, as seen in table 4.24. The same tables show that the negative recall has decreased by around 2% for higher confidences and up to around 3% for lower confidences.

These numbers show that there are no best version for every scenario. We therefore chose to include both in at least one additional experiment in order to see if their properties would continue even when different optimizers were used.

4.5.4 Comparing optimizers

For all experiments up until now, RMS has been used as the optimizer. In addition to RMS, TensorBox comes with support for SGD and Adam, which are explained in section 3.4. We will compare the results from SGD and Adam to those of RMS, which can be viewed in table 4.24 for the version using Rezoom and in table 4.26 for the one with LSTM.

4.5.4.1 SGD

In this experiment, SGD was used as the optimizer for two different runs, one with Rezoom and the other with LSTM.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Rezoom + SGD											
10%	60.04%	21.46	60.04	30.64	88.60%	97.48	88.60	92.81	**89.65**	**93.62**	**86.98**
20%	56.43%	24.60	56.43	33.16	91.27%	97.35	91.27	94.19	**91.10**	**93.68**	**89.33**
30%	53.78%	27.02	53.78	34.81	92.91%	97.26	92.91	95.02	**91.97**	**93.73**	**90.76**
40%	51.46%	29.83	51.46	36.55	94.27%	97.17	94.27	95.68	**92.68**	**93.79**	**91.93**
50%	49.63%	32.66	49.63	38.22	95.33%	97.11	95.33	96.20	**93.25**	**93.89**	**92.85**
60%	47.28%	35.66	47.28	39.43	96.28%	97.02	96.28	96.63	**93.72**	**93.96**	**93.63**
70%	43.60%	39.63	43.60	40.30	97.28%	96.85	97.28	97.06	**94.15**	**94.01**	**94.40**
80%	39.18%	45.36	39.18	40.70	98.26%	96.64	98.26	97.44	**94.51**	**94.10**	**95.09**
90%	29.28%	54.70	29.28	36.58	99.27%	96.11	99.27	97.66	**94.41**	**94.02**	**95.49**
LSTM + SGD											
10%	76.30%	07.53	76.30	13.57	49.82%	97.53	49.81	65.56	**62.86**	**92.84**	**51.12**
20%	66.48%	08.74	66.48	15.14	62.72%	97.08	62.72	75.89	**72.72**	**92.49**	**62.72**
30%	56.59%	10.05	56.59	16.30	72.14%	96.59	72.14	82.30	**78.84**	**92.12**	**70.99**
40%	47.15%	12.93	47.15	17.45	80.22%	96.23	80.22	87.19	**83.52**	**91.98**	**78.02**
50%	40.97%	24.92	40.97	20.42	86.74%	96.11	86.74	90.96	**87.29**	**92.77**	**83.88**
60%	29.67%	28.98	29.67	13.38	92.03%	95.58	92.03	93.65	**89.24**	**92.53**	**88.23**
70%	21.04%	30.76	21.04	13.39	96.44%	95.32	96.44	95.82	**91.27**	**92.32**	**92.04**
80%	10.38%	54.69	10.38	11.79	99.68%	95.03	99.68	97.28	**92.53**	**92.96**	**94.74**
90%	00.00%	00.00	00.00	00.00	100.0%	94.74	100.0	97.29	**92.20**	**95.04**	**94.74**

Table 4.27: The results of using Rezoom + SGD and LSTM + SGD

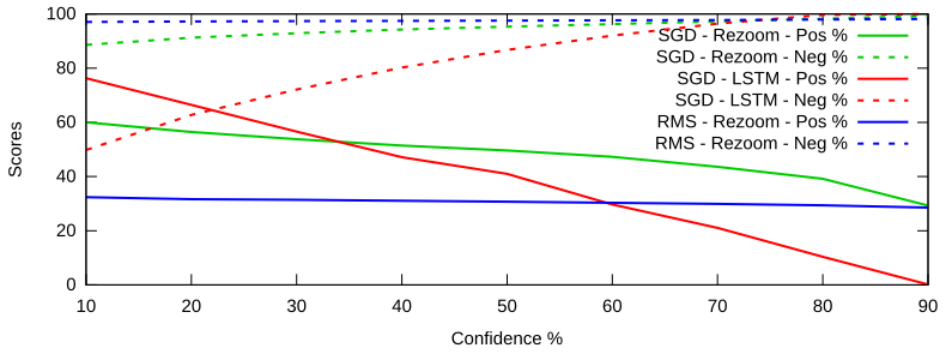


Figure 4.24: Graph of the results using Rezoom + SGD and LSTM + SGD

By looking at the results in table 4.27 and figure 4.24, we can see that LSTM produces the bigget spread in both positive and negative recall. The positive recall varies from 76.30% all the way to 0%, for 10% to 90% confidence, which makes LSTM unable to detect a single polyp at 90% confidence. The negative recall also varies, but less drastically, where the highest is 100% and lowest is 49.82%. None of the confidences show a result that provides a good enough balance to beat the results from other versions.

When looking at the version with Rezoom, it clearly provides a much better balance. Spreads are also present here, but to a lesser degree, making it easier to find a good balance for most scenarios. In a scenario where the most important aspect is to detect as many polyps as possible, it can provide a detection rate of over 50% while maintaining negative rate of around 90%. In other scenarios, where the FPs need to be avoided, a negative rate of around 99% can be provided while also providing a positive rate of over 30%.

When comparing the Rezoom + SGD results to those of Rezoom + RMS in table 4.24, SGD provides a much higher positive recall and also a higher negative recall. For all scenarios, SGD is clearly able to outperform RMS.

From this, we conclude Rezoom + SGD to better than both Rezoom + RMS and LSTM + SGD in every aspect, because it provides the best compromise between positive and negative recall. We therefore decided to only use Rezoom + SGD for future experiments, and thus leaving out LSTM.

4.5.4.2 Adam

In this experiment, Adam was used as the optimizer to compare against RMS and SGD.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Rezoom + Adam											
10%	43.21%	11.31	43.21	17.27	78.13%	96.18	78.13	86.16	**82.66**	**91.84**	**76.32**
20%	42.65%	11.65	42.65	17.61	79.08%	96.19	79.08	86.73	**83.23**	**91.88**	**77.19**
30%	42.22%	11.91	42.22	17.87	79.79%	96.20	79.79	87.17	**83.66**	**91.90**	**77.84**
40%	41.80%	12.16	41.80	18.10	80.41%	96.21	80.41	87.54	**84.03**	**91.92**	**78.42**
50%	41.42%	12.44	41.42	18.36	81.05%	96.21	81.05	87.92	**84.40**	**91.94**	**79.00**
60%	40.65%	12.71	40.65	18.57	81.84%	96.20	81.84	88.38	**84.85**	**91.96**	**79.71**
70%	39.50%	13.11	39.49	18.83	82.84%	96.18	82.84	88.95	**85.41**	**91.96**	**80.59**
80%	38.03%	14.29	38.03	19.67	84.50%	96.16	84.51	89.89	**86.36**	**92.03**	**82.09**
90%	35.95%	15.78	35.94	20.65	86.81%	96.15	86.81	91.18	**87.65**	**92.12**	**84.17**

Table 4.28: The result of using Rezoom + Adam

The results in table 4.28 and figure 4.25 show that Adam is able to produce a higher positive recall than RMS, found in table 4.24, but has to sacrifce some negative recall to achieve this. The increase in positive recall

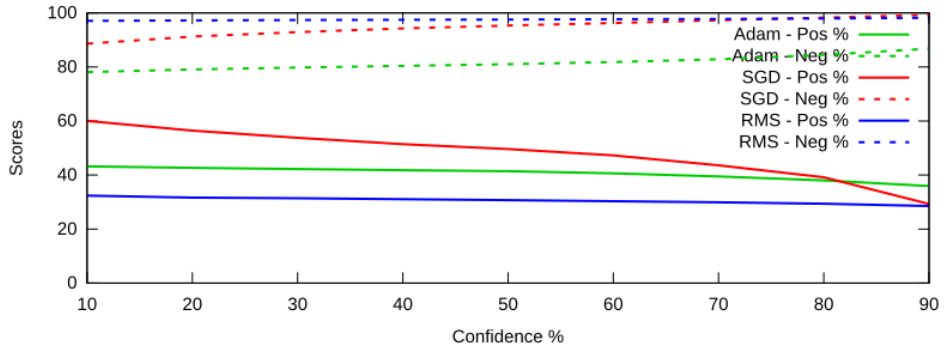


Figure 4.25: Graph of the results using Rezoom + Adam

is between 7.5% - 11.5%, but decrease the negative recall by between 11.5% - 19%. This trade-off could be worth it if the negative recall is less important, but SGD produces equal or better positive recall, while at the same time maintaining a significantly higher negative recall. SGD is able to beat both the highest positive and negative recall of Adam simultaneously, making SGD the better option for all use-cases, regardless of the needed focus.

We have now experimented with three different optimizers, RMS, SGD and Adam, and conclude that SGD outperforms the others in both positive and negative recall. Therefore, only SGD is used as the optimizer in the next experiment.

4.5.5 Combining optimized training and data enhancements

We have experimented with different training optimizations, and found out that a combination of RNN, a low negative dataset, Rezoom and SGD produces the best results. We will now combine this with the two most interesting data enhancement methods from section 4.4. This is done to see if the data enhancements affect the optimized training in a similar fashion as the default setup used during the data enhancement experiment. R is interesting as it showed a good positive recall combined with a relatively high negative recall, making it the only enhancement to produce a higher weighted F1-score than NP. RC were not able to produce a higher weighted F1-score in the same way, but is interesting since it had the highest positive recall of all the enhancements.

The two data enhancement methods combined with optimized training, as can be seen in table 4.29 and figure 4.26, produces relatively parallel results. At 90% confidence, the difference in positive and negative recall is roughly 1-2%, which steadily grows to 5-6% at 10% confidence, with R always having the highest percentages. As such, R outperforms RC in every aspect and for every confidence, and is therefore the one we discuss and compare with the previous results in table 4.27.

Optimized training combined with R at 10% confidence has a negative recall of 81.21%, 7.39% lower than the 88.60% of the no data enhancement version. However, when positive and negative recall is considered together, R is able to produce a negative recall of around 95% and still

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Optimized training + RC											
10%	71.62%	13.36	71.62	22.23	74.22%	97.99	74.22	84.45	**81.27**	**93.61**	**74.11**
20%	67.59%	15.63	67.59	25.01	79.84%	97.87	79.84	87.91	**84.71**	**93.63**	**79.22**
30%	64.63%	17.88	64.63	27.50	83.64%	97.77	83.64	90.12	**86.95**	**93.67**	**82.66**
40%	61.72%	20.30	61.72	29.88	86.70%	97.67	86.70	91.82	**88.69**	**93.71**	**85.40**
50%	58.61%	23.25	58.61	32.41	89.41%	97.55	89.41	93.28	**90.21**	**93.77**	**87.81**
60%	54.86%	26.98	54.86	35.14	91.95%	97.39	91.95	94.57	**91.58**	**93.84**	**90.01**
70%	49.73%	32.19	49.73	37.85	94.45%	97.17	94.45	95.77	**92.86**	**93.93**	**92.10**
80%	42.33%	39.89	42.33	39.90	96.91%	96.83	96.91	96.86	**93.98**	**94.03**	**94.05**
90%	26.42%	48.09	26.42	32.48	97.71%	96.03	97.71	96.83	**93.54**	**93.74**	**93.98**
Optimized training + R											
10%	76.90%	18.22	76.90	29.08	81.21%	98.44	81.21	88.97	**85.91**	**94.29**	**80.97**
20%	72.64%	22.42	72.64	33.65	86.45%	98.27	86.45	91.95	**88.99**	**94.37**	**85.69**
30%	69.33%	26.56	69.33	37.54	89.73%	98.13	89.73	93.72	**90.88**	**94.48**	**88.62**
40%	66.25%	30.66	66.25	40.79	92.05%	98.00	92.05	94.91	**92.18**	**94.59**	**90.65**
50%	63.06%	35.29	63.06	43.82	93.96%	97.85	93.96	95.85	**93.23**	**94.71**	**92.29**
60%	59.44%	40.30	59.44	46.25	95.48%	97.67	95.48	96.55	**94.01**	**94.82**	**93.53**
70%	53.61%	47.25	53.61	47.98	97.00%	97.38	97.00	97.18	**94.68**	**94.93**	**94.66**
80%	44.66%	55.34	44.66	46.60	98.26%	96.90	98.26	97.56	**94.92**	**94.91**	**95.35**
90%	27.58%	65.81	27.58	35.38	99.38%	95.97	99.38	97.63	**94.28**	**94.56**	**95.45**

Table 4.29: The results of combining optimized training with optimal data enhancement methods

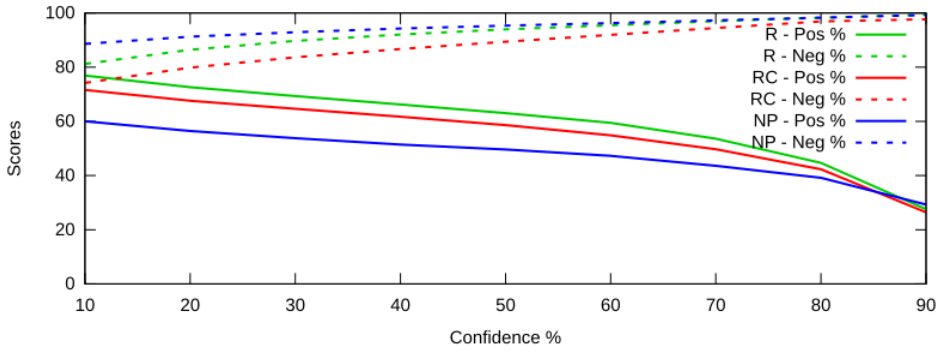


Figure 4.26: The graph of the results from combining optimized training with optimal data enhancement methods

produce a positive recall similar to the version with no data enhancement. The same is true in the opposite direction, where matching a negative recall will yield a higher positive recall when using R. This shows that R is able to outperform the version with no data enhancement in every aspect by choosing an optimal confidence.

4.5.6 Summary

In this section, we have performed experiments with various training optimizations. We began by performing a limited experiment with different types of neural networks where Inception and Resnet were tested to see the effects of the different types of networks. They both produced very high positive recall, but at the same time very low negative recall, making them unoptimal for most common polyp detection scenarios. Due to this, we did not perform in-depth experiments with Inception and Resnet, and decided to keep the default neural network.

The next optimization persued was with different balances of positive and negative samples in the dataset. This was done to see whether a more positive focused dataset could improve the detection rate. We tested a balanced dataset, with around 50-50 split between the positive and negative samples, and a low negative dataset, with no explicit negative videos. The balanced dataset increased the positive recall by around 2% and decreased the negative recall by around 0.1-0.2%, making it a slight overall improvement. The low negative dataset achieved roughly the same positive recall as the balanced dataset and the negative recall of the full dataset. The low negative dataset is thus the best of both worlds, and is the reason this balance was used in the following experiments.

We then experimented with different training techniques, consisting of LSTM and Rezoom. First a limited experiment was performed with the different combinations since LSTM and Rezoom can both be enabled and disabled, resulting in four different combinations. LSTM produced the highest positive recall while Rezoom produced the highest weighted F1-score, which is why those versions were used for the next step.

The next step was to examine how different optimizers affect the results. RMS, SGD and Adam were compared, where RMS has been used as the optimizer for all experiments until now. SGD was run with both LSTM and Rezoom from the previous step, where Rezoom and SGD produced the better results, therefore only Rezoom was tested with Adam. The results of SGD saw the highest increase in positive recall without a large sacrifice in negative recall, almost doubling its positive recall from 32.38% to 60.04%. Adam produced both lower positive and negative recall compared to SGD with no other benefits, and as such SGD was the optimizer used in the last experiment.

For the last experiment, we combined the optimized training with the most optimal data enhancements from section 4.4. From the experiments with data enhancements, R produced the highest weighted F1-score and RC the highest positive recall, which is why R and RC were used in this experiment. In combination with SGD, both produced similar results, with R slightly higher than RC in every aspect for every confidence. By applying R, improvements can be had, especially in positive recall, with little no no decrease in negative recall. This shows that data enhancement and training optimization should be combined to achieve optimal results.

4.6 A higher number of training iterations

Initially, before the experiments with the different data enhancement methods, we conducted some small experiments to determine the necessary number of training iterations. From these experiments, we concluded that 500k iterations lead to stable results and fit our time constraints.

Version	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
100k	11.61%	10.20	11.61	10.86	95.44%	96.03	95.44	95.73	**92.11**	**92.37**	**91.86**
200k	20.32%	14.37	20.32	16.83	94.60%	96.38	94.60	95.48	**92.12**	**92.88**	**91.43**
300k	19.35%	14.78	19.35	16.76	95.02%	96.35	95.02	95.68	**92.31**	**92.87**	**91.79**
400k	20.48%	14.82	20.48	17.20	94.75%	96.39	94.75	95.56	**92.22**	**92.91**	**91.58**
500k	19.84%	14.68	19.84	16.87	94.86%	96.37	94.86	95.61	**92.25**	**92.88**	**91.66**
600k	20.16%	14.88	20.16	17.12	94.86%	96.38	94.86	95.61	**92.26**	**92.90**	**91.67**
700k	20.16%	14.88	20.16	17.12	94.86%	96.38	94.86	95.61	**92.26**	**92.90**	**91.67**
800k	20.16%	14.88	20.16	17.12	94.86%	96.38	94.86	95.61	**92.26**	**92.90**	**91.67**
900k	20.16%	14.88	20.16	17.12	94.86%	96.38	94.86	95.61	**92.26**	**92.90**	**91.67**
1000k	20.16%	14.88	20.16	17.12	94.86%	96.38	94.86	95.61	**92.26**	**92.90**	**91.67**

Table 4.30: The results of using 1 million training ieterations

The experiment to determine the amount of training were performed using non-preprocessed data. Two of our data enhancement methods, rotation and brightness variations, increased the number of polyp images up to 16 times. During the experiment with different data enhancements, we noticed that the methods with additional data also stabilized around 300k, with very slight variations up to 500k. We were curious if the extra data would produce variations in the results after 500k, so we decided to conduct an experiment with a combination of rotation, brightness variations, masking reflections and contrast enhancement all the way to 1 million iterations.

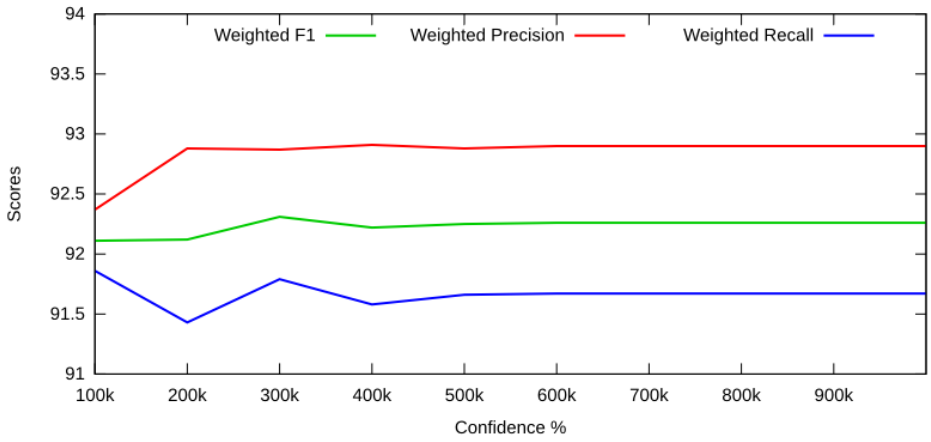


Figure 4.27: Graph of the results using 1 million training iterations

As can be seen in table 4.30 and figure 4.27, the last iteration with any changes was 600k, which was a slight increase of two frames in TPs from 500k. From this, we can conclude that the added data does not require any substantial amount of extra training.

4.7 Evaluation against external dataset

Towards the end of the thesis, we got access to a new dataset [39]. We assess the systems general understanding of polyps by evaluating existing weights trained with the ASU Mayo Clinic dataset [56] against the new dataset, which can be seen in table 4.31.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
10%	77.60%	58.70	77.60	66.84	45.40%	66.96	45.40	54.11	**60.48**	**62.83**	**61.50**
20%	73.40%	60.36	73.40	66.24	51.80%	66.07	51.80	58.07	**62.16**	**63.22**	**62.60**
30%	68.80%	61.10	68.80	64.72	56.20%	64.30	56.20	59.98	**62.35**	**62.70**	**62.50**
40%	65.80%	62.55	65.80	64.13	60.60%	63.92	60.60	62.22	**63.18**	**63.24**	**63.20**
50%	61.60%	64.44	61.60	62.99	66.00%	63.22	66.00	64.58	**63.79**	**63.83**	**63.80**
60%	54.80%	65.87	54.80	59.83	71.60%	61.30	71.60	66.05	**62.94**	**63.59**	**63.20**
70%	49.60%	68.51	49.60	57.54	77.20%	60.50	77.20	67.84	**62.69**	**64.51**	**63.40**
80%	41.20%	73.05	41.20	52.69	84.80%	59.05	84.80	69.62	**61.16**	**66.05**	**63.00**
90%	21.20%	77.37	21.20	33.28	93.80%	54.35	93.80	68.82	**51.05**	**65.86**	**57.50**

Table 4.31: The results from evaluation against the external dataset [39]

The performance is surprisingly good considering how different the polyps in the ASU Mayo and the new dataset are. As shown in earlier examples, the polyps in the ASU Mayo dataset are usually small protrutions on the surface, while the polyps in the new dataset are often highly clustered and covers a greater part of the frame. Since the system is trained with no samples resembling the new types of polyps and sizes, detecting such polyps are challenging. Even with such problems, the system is able to achieve a decent result with a positive recall of between 21% and 77% depending on the confidence. The balance in the new dataset is vastly different from the ASU Mayo dataset, which means that there is no point in comparing weighted scores.

In figure 4.28, we show two examples of detected polyps. The polyps are similar in texture, color, shape, and size to the polyps in the ASU Mayo dataset, making detection possible.

In figure 4.29, we show two examples of undetected polyps. They show a high degree of clustering, different textures and colors, and also different shapes and sizes compared to those in the ASU Mayo dataset, making detection unlikely. The results could be improved by including samples of both datasets as part of training, but since the new dataset has not yet been annotated, this was not an option.

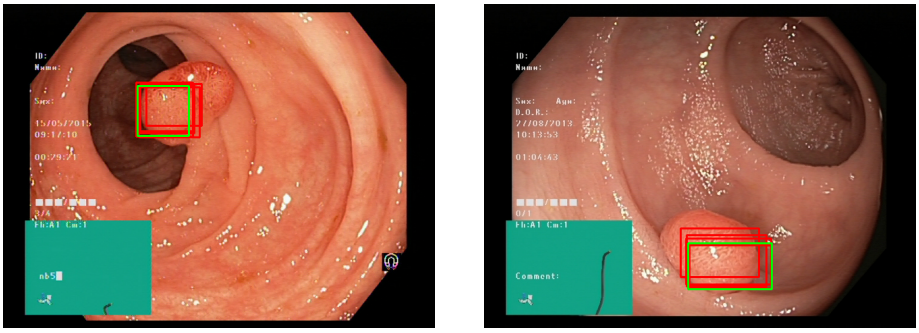


Figure 4.28: Polyps the system are able to detect

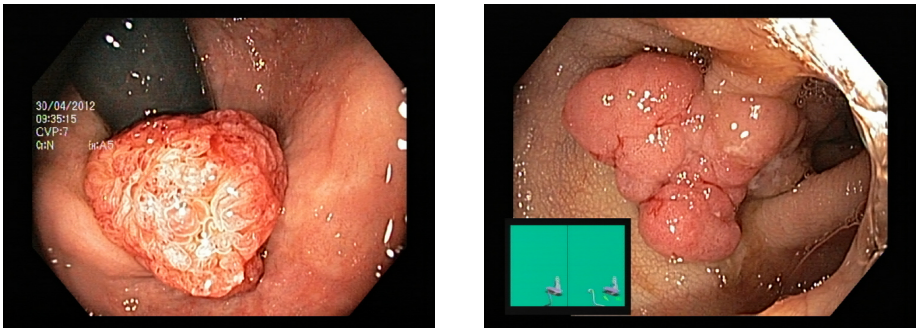


Figure 4.29: Polyps the system are unable to detect

4.8 Discussions

In the experiments performed, a number of topics were uncovered which deserve their own discussion. We therefore discuss these topics in greater detail in this section.

4.8.1 Time requirements for training and evaluation

Neural networks requires a substantial amount of time to train in order to gain enough knowledge. This is not necessarily a problem, because it only needs to be trained once. However, if one needs to make small modifications to optimize training, it needs to be trained, evaluated and the results analyzed to assess the results of each modification. Depending on the dataset size and hardware, this could potentially take days or even weeks. As an example, training times we have experienced can be seen in table 4.32.

TensorFlow likes powerful graphics cards with high amount of memory and TFLOPS. We can see that the reduction in training time follows power and memory in the GPUs to a certain degree. The GPUs we have used are among of the most powerful consumer cards on the marked, and even using these cards, the time it takes to train is still substantial. For our thesis, we have performed more than 120 training runs in total to optimize and

Machine	GPU	Training time
Machine 1	NVIDIA GTX 1080 8GB — 8.8 TFLOPS	13.5 hours
Machine 1	NVIDIA GTX 1080 TI 11GB — 11.3 TFLOPS	9.5 hours
Machine 2	NVIDIA TITAN 6GB — 4.5 TFLOPS	20.5 hours

Table 4.32: Approximate training time on different hardware for 500k training iterations

conduct experiments, taking upwards of 2 000 hours, including evaluation.

Evaluation has to be performed on each video to detect polyps. For some scenarios, like ACD, the videos would normally be evaluated after the fact. While in others, like CAD, the evaulation runs in real time during an examination. In the latter scenarios, evaluation speed is key factor as the evaluation needs to be able to handle incoming data as fast as it arrives.

During the evaluations, we have measured evaluation times for around 15 000 frames to 15 minutes for GTX 1080 TI, 20 minutes for GTX 1080 and 30 minutes for GTX TITAN. This gives a frames per second (FPS) count of 16.6, 12.5 and 8.3, respectively. The FPS numbers are below normal requirments for real time systems, which often are 30 or 60 FPS. But this is expected, as TensorBox is not built as a real time system. If it were to be used as part of a realtime system, TensorBox would need to be modified to increase performance.

4.8.2 Video quality differences and data enhancement effects

During the experiments with different data enhancement methods, only rotation was able to show any kind of improvements compared to the non-preprocessed version in weighted F1-score averaged over all splits.

When inspecting the F1-score of individual splits, every data enhancement method except brightness, and every combination except RB and RBMC, are able to improve the score in one or two of the splits. The issue is that no combination of data enhancements are able to increase the score in more than two splits, and that the other 3 or 4 splits usually see a decrease. The result of this is that the positive effect in some splits are negated by the others, producing a slightly lower average score. These results can be seen in table 4.33.

A reason for the data enhancement methods producing such variable results between each split is that the videos are of different quality. Some videos contain polyps that are easily detectable, while others contain polyps that almost blend into its environment. The differences in image quality in the different videos also differs, such as resolution, motion blur and focus problems. These problems lead to a high variation in positive recall, which can be seen in table 4.34.

An example of polyps that blend into their environments can be seen in figure 4.30. If the polyp has a visible contour that are separable from other parts of the colon, data enhancement methods such as contrast enhancement might be able to increase the contour, making the polyp

Name	Split 1	Split 2	Split 3	Split 4	Split 5	Average
NP	95.13	92.20	90.36	97.00	93.12	93.56
R	94.96	92.12	89.46	**97.33**	**94.21**	**93.61**
B	90.85	92.02	89.74	95.82	90.81	91.85
RB	93.56	91.85	89.28	96.40	92.83	92.78
M	**95.85**	91.89	**90.40**	95.40	92.45	93.20
C	**95.70**	91.89	**90.37**	96.22	92.24	93.28
MC	**96.30**	91.79	90.08	96.08	92.98	93.45
RBM	93.25	**92.30**	88.08	96.41	91.22	92.25
RBC	91.18	**92.53**	87.39	95.06	90.97	91.43
RBMC	91.17	92.01	87.97	95.51	92.17	91.77
RM	94.55	**92.67**	89.00	96.56	**93.44**	93.24
RC	94.22	**93.11**	88.50	95.47	**94.11**	93.08
RMC	93.38	**92.30**	87.86	95.20	**93.33**	92.41

Table 4.33: Weighted F1-score for all splits and data enhancement methods. The data enhancement methods which improve the score, and the best for each split, has been highlighted

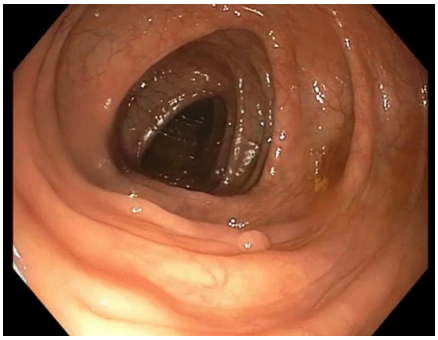
Split	Positive recall	Negative recall
Split 1	66.32	97.66
Split 2	4.44	99.01
Split 3	0.22	98.32
Split 4	48.23	98.41
Split 5	12.26	97.18
Average	26.29	98.14

Table 4.34: Positive and negative recall for the different splits using NP data

easier to distinguish, which can be seen in figures 4.30a and 4.30b. If the polyp is part of a fold in the colon, the data enhancement methods are unable to increase visibility in a meaningful way. An example is shown in figures 4.30c and 4.30d.

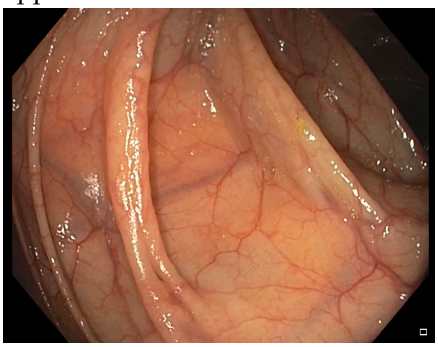
An example of different video quality is shown in figure 4.31. The image on the left is from split 1, where the detection is the highest. One of the reason for this, is that the quality is mostly high with clear high resolution images, and low amounts of motion blur and focus problems. The image on the right is from split 3, where the detection is abysmal. It suffers from a great deal of motion blur and focus problems, making detection nearly impossible.

With most images having a quality somewhere between the left and right image of figure 4.31, but suffering from different types of problems, different data enhancement methods affect them differently. Applying a data enhancement method to improve polyp detection in one video, could very well have a negative effect on another. It is therefore hard to apply filters that improve polyp detection across a range of videos unless they posses the same qualities.



(a) Image with no data enhancement applied

(b) Image with masking reflections and contrast enhancement

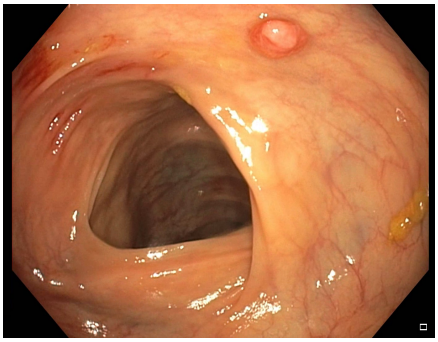


(c) Image with no data enhancement applied

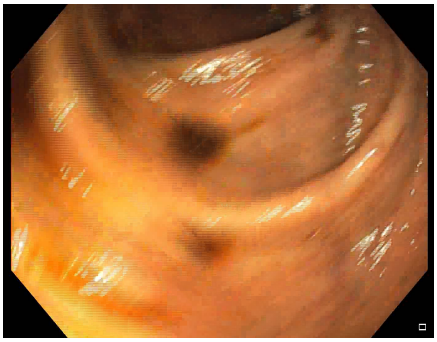
(d) Image with masking reflections and contrast enhancement

Figure 4.30: Effectiveness of data enhancement on different polyps

From this, we can see that the systems performance is highly dependent on the quality of the videos. Image quality in small form factor cameras continiues to improve yearly. By improving video quality, the detection rate would be improved as well.



(a) An image with good quality from split 1



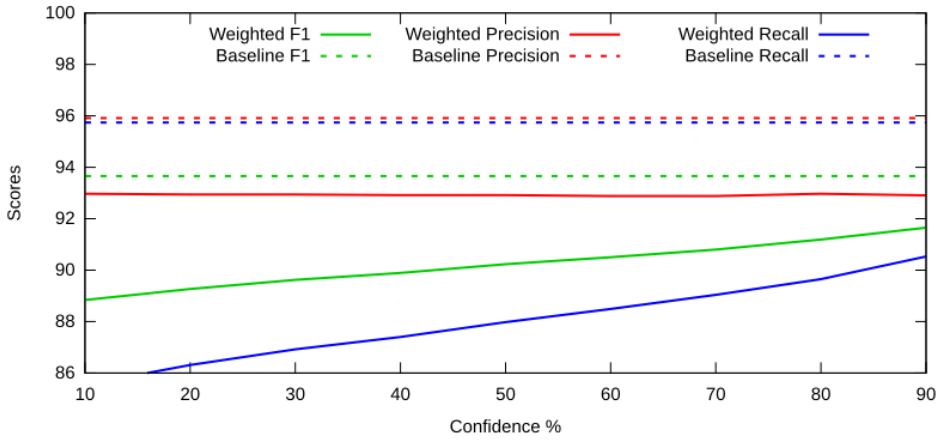
(b) An image with bad quality from split 3

Figure 4.31: Illustration of different video qualities in different videos

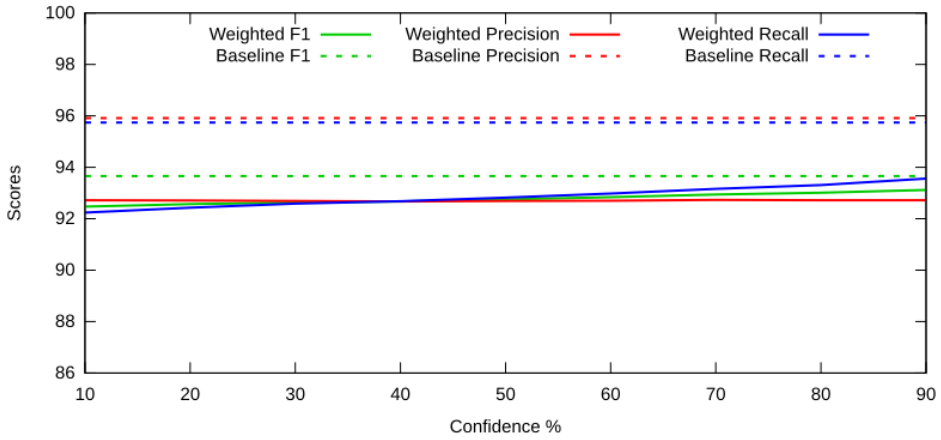
4.8.3 Training iterations and confidences

As seen during section 4.4, the network is very unstable before 300k iterations have been reached, as this is the phase where the network makes major adjustments to its weights. From 300k until 500k iterations, the network stabilizes to a larger extent, and while it is still adjusting its weights, the changes in the results become minor. In section 4.6, it was demonstrated that iterations beyond 500k iterations have no real impact.

The reason a network will often fluctuate initially, is that it is less sure about its classifications until additional iterations have been executed. The confidence of the network for its classifications increases gradually, and will thus result in less fluctuations.



(a) The confidence spread after 100k iterations



(b) The confidence spread after 500k iterations

Figure 4.32: How the confidence spreads between 100k and 500k iterations for split 5

Every classification a network performs is accompanied by a confidence, a number between 0 and 1 indicating how sure the network is of the existence of a polyp within the image. A low confidence number indicates a certainty about the non-existence of a polyp, while a high confidence indi-

cates a certainty about the existence of a polyp. When evaluating, a certain confidence number needs to be used as to what will be defined as a polyp, for instance, using 0.9 would mean that the classification needs to have a 90% certainty before we classify it as an image containing a polyp.

During the experiments with different data enhancement methods in section 4.4, we discovered that networks trained for 100k iterations produced a high spread of confidences. A comparison of confidence variations between 100k and 500k iterations is shown in figure 4.32. With many confidence numbers towards the middle of the spectrum, the results become more randomized. When training the network further, the confidences were to a higher degree either close to 0 or 1.

When requiring a confidence of 0.9 in evaluation, the number of FPs are kept relatively low, but at the same time some TPs may be missed. By changing the required confidence, it is possible to trade FPs for TPs. A general balance does not exist, as different scenarios may focus on different aspects. Some scenarios may prioritize the detection of polyps and accept FPs, while others may require a high positive precision and thus reduce FPs as much as possible. Two such scenarios are discussed during the next section.

4.8.4 Real world scenarios

Two important real world scenarios in polyp detection are CAD and ACD, as described in sections 2.1.4 and 2.1.5. Both obviously have a need for a high positive and negative recall, but these work as opposite forces. By increasing the positive recall, the negative recall will usually decrease as a consequence, and vice versa. Thus, it becomes vital to be able to balance the positive and negative recall in a way that suits the scenario in which the system is used in. For instance, CAD is mainly a support system used in addition to a medical professional, and as such a lower negative recall could be acceptable since the doctor can discard false positives during the examination. ACD, on the other hand, is more independent in nature, with no aid from doctors during analyzis, for example an automated screening processes combined with WCE. In such a scenario, FPs for healthy individuals will require manual examination, negating some of the purpose of the automatic screening process. Because of this, automatic screening processes require a very high negative recall, and also a high enough positive recall.

The two scenarios above illustrate that different balances are required for different scenarios. The results of the overall best performing combination are listed in table 4.35, and are used to discuss how they could fit the two scenarios.

CAD has the aid of doctors to discard FPs, which puts less empahis on negative recall, making positive recall the priority. As such, a lower confidence would probably be the most suitable, with a positive recall of 66% or higher and still a negative recall of around 90%. If a negative recall of 80% is acceptable, the positive recall can be upwards of 77%.

ACD, as previously explained, is more reliant on a high negative recall.

Confidence	Positive %	Positive precision	Positive recall	Positive F1	Negative %	Negative precision	Negative recall	Negative F1	**Weighted F1**	**Weighted Precision**	**Weighted Recall**
Optimized training + R											
10%	76.90%	18.22	76.90	29.08	81.21%	98.44	81.21	88.97	**85.91**	**94.29**	**80.97**
20%	72.64%	22.42	72.64	33.65	86.45%	98.27	86.45	91.95	**88.99**	**94.37**	**85.69**
30%	69.33%	26.56	69.33	37.54	89.73%	98.13	89.73	93.72	**90.88**	**94.48**	**88.62**
40%	66.25%	30.66	66.25	40.79	92.05%	98.00	92.05	94.91	**92.18**	**94.59**	**90.65**
50%	63.06%	35.29	63.06	43.82	93.96%	97.85	93.96	95.85	**93.23**	**94.71**	**92.29**
60%	59.44%	40.30	59.44	46.25	95.48%	97.67	95.48	96.55	**94.01**	**94.82**	**93.53**
70%	53.61%	47.25	53.61	47.98	97.00%	97.38	97.00	97.18	**94.68**	**94.93**	**94.66**
80%	44.66%	55.34	44.66	46.60	98.26%	96.90	98.26	97.56	**94.92**	**94.91**	**95.35**
90%	27.58%	65.81	27.58	35.38	99.38%	95.97	99.38	97.63	**94.28**	**94.56**	**95.45**

Table 4.35: The best results achieved

Therefore, a higher confidence would likely be the better choice. For instance, a 99.38% negative recall can be achieved while still detecting almost 30% of the frames that contain a polyp. If the negatice recall can be slightly lowered, a much higher positive recall can be achieved.

The positive recall can also be further improved by considering polyps or sequences instead of frames, which is discussed in the next section. This combined with object tracking could potentially lead to a detection rate of close to 100%.

4.8.5 Per polyp and per sequence versus per frame detection

In this thesis, we do per frame detection, which means we examine every frame individually with no video context. For instance, if three sequential frames contains a polyp, and we only detect one, the detection rate would be 33.3%. For medical professionals, detecting every polyp is much more important than detecting every frame that contain a polyp. This is why Polyp-Alert [60] does per polyp detection, where a single detection of a polyp counts a polyp as detected, even if it is detected in only 1 of 100 frames. This can also be combined with object tracking, where polyps are tracked in preceding and subsequent frames.

In this section, we will see how our system performs with per sequence and per polyp detection. Each polyp-video in our dataset contains a single polyp, but may not be part of every frame as the camera moves. We define a sequence as all continuous frames that contain the polyp, where gaps in the ground truth divides sequences. A sequence is considered detected if one or more frames in the sequence are detected. We define a polyp as a physical polyp, where we consider it detected if one or more of its sequences are detected.

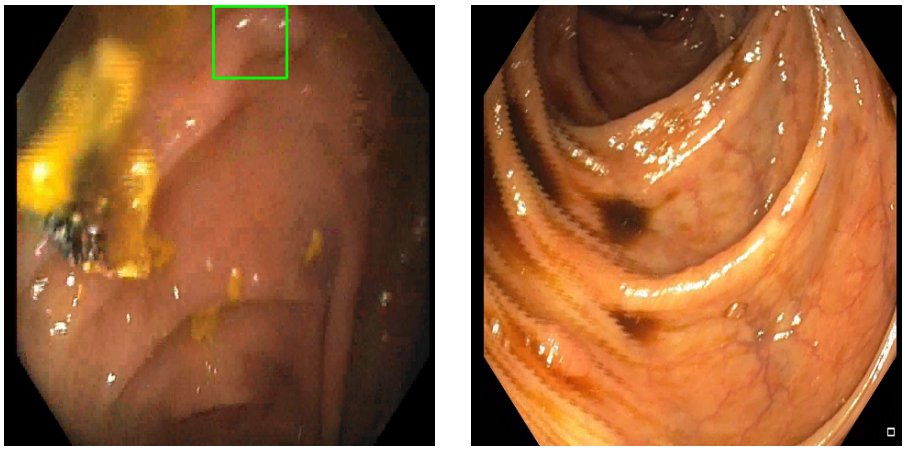
When we consider our results on a per sequence or a per polyp basis, as shown in table 4.36, we have a high degree of sequence detection and a 100% polyp detection rate. All sequence are detected using 10% and 30% confidence, and only a few are missed using 50% and 70%.

Confidence	Sequences	Detected	Detection %	Polyps	Detected	Detection %
10%	43	43	100 %	10	10	100 %
30%	43	43	100 %	10	10	100 %
50%	43	41	95.3 %	10	10	100 %
70%	43	39	90.7 %	10	10	100 %
90%	43	32	74.4 %	10	10	100 %

Table 4.36: Detection rate per sequence and per polyp for all videos

All detections for every video using a confidence of 10%, 30%, 50%, 70% and 90% are displayed in figure 4.34 together with the ground truth that indicates which frames contain a polyp. The figures show how the detection is spread throughout the videos, where both TP, FP, TN and FN can be read. They clearly show how different confidences affect detection, how they "evolve", and how lowering the confidence increases detection, but also produces additional FPs.

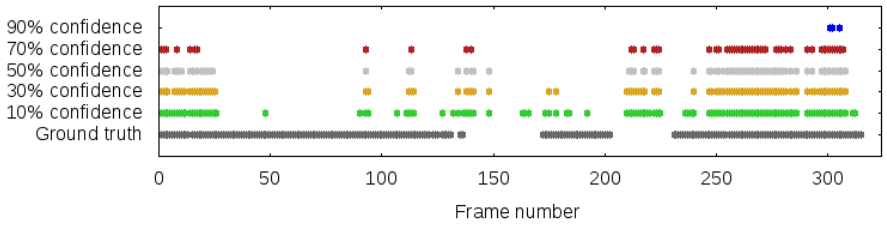
They also show which sequences that are hard to detect. For instance, `wp_69` contains 8 seperate sequences, which indicates a lot of camera movements where the polyp rapdily enters and leaves the frame. This usually leads to heavy motion blur, which makes the polyp more challenging to detect. In a similar manner, they show which parts of the videos that are mistaken as containing polyps. Analyzing the figures together with the videos could yield valuable information for optimizating the system, in order to increase both TPs and TNs, and decrease FPs and FNs.



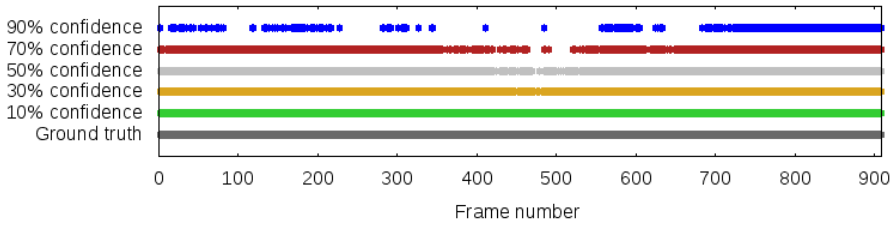
(a) FP from wp69, where the structure is mistaken as a polyp

(b) FN from wp61, where the polyp is in the lower left corner

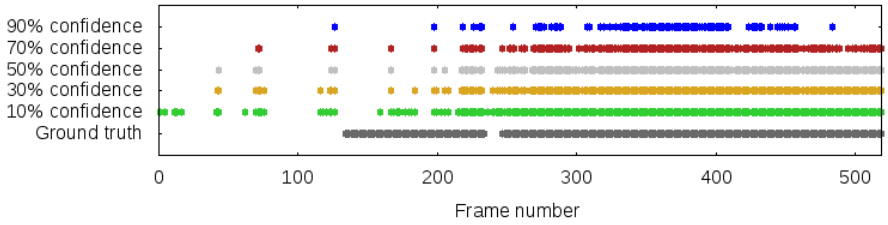
Figure 4.33: Example of a FP and FN where all confidences make the same mistakes



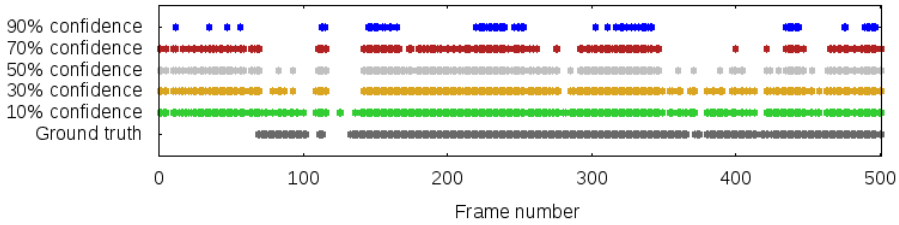
(a) wp2 — used for evaluation in split 1



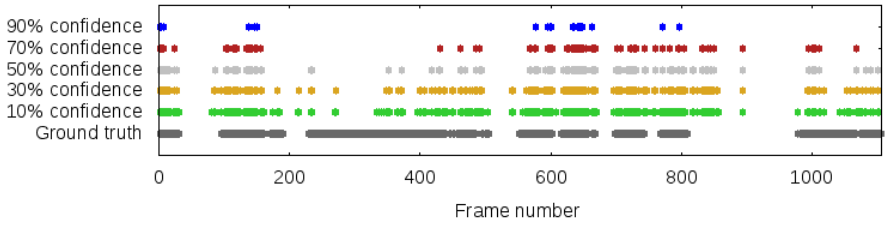
(b) wp4 — used for evaluation in split 1



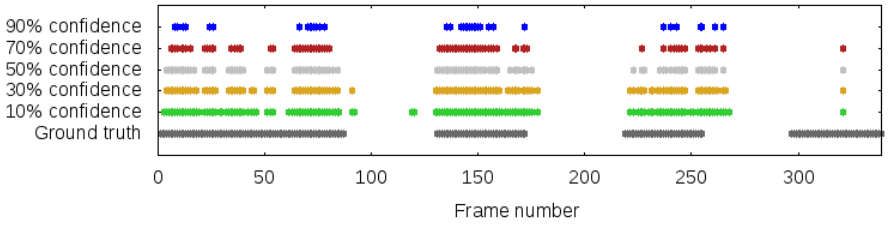
(c) wp24 — used for evaluation in split 2



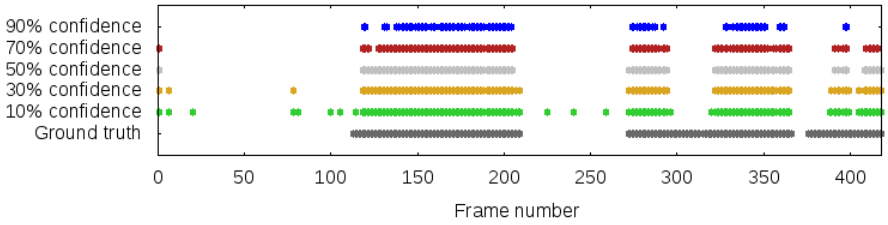
(d) wp49 — used for evaluation in split 2



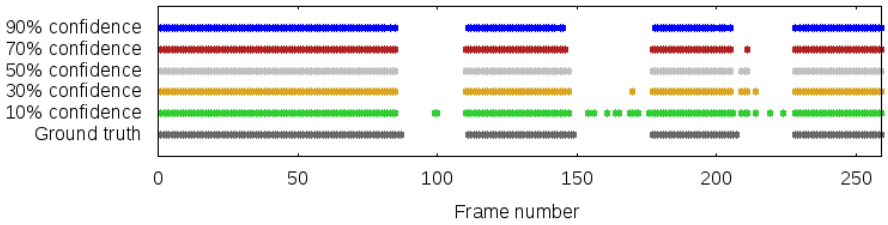
(e) wp52 — used for evaluation in split 3



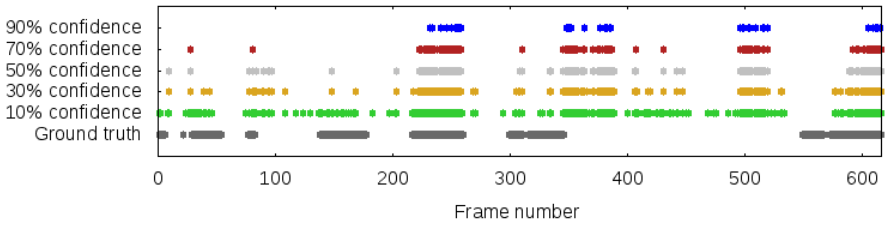
(f) wp61 — used for evaluation in split 3



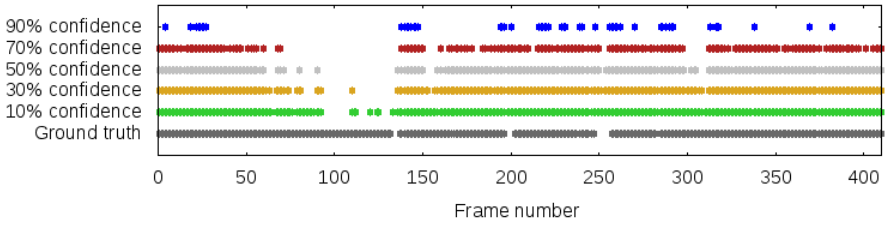
(g) wp66 — used for evaluation in split 4



(h) wp68 — used for evaluation in split 4



(i) wp69 — used for evaluation in split 5



(j) wp70 — used for evaluation in split 5

Figure 4.34: Polyp detection plot for all polyp-videos using various confidences

An example of both a FP and FN can be found in figure 4.33. The FP in figure 4.33a is from the latter part of `wp_69`, where all confidences find FPs. By looking at the video, we can see that the FPs are due to a structure similar to that of a polyp in the top of the image. Upon manual inspection of the video, we also find it hard to distinguish the structure from a polyp. The FN in figure 4.33b is from the end of `wp_61`, where there is a lot of camera movement that causes motion blur. This, combined with a polyp without a clear contour, makes the polyp very challenging to detect. Both of these problems would be hard to solve with data enhancements, likely requiring cameras producing better quality videos as previously discussed in section 4.8.2.

4.8.6 Comparison with other systems

Publication/ System	**Positive Recall**	**Positive Precision**	**Negative Recall**	**Negative Precision**	**Dataset size**
Wang et al. [60]	97.7%*	-	95.7%	-	1 800 000 images
Wang et al. [61]	81.40%	-	-	-	1 513 images
Mamonov et al. [31]	47%	-	90%	-	18 968 images
Hwang et al. [20]	96%	83%	-	-	8 621 images
Li et al. [27]	95.07%	-	93.33%	94.20%	300 images
Li and Meng [29]	88.60%	-	96.20%	92.40%	-
Zhou et al. [66]	75%	-	95.92%	90.77%	-
Alexandre et al. [3]	93.69%	-	76.89%	-	35 images
Cheng et al. [8]	86.20%	-	-	-	74 images
Ameling et al. [5]	AUC=95%***	-	-	-	1 736 images
EIR [45, 46]	98.50%	93.88%	72.49%	87.70%	18 781 images
Optimal data enhancement and training optimization — per frame					
10% confidence	76.90%	18.22%	81.21%	98.44%	18 902 images
20% confidence	72.64%	22.42%	86.45%	98.27%	18 902 images
30% confidence	69.33%	26.56%	89.73%	98.13%	18 902 images
40% confidence	66.25%	30.66%	92.05%	98.00%	18 902 images
50% confidence	63.06%	35.29%	93.96%	97.85%	18 902 images
60% confidence	59.44%	40.30%	95.48%	97.67%	18 902 images
70% confidence	53.61%	47.25%	97.00%	97.38%	18 902 images
80% confidence	44.66%	55.34%	98.26%	96.90%	18 902 images
90% confidence	27.58%	65.81%	99.38%	95.97%	18 902 images
Optimal data enhancement and training optimization — per sequence					
10% confidence	100.0%**	-	81.21%	-	18 902 images
30% confidence	100.0%**	-	89.73%	-	18 902 images
50% confidence	95.35%**	-	93.96%	-	18 902 images
70% confidence	90.70%**	-	97.00%	-	18 902 images
90% confidence	74.42%**	-	99.38%	-	18 902 images
Optimal data enhancement and training optimization — per polyp					
10% confidence	100.0%*	-	81.21%	-	18 902 images
30% confidence	100.0%*	-	89.73%	-	18 902 images
50% confidence	100.0%*	-	93.96%	-	18 902 images
70% confidence	100.0%*	-	97.00%	-	18 902 images
90% confidence	100.0%*	-	99.38%	-	18 902 images

* The sensitivity is based on the number of detected polyps.
** The sensitivity is based on the number of detected sequences.
*** Reported only area under the curve (AUC) instead of sensitivity.

Table 4.37: Performance comparison of our system against state-of-the-art systems

Here, we will compare our best results with data enhancement in combination with training optimization, which consists of RNN, a low negative dataset, Rezoom, SGD and rotation, against the state-of-the-art systems listed in Riegler's PhD Thesis [43]. The full results are listed in table 4.37.

We can see that our system, using per frame detection, achieves a lower positive recall and precision, but has a higher negative recall and precision. From the results per sequence, we see that the system is able to detect all sequences for some of the confidences, so per frame detection could likely be improved by implementing object tracking, which should be able to track the polyp within a sequence. This could detect polyps which were previously undetected, but could also lead to additional FPs if FPs were also tracked.

Using per sequence or per polyp detection, our system is able to outperform most, if not all, of the other systems in positive recall. This combined with the very high negative recall, makes us consider the performance of our system to be quite high.

4.9 Summary

In this section, we have conducted various experiments and discussed topics and questions that were uncovered during the experiments. We began by describing the testbeds and the dataset used for the experiments, and the evaluation metrics to measure performance.

The first experiment examined how different data enhancement methods affected detection rates. The data enhancement methods either increased the quantity or the quality of the dataset, by adding additional samples or by preprocessing existing samples. To increase the quantity, rotation and brightness variations were used. Rotation was able to improve the positive recall while almost maintaing the same negative recall as NP, slightly increasing the weighted F1-score. Brightness variations had adverse effects, decreasing both positive and negative recall, producing a clear decrease in the weighted F1-score. To increase the quality, masking reflections and contrast enhancement were used. Masking reflections was able to handle some reflections better than others, leading to mixed results for different videos, producing a slight decrease in weighted F1-score. The result in two of the splits showed its potential, which means a more advanced implementation could be able to improve results for all videos. Contrast enhancement had a similar pattern as masking reflections, where it was able to achieve an improvement in some splits, while other splits saw a decrease, resulting in a slight decrease in the weighted F1-score. The data enhancement methods were then combined too see if the combined results were greater than the sum of its parts. Of all combinations, R and RC proved to be the most interesting. R because of the higher weighted F1-score and RC because of the high positive recall.

The second experiment was conducted to optimize the training, where we tested the effect the different neural networks, dataset balances, training

techniques and optimizers had on detection. We first conducted limited experiments with Inception and Resnet. Both showed a high positive recall, but also a very low negative recall. We considered the low negative recall as being too limiting for most polyp detection scenarios, and as such we kept the default network for the following experiments. Then we experimented with two different balances of the dataset in order to shift the focus more towards polyps. The first balance used close to a 50-50 distribution between positive and negative samples, while the second was heavily weighted towards positive samples. They both showed improvements in overall performance, and was able to improve positive recall by 2-3%. The dataset with the lowest number of negative samples produced slightly better results, and was thus the one we used for the following experiments. To decide the best combination of training techniques, we began with a limited experiment to find the optimal combination of LSTM and Rezoom, as none, one and both of them can be used. LSTM was interesting as it had a high positive recall, and Rezoom as it had the highest weighted F1-score. A full experiment with LSTM was then performed, and compared to the previous experiments where Rezoom had been used. Both had their advantages, and were therefore both kept. Finally, we performed experiments to compare optimizers. SGD and Adam was compared to RMS, which had been used previously. SGD combined with Rezoom produced very good results, with a substantial increase in positive recall and only a minor decrease in negative recall. SGD combined with LSTM also produced a high positive recall, but because of a noticable lower negative recall only Rezoom was part of the next combination. Adam also saw an increase in positive recall, but the increase was smaller, and had a higher decrease in negative recall. SGD was therefore decided to be the best optimizer in our case.

The optimized training for us was thus RNN combined with Rezoom and SGD using a low negative dataset. We combined this with R and RC, the most interesting combinations from the data enhancement experiment, to see if combining both could achieve better results. The combination with R produced the best results, both in positive and negative recall, compared to all other results.

The third and forth experiments were of a smaller scale, where we examined the effect of additional training iterations and evaluated how our pre-trained system performed on a completely different dataset. The third experiment showed that additional training iterations did not have any noticable effect on the results, while the forth experiment showed that our system was able to show quite good performance, but that unseen versions of polyps were less likely to be detected.

Finally, we discussed interesting topics related to the experiments. The first topic was the time requirement of training and evaluating neural networks. The second was how different video qualities affected data enhancement methods and their results. The third discussed the concept of training iterations and confidences and how confidences can be used to make trade-offs between positive and negative recall. The forth considered how the results from the system could be used in real world scenarios such as CAD and ACD. The fifth discussed other detection metrics, like per

frame, per sequence, and per polyp detection, and their relevance in the medical field. During this discussion, we included plots of the detection rates per video with ground truths and five different confidences. The sixth and last was a comparison with state-of-the-art systems where our system was shown to perform well.

Chapter 5

Conclusion

In this chapter, we summarize the work presented in this thesis. We then list our main contributions, present ideas for future work and make some final remarks.

5.1 Summary

In this thesis, we have investigated if neural networks can be used in a polyp detection scenario. We have also examined how different data enhancements and training optimizations affect the polyp detection rate.

We began by researching previous work related to polyp detection systems and data enhancements. Among others, we took a look at Polyp-Alert and EIR, which are complete systems used as aids during examinations. Both are among the state-of-the-art systems used for comparison with our system towards the end.

We then proceeded to make a system, implemented as a pipeline, able to go from annotated videos to enhanced data and meta files used for training and evaluation. This is done by extracting frames from videos, scanning each frames corresponding annotation file for the ground truth, and storing the information in a meta file. Depending on parameters, data enhancements could be applied as part of the extraction.

Using this system, we experimented with different data enhancement methods and combinations. We began by performing an experiment with NP data and compared it against a majority class baseline, to show that the system had a good understanding of polyps. After that, we conducted 12 experiments with combinations of rotation, brightness variations, masking reflections and contrast enhancement, where we for each experiment listed the results in a table and graph, and discussed their performance. We found and outlined the effect of each of the data enhancement methods seperately and when used in combinations. The two most interesting combinations were R, as it was able to increase the weighted F1-score, and RC, as it produced the highest positive recall.

We then did experiments with different training optimizations to see if it was possible to increase the performance by altering the network. We tested various networks, dataset balances, training techniques and

optimizers, where we found out that RNN, a dataset balanced towards positive samples, Rezoom and SGD produced the best results for us. By combining this with R and RC from the previous experiment, we were able to improve the performance further, with R providing the best results seen in the thesis.

After experiments with a higher number of training iterations and an evaluation against a different dataset, we discussed interesting topics related to the experiments, such as time requirements and the concept of training iterations and confidences. And more importantly, we discussed how our system fits into real world scenarios and what the results from our system look like when using per frame, per sequence and per polyp detection and why these metrics are important. Finally, we compared our system against state-of-the-art systems, concluding that when using per sequence or by polyp detection, our system could match up to most or all of the entries.

5.2 Main Contributions

We have provided a deeper understanding of the potential in using neural networks for medical scenarios, especially for polyp detection. We have used polyp detection as a scenario to explore how data enhancement methods affect the training and evaluation of neural networks, and what effect each method have on performance. We have also explored how various training techniques, including different network models and optimizers, can be used to optimize performance of the overall system. Towards the end, we discussed interesting topics related to neural networks and polyp detection.

In order to achieve this, we have created a pipeline to apply data enhancements, and prepare the training and evaluation sets. Each set, consisting of either a combination of data enhancement methods or training settings, was then used to train a network, and later evaluated. We trained and evaluated approximately 120 sets, each requiring about 17 hours to complete for a total of around 2 000 hours, to be able to draw meaningful conclusions. These conclusions and results were the subjects in a comparison with other state-of-the-art systems and various discussions regarding how they could be seen in relation to real world scenarios, how confidences can be used to optimize for different scenarios, and how per frame, per sequence and per polyp detection can be visualized and their use in medical fields.

In section 1.2, we outlined three main questions that we are able to answer as follows:

1. *Does neural networks work for polyp detection?*
   Yes. Compared to state-of-the-art systems, neural networks produces good results, where depending on metrics and scenario, it produces comparable or better results.

2. *Can data enhancement methods improve the polyp detection rate?*
   Yes. Rotation increases the overall performance, and a combination of rotation and contrast enhancement results in the highest number of detected polyps. Additionally, both masking reflections and contrast enhancement show potential depending on the video. Brightness variations, on the other hand, seems unable to produce positive effects.

3. *Can the network architecture be modified to improve the polyp detection rate?*
   Yes. We found that using RNN as the network architecture, a dataset balance with a focus on positive samples, Rezoom as a training technique and SGD as a training optimizer, produce the best results where the detection is increased by up to 300%, while keeping the number of false positives relatively stable.

5.3 Future work

In this thesis, we have experimented with different data enhancement methods and optimizations in training. While we have performed many experiments, there are still some questions and experiments remaining.

In sections 4.8.5 and 4.8.6 we mentioned object tracking, which is the concept of tracking the polyp in preceding and subsequent frames of a video. Instead of treating every frame as a seperate entity, the video itself is treated as an entity where camera movement is tracked between frames. This could lead to a higher positive recall, because even if the polyp is not detected in every frame, its location can be tracked by using the video as context. The implementation should be fairly straight forward, as it could be implemented as an independent module working alongside the existing system.

In section 4.4.3.1 we mentioned that a more advanced implementation of masking reflections could improve the results in difficult images. Judging by the effect masking reflections have on some of the videos, we believe it to have great potential. Masking reflections is hard to do as a one size fits all solution, so an implementation taking each individual frame of the video into account in order to mask each type of reflection properly should have a noticable positive effect on detection.

While writing this thesis, new versions of TensorFlow and TensorBox have been released. Additionally, new checkpoints of Inception and Resnet have also been released. While our experiments with Inception and Resnet did not yield optimal results, these newer versions and checkpoints could potentially produce better results. It would be interesting to see how they perform with data enhancement and optimized training.

5.4 Final remarks

We have solved the tasks outlined in section 1.2 and achieved good results. However, in retrospect we realize that the training optimization experiment should have been performed before the data enhancement experiment. By doing it in that order, all data enhancements would have been tested with optimized training as opposed to the default settings, possibly leading to a more thorough examination.

Appendices

Appendix A

Source Code

The source code for the system developed during this thesis, as well as the data from the experiments, are located at `https://github.com/FredrikAndRuneMaster/MasterThesis`.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." In: *CoRR* abs/1603.04467 (2016). URL: http://arxiv.org/abs/1603.04467.

[2] O. Abdel-Hamid, A. r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu. "Convolutional Neural Networks for Speech Recognition." In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.10 (Oct. 2014), pp. 1533–1545. ISSN: 2329-9290. DOI: 10.1109/TASLP.2014.2339736.

[3] Luis A Alexandre, Joao Casteleiro, and Nuno Nobreinst. "Polyp detection in endoscopic video using SVMs." In: *Proc. of PKDD*. 2007, pp. 358–365.

[4] Telmo Amaral, Luís M. Silva, Luís A. Alexandre, Chetak Kandaswamy, Joaquim Marques de Sá, and Jorge M. Santos. "Transfer Learning Using Rotated Image Data to Improve Deep Neural Network Performance." In: *Image Analysis and Recognition: 11th International Conference, ICIAR 2014, Vilamoura, Portugal, October 22-24, 2014, Proceedings, Part I*. Ed. by Aurélio Campilho and Mohamed Kamel. Cham: Springer International Publishing, 2014, pp. 290–300. ISBN: 978-3-319-11758-4. DOI: 10.1007/978-3-319-11758-4_32. URL: http://dx.doi.org/10.1007/978-3-319-11758-4_32.

[5] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vilarino. "Texture-based polyp detection in colonoscopy." In: *Bildverarbeitung für die Medizin*. Springer, 2009, pp. 346–350.

[6] C. Bowley, A. Andes, S. Ellis-Felege, and T. Desell. "Detecting wildlife in uncontrolled outdoor video using convolutional neural networks." In: *2016 IEEE 12th International Conference on e-Science (e-*

Science). Oct. 2016, pp. 251–259. DOI: 10.1109/eScience.2016.7870906.

[7] T. Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. RFC Editor, Mar. 2014, pp. 1–16. URL: http://www.rfc-editor.org/rfc/rfc7159.txt.

[8] Da-Chuan Cheng, Wen-Chien Ting, Yung-Fu Chen, Qin Pu, and Xiaoyi Jiang. "Colorectal polyps detection using texture features and support vector machine." In: *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*. Springer, 2008, pp. 62–72.

[9] Jianpeng Cheng, Li Dong, and Mirella Lapata. "Long Short-Term Memory-Networks for Machine Reading." In: *CoRR* abs/1601.06733 (2016). URL: http://arxiv.org/abs/1601.06733.

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

[11] Kunio Doi. *Computer-Aided Diagnosis in Medical Imaging: Historical Review, Current Status and Future Potential*. Kurt Rossmann Laboratories, For Radiologic Image Research, Department of Radiology, The University of Chicago, 2007.

[12] ECMA. *ECMA-404: The JSON Data Interchange Format*. Oct. 2013. URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

[13] R. Fainii, M. Alamaniotis, and L. H. Tsoukalas. "Distribution congestion prediction using artificial neural networks for big data." In: *Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MedPower 2016)*. Nov. 2016, pp. 1–7. DOI: 10.1049/cp.2016.1108.

[14] Gondal G., Grotmol T., Hofstad B., Bretthauer M., Eide T. J., and Hoff G. "The Norwegian Colorectal Cancer Prevention (NORCCAP) Screening Study." In: *Scandinavian Journal of Gastroenterology* 38.6 (2003), pp. 635–642. DOI: 10.1080/00365520310003002.

[15] Google. *TensorFlow*. 2017. URL: https://www.tensorflow.org/ (visited on 03/25/2017).

[16] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. "Speech Recognition with Deep Recurrent Neural Networks." In: *CoRR* abs/1303.5778 (2013). URL: http://arxiv.org/abs/1303.5778.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In: *CoRR* abs/1512.03385 (2015). URL: http://arxiv.org/abs/1512.03385.

[18] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups." In: *IEEE Signal Processing Magazine* 29.6 (Nov. 2012), pp. 82–97. ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2205597.

[19] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[20] Sae Hwang, JungHwan Oh, W. Tavanapong, J. Wong, and P.C. de Groen. "Polyp Detection in Colonoscopy Video using Elliptical Shape Feature." In: *Proc. of ICIP*. Sept. 2007, pp. 465–468.

[21] I. Iseri. "An Artifical intelligence based software application for microcalcification detection on mammogram images." In: *2016 24th Signal Processing and Communication Application Conference (SIU)*. May 2016, pp. 1973–1976. DOI: 10.1109/SIU.2016.7496154.

[22] H. J. Jeong, M. J. Lee, and Y. G. Ha. "Integrated Learning System for Object Recognition from Images Based on Convolutional Neural Network." In: *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. Dec. 2016, pp. 824–828. DOI: 10.1109/CSCI.2016.0160.

[23] M. I. Jordan and T. M. Mitchell. "Machine learning: Trends, perspectives, and prospects." In: *Science* 349.6245 (2015), pp. 255–260. ISSN: 0036-8075. DOI: 10.1126/science.aaa8415. eprint: http://science.sciencemag.org/content/349/6245/255.full.pdf. URL: http://science.sciencemag.org/content/349/6245/255.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[25] J. Larsen and C. Goutte. "On optimal data split for generalization estimation and model selection." In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*. Aug. 1999, pp. 225–234. DOI: 10.1109/NNSP.1999.788141.

[26] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539.

[27] B. Li, Y. Fan, M. Q. H. Meng, and L. Qi. "Intestinal polyp recognition in capsule endoscopy images using color and shape features." In: *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Dec. 2009, pp. 1490–1494. DOI: 10.1109/ROBIO.2009.5420969.

[28] B. Li, M. Q. H. Meng, and C. Hu. "Motion analysis for capsule endoscopy video segmentation." In: *2011 IEEE International Conference on Automation and Logistics (ICAL)*. Aug. 2011, pp. 46–51. DOI: 10.1109/ICAL.2011.6024682.

[29] Baopu Li and M.Q.-H. Meng. "Tumor Recognition in Wireless Capsule Endoscopy Images Using Textural Features and SVM-Based Feature Selection." In: *ITBM, IEEE* (2012).

[30] Min Lin, Qiang Chen, and Shuicheng Yan. "Network In Network." In: *CoRR* abs/1312.4400 (2013). URL: http://arxiv.org/abs/1312.4400.

[31] A. V. Mamonov, I. N. Figueiredo, P. N. Figueiredo, and Y. H. Richard Tsai. "Automated Polyp Detection in Colon Capsule Endoscopy." In: *IEEE Transactions on Medical Imaging* 33.7 (July 2014), pp. 1488–1502. ISSN: 0278-0062. DOI: 10.1109/TMI.2014.2314959.

[32] Microsoft. *Advanced Systems Format (ASF) Specification*. 2004. URL: http://go.microsoft.com/fwlink/p/?linkid=31334.

[33] H. T. Nguyen, C. T. Nguyen, P. T. Bao, and M. Nakagawa. "Preparation of an Unconstrained Vietnamese Online Handwriting Database and Recognition Experiments by Recurrent Neural Networks." In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Oct. 2016, pp. 144–149. DOI: 10.1109/ICFHR.2016.0038.

[34] NHS. *Gastroscopy - How it's performed*. 2015. URL: http://www.nhs.uk/Conditions/gastroscopy/Pages/How-it-is-performed.aspx.

[35] G. Orchard, X. Lagorce, C. Posch, S. B. Furber, R. Benosman, and F. Galluppi. "Real-time event-driven spiking neural network object recognition on the SpiNNaker platform." In: *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*. May 2015, pp. 2413–2416. DOI: 10.1109/ISCAS.2015.7169171.

[36] C. Phanikrishna and A. V. N. Reddy. "Contour tracking based knowledge extraction and object recognition using deep learning neural networks." In: *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. Oct. 2016, pp. 352–354. DOI: 10.1109/NGCT.2016.7877440.

[37] Michael PIGNONE and Harold C SOX. "Screening for colorectal cancer: U.S. Preventive Services Task Force recommendation statement." In: *Annals of Internal Medicine* 149.9 (2008), pp. 627–637.

[38] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart Ter Haar Romeny, and John B. Zimmerman. "Adaptive Histogram Equalization and Its Variations." In: *Comput. Vision Graph. Image Process.* 39.3 (Sept. 1987), pp. 355–368. ISSN: 0734-189X. DOI: 10.1016/S0734-189X(87)80186-X. URL: http://dx.doi.org/10.1016/S0734-189X(87)80186-X.

[39] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Sigrun Losada Eskeland, Thomas de Lange, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Peter Thelin Schmidt, Michael Riegler, and Pål Halvorsen. "Kvasir: A Multi-Class Image Dataset for Computer Aided Gastrointestinal Disease Detection." In: *Proc. of MMSYS*. 2017.

[40] David Martin Ward Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." In: *International Journal of Machine Learning Technology* 2.1 (2011), pp. 37–63.

[41] International Agency for Research on Cancer. *World Cancer Report 2014 (International Agency for Research on Cancer)*. World Health Organization, 2014. Chap. The Global and Regional Burden of Cancer.

[42] M. Riegler, K. Pogorelov, S. L. Eskeland, P. T. Schmidt, Z. Albisser, D. Johansen, C. Griwodz, P. l. Halvorsen, and T. de Lange. "From annotation to computer aided diagnosis: Detailed evaluation of a medical multimedia system." In: *ACM Trans. Multimedia Comput. Commun. Appl.* 9.4 (2016), pp. 1–23.

[43] Michael Riegler. "EIR - A Medical Multimedia System for Efficient Computer Aided Diagnosis." PhD thesis. University of Oslo, 2017.

[44] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Thomas de Lange, Carsten Griwodz, Peter Thelin Schmidt, Sigrun L. Eskeland, and Dag Johansen. "EIR - Efficient Computer Aided Diagnosis Framework for Gastrointestinal Endoscopies." In: *Proc. of CBMI*. 2016.

[45] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Thomas de Lange, Carsten Griwodz, Peter Thelin Schmidt, Sigrun L. Eskeland, and Dag Johansen. "EIR - Efficient Computer Aided Diagnosis Framework for Gastrointestinal Endoscopies." In: *Proc. of CBMI*. 2016.

[46] Michael Riegler, Konstantin Pogorelov, Jonas Markussen, Mathias Lux, Håkon Kvale Stensland, Thomas de Lange, Carsten Griwodz, Pål Halvorsen, Dag Johansen, Peter T Schmidt, and Sigrun L. Eskeland. "Computer Aided Disease Detection System for Gastrointestinal Examinations." In: *Proc. of MMSys*. 2016.

[47] Elisabeth Rosenthal. "The $2.7 Trillion Medical Bill." In: *New York Times* (June 1, 2013).

[48] Hasim Sak, Andrew W. Senior, and Françoise Beaufays. "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition." In: *CoRR* abs/1402.1128 (2014). URL: http://arxiv.org/abs/1402.1128.

[49] Jürgen Schmidhuber. "Deep Learning in Neural Networks: An Overview." In: *CoRR* abs/1404.7828 (2014). URL: http://arxiv.org/abs/1404.7828.

[50] American Cancer Society. *What is endoscopy?* 2015. URL: http://www.cancer.org/treatment/understandingyourdiagnosis/examsandtestdescriptions/endoscopy/endoscopy-what-is-endoscopy.

[51] Amnon Sonnenberg and Robert M. Genta. "Low prevalence of colon polyps in chronic inflammatory conditions of the colon." In: *American Journal of Gastroenterology* 110.7 (July 2015), pp. 1056–1061. ISSN: 0002-9270. DOI: 10.1038/ajg.2015.130.

[52] S. J. Stryker, Wolff B. G., Culp C. E., Libbe S. D., Ilstrup D. M., and MacCarty R. L. "Natural history of untreated colonic polyps." In: *Gastroenterology* 93.5 (Nov. 1987), pp. 1009–1013.

[53] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning." In: *CoRR* abs/1602.07261 (2016). URL: http://arxiv.org/abs/1602.07261.

[54] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going Deeper with Convolutions." In: *CoRR* abs/1409.4842 (2014). URL: http://arxiv.org/abs/1409.4842.

[55] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. "Rethinking the Inception Architecture for Computer Vision." In: *CoRR* abs/1512.00567 (2015). URL: http://arxiv.org/abs/1512.00567.

[56] N. Tajbakhsh, S. R. Gurudu, and J. Liang. "Automated Polyp Detection in Colonoscopy Videos Using Shape and Context Information." In: *IEEE Transactions on Medical Imaging* 35.2 (Feb. 2016), pp. 630–644. ISSN: 0278-0062. DOI: 10.1109/TMI.2015.2487997.

[57] F. Toqué, E. Côme, M. K. El Mahrsi, and L. Oukhellou. "Forecasting dynamic public transport Origin-Destination matrices with long-Short term Memory recurrent neural networks." In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Nov. 2016, pp. 1071–1076. DOI: 10.1109/ITSC.2016.7795689.

[58] Linda Villarosa. *Done Right, Colonoscopy Takes Time, Study Finds*. Dec. 2006. URL: http://www.nytimes.com/2006/12/19/health/19colo.html.

[59] P. Voigtlaender, P. Doetsch, and H. Ney. "Handwriting Recognition with Large Multidimensional Long Short-Term Memory Recurrent Neural Networks." In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Oct. 2016, pp. 228–233. DOI: 10.1109/ICFHR.2016.0052.

[60] Yi Wang, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C de Groen. "Polyp-Alert: Near Real-time Feedback during Colonoscopy." In: *Computer methods and programs in biomedicine* 120.3 (2015), pp. 164–179.

[61] Yi Wang, Wallapak Tavanapong, Johnson Wong, JungHwan Oh, and Piet C de Groen. "Part-Based Multiderivative Edge Cross-Sectional Profiles for Polyp Detection in Colonoscopy." In: *Journal of BMHI* 18.4 (2014), pp. 1379–1389.

[62] G. Yadav, S. Maheshwari, and A. Agarwal. "Contrast limited adaptive histogram equalization based enhancement for real time video system." In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. Sept. 2014, pp. 2392–2397. DOI: 10.1109/ICACCI.2014.6968381.

[63] H. M. Yang, X. Y. Zhang, F. Yin, Z. Luo, and C. L. Liu. "Unsupervised Adaptation of Neural Networks for Chinese Handwriting Recognition." In: *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*. Oct. 2016, pp. 512–517. DOI: 10.1109/ICFHR.2016.0100.

[64] J. Zhang and C. Zong. "Deep Neural Networks in Machine Translation: An Overview." In: *IEEE Intelligent Systems* 30.5 (Sept. 2015), pp. 16–25. ISSN: 1541-1672. DOI: 10.1109/MIS.2015.69.

[65] An Zheng and Mingyang Wang. "Convolutional Neural Networks-based Plankton Image Classification System." In: 2015.

[66] M. Zhou, G. Bao, Y. Geng, B. Alkandari, and X. Li. "Polyp detection and radius measurement in small intestine using video capsule endoscopy." In: *2014 7th International Conference on Biomedical Engineering and Informatics*. Oct. 2014, pp. 237–241. DOI: 10.1109/BMEI.2014.7002777.

[67] M. Zhou, G. Bao, Y. Geng, B. Alkandari, and X. Li. "Polyp detection and radius measurement in small intestine using video capsule endoscopy." In: *2014 7th International Conference on Biomedical Engineering and Informatics*. Oct. 2014, pp. 237–241. DOI: 10.1109/BMEI.2014.7002777.

[68] Karel Zuiderveld. "Graphics Gems IV." In: ed. by Paul S. Heckbert. San Diego, CA, USA: Academic Press Professional, Inc., 1994. Chap. Contrast Limited Adaptive Histogram Equalization, pp. 474–485. ISBN: 0-12-336155-9. URL: http://dl.acm.org/citation.cfm?id=180895.180940.