A Medical Multimedia Real-Time Polyp Detection System using Low Computational Resources

Pedram Sherafat
Khan Asif Qayyum Tanoli

Thesis submitted for the degree of
Master in Informatics: Programmering and
Software Engineering, Nanoelectronics and
Robotics (Signal Processing)
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2017

A Medical Multimedia Real-Time Polyp Detection System using Low Computational Resources

Pedram Sherafat
Khan Asif Qayyum Tanoli

A Medical Multimedia Real-Time Polyp Detection System using Low Computational Resources

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Acknowledgment

We would like to express our gratitude to our supervisor Pål Halvorsen and assistant supervisors Michael Riegeler and Konstantin Pogorelov. Without their understanding and patience, this thesis would never have been completed.

Furthermore, we would like to thank Simula research lab for giving us the opportunity to work in a great environment, filled with joy and wisdom. We will also like to appreciate Tonny Bekoe, Farrukh Wahab, Salman Asskali for visiting us continuously in the lab and keeping us with company during nights and weekends.

All praise is due to Allah alone, the Sustainer of all the worlds. I, Asif Khan, will express gratitude to my late father Abdul Qayyum Khan, who motivated me to pursue my master degree and would have been proud today. I truly appreciate my mother for helping me reaching my goals, my siblings and their families for all their encouragement and not letting me die because of hunger. Furthermore, I would like to say thank you to my friends Abdullah Celik, Ali Zen, Fazal-ul Rehman, Sawera Tariq, Hersa Khoirunisa and Mustafa Stanikzai for all their support, motivation, and most importantly joyful time during our time at university. Finally, A praise to Pedram Sherafat for cyka rushing B with me, not only during this project but also during 5 years at the university and we really enjoyed ourselves.

I, Pedram Sherafat, would like to sincerely express my deepest gratitude to my dear parents and sister, Shifteh Sherafat. Going through the roller coaster ride which is life, I have learned many valuable lessons and thank you all for the love, motivation, encouragement and support which have led me further and further towards a more joyful, educational life and opened up many opportunities for me. To the best sister anyone could ask for; Thanks for supporting me through, not only this research, but through life in general. I am lucky to have such a clever sister which always looks out for me and gives on-point advice with honesty. To my, geographical, distant relatives; Thank you all for your support and love. I miss you and hope to see you soon.

I would like to express my gratitude to the Tørudbakken twins, Øystein and Vebjørn and to their family for being such good friends and contribute to memorable journeys. I would also like to thank my friends from the newer and old days, which some, Asif also listed. To Kjell Kael Arne Eldorhagen Diaz and Marius E. Gomes for their joyful experiences throughout many years, supporting me and understanding the grind, to the guys at the university that rowed in the same boat, to Mats Ødegaard Jensen and to those that have kept my spirit fueled for greatness to this day. Lastly, to Asif Khan, which have always dropped me an AWP, smoked and rushed mid with me on the first round of the CT. We have had great runs throughout the years at UIO and the time off and become good friends. Thanks for being such a considerate friend and even though our run is over for now, we still have much to check of our to-do list.

Education is the passport
to the future,
for tomorrow belongs to
those
who prepare for it
today

- Malcolm X.

Abstract

Within the colorectal-tract the developed polyps are common originators to the colon cancer. With the early detection of these polyps, colorectal cancer will in most cases be avoided. As of now, colorectal examinations is both costly and time consuming which is a hurdle against mass polyp screenings. As an alternative examination process, researchers has done a lot of work on recent technologies, such as Computer-aided diagnosis (CAD) and Automated Computer Diagnosis (ACD) could in the near future become a more reliable option.

In this research, the focal point has been on real-time polyp detection on computers with low computational resources with the help of open source libraries such as LIRE Lucene, OpenCV. To achieve reliable detection scores, we have implemented our system with different algorithms alongside LIRE's global image features. For further improvements, we have experimented with sophisticated deep learning approaches. Our system includes the complete pipeline from extraction of the feature values with the selected features and indexing the training-set, to the classification of frames the enabled camera captures and outputs the appropriate class to the GUI which the endoscopist is reviewing. Separately, this research has also experimented with unorthodox features to improve classification and evaluate if there are any benefits from doing so. Therefore, experimentation has been done with Google Tensorflow Inception v3 model and used the 1008 probabilities for classification as feature values to describe our data-set. These feature values in combination with eight different machine learning algorithms show propitious results. With further research, these methods have the potential to achieve better classification than JCD, CEDD and Tamura. The system combines machine learning, image recognition and extraction of global image features, and it is built in a modular way, so that it can easily be extended to other disease and further developed.

Real-Time polyp detection using global features on a Intel NUC 5i3RYH achieve reasonable high classification - and FPS scores, compared to systems used under previous - and related work in this field. For the future, these results are encouraging and might be possible to be made compatible for mobile devices and accomplish scalable polyp detection using mobile smart phones that process data received from the wireless capsule. With proceeded work, the combination of implementation on low scale computers and taking advantages of evolving hardware in mobile devices, examination cost will drastically be reduced. This would decrease mortality rates as well as reduce the burden that is placed on the health care system by the endoscopy examination procedure.

Keywords

Image Classification, Performance, Polyp Detection, Real-time, LIRE, OpenCV, GI tract, Medical Multimedia, Pill-cam, Deep Learning

Table of Contents

Abstract **3**

Table of Contents **8**

List of Tables **10**

List of Figures **12**

Abbreviations **13**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Problem Statement 3

1.3 Limitation and Scope 4

1.4 Research and Method 5

1.5 Main Contribution 6

1.6 Outline 7

2 Background and Related Work **9**

2.1 Medical Scenario 9

2.1.1 Diseases in the GI tract 9

2.1.2 GI Examination Procedures and Methods of Screening 14

2.2 Related work 21

2.3 Classification 23

2.4 EIR - Efficient computer aided diagnosis 23

2.5 Summary 25

3 Real-Time Polyp Detection 27

3.1 Global- and Local Features 28

3.1.1 Feature selection 29

3.2 Feature Combination 31

3.2.1 Early Fusion 32

3.2.2 Late Fusion 32

3.3 Machine Learning Overview 33

3.3.1 K Nearest Neighbour 34

3.3.2 Artificial Neural Networks 35

3.3.3 Random Forests 36

3.3.4 Decision Tree 37

3.3.5 Ada Boost 37

3.3.6 Naive Bayes 37

3.3.7 Support vector machines -SVM 38

3.4 Deep Learning 39

3.5 The LIRE platform 39

3.5.1 Indexing 40

3.5.2 Search 41

3.6 Model Creation 42

3.7 Summary 45

4 Iterative Development, Experiments and Results 47

4.1 Medical data 47

4.1.1 Data-set 49

4.2 Machine Setup 49

4.3 Evaluation Method and Metrics 51

4.4 System Development Iterations 53
4.5 Iteration I: Real-Time Color Detection 53
4.5.1 Model Creation 54
4.5.2 Architecture and Pipeline 55
4.5.3 System Experiments and Evaluation 56
4.6 Iteration II: Prototype of Real-Time Object Detection 56
4.6.1 Model Creation 57
4.6.2 Architecture and Pipeline 61
4.6.3 System Experiments and Evaluation 62
4.6.4 Iteration II: Summary 68
4.7 Iteration III: Real-Time Polyp Detection 68
4.7.1 Model Creation 68
4.7.2 Architecture and Pipeline 69
4.7.3 Algorithm 70
4.7.4 System Experiments and Evaluation 72
4.7.5 Iteration III: Summary 79
4.8 Classification improvements 80
4.8.1 Model Creation 81
4.8.2 Classifier Structure 81
4.8.3 Classifier Experiments and Evaluation 82
4.8.4 Rotation 86
4.8.5 Iteration IV: Summary 86
4.9 Summary of chapter 4 88

5 Analysis and Conclusion **91**
5.1 Summary 91
5.2 Main contribution 93
5.3 Problem Statement 94
5.4 Future Work 96

Bibliography **96**

List of Tables

2.1 The comparison of the state-of-the-art systems for polyp detection. 22

3.1 Leave-one-out cross validation with different features 30

3.2 Inception v3 data structure 39

4.1 Data-set T1 folder in details 48

4.2 Data-set T2 folder in details 49

4.3 List of HW and SW utilized during this research 51

4.4 Measured processing time table 56

4.5 Red Folder data-set 58

4.6 Algorithm calculation for classification of a single frame 71

4.7 Image function summation 71

4.8 ASU Mayo Clinic Polyp data-set 86

5.1 JCD classification characteristics with ski-learn 110

5.2 JCD classification characteristics with 180 degrees rotated copied data . . 111

5.3 Early fusion of JCD and Tamura classification characteristics 112

5.4 Early fusion of JCD and Tamura classification characteristics 180 degrees rotated copied dat 113

5.5 Machine learning classification characteristics with inception v3 1008 features 114

5.6 Machine learning classification characteristics with inception v3 1008 features, this time 180 degrees rotated copied data 115

5.7 Early fusion of JCD and Tamura classification characteristics, rotated 90, 180 and 270 degrees of the copied data. 116

List of Figures

2.1 Gastrointestinal (GI) tract 10

2.2 The Mucosa 11

2.3 Polyp Types 12

2.4 Survival rates 14

2.5 Screening methods 16

2.6 Organ Radiation 17

2.7 Wireless-Capsular 19

2.8 EIR System 25

3.1 A prototype of the Job Information dialog 32

3.2 Late fusion sketch 33

3.3 Example of k-NN classification 35

3.4 Neural Network 36

3.5 SVM model 38

3.6 Model for indexing images in training-set 40

3.7 Pipeline for the image ranking 42

3.8 Main-Model of classification and our real-time detection system 43

3.9 Real-time Classification GUI 44

4.1 Leave one-out-cross-validation splits 50

4.2 Instance comparison of color detection 54
4.3 Frame capture and processing time 57
4.4 Separating data points 60
4.5 Red-folder architecture and pipeline 62
4.6 Graph shows how image resolution affects processing and increases FPS . 64
4.7 Red-folder weighted score 65
4.8 Algorithm avg time 67
4.9 FPS with different features 73
4.10 F1 score for iteration III 74
4.11 Precision for iteration III 75
4.12 Recall for iteration III 76
4.13 Frame and algorithm time consumption 78
4.14 Total processing time per frame 79
4.15 Machine learning classification characteristics with JCD 83
4.16 Machine learning classification characteristics with early fusion of JCD and Tamura 84
4.17 Machine learning classification characteristics with inception v3 85

5.1 Diagram showing the percentage with 180 degrees rotated data and JCD. . 108
5.2 Diagram showing the percentage with 180 degrees rotated data and JCD in early fusion with Tamura 109
5.3 Diagram showing the percentage with 180 degrees rotated data and Inception v3 117

Abbreviations

CRC	=	Colorectal Cancer
GI	=	Gastrointestinal
FPS	=	Frames Per Second
CCD	=	Carged Couple Device
LCD	=	Liquid Crystal Display
VCE	=	Video Capsular Endoscopy
WCE	=	Wireless Capsular Endoscopy
SVM	=	Support Vector Machine
GPU	=	Graphical Processing Unit
RGB	=	Red Green Blue
CEDD	=	Color and Edge Directivity Descriptor
JCD	=	Joint Composit Descriptor
LSH	=	Locality Sensitive Hashing
USPSTF	=	United States Preventive Services Task Force
LIRE	=	Lucene Image Retrival
ACM	=	Associattion for Computing Machinery
CT	=	Computer Tomography
ACD	=	Automated Computer Diagnosis
CAD	=	Computer Aided Diagnosis
ACD	=	Automated Computer Diagnosis
ANN	=	Artificial Neural Networks
kNN	=	K Nearest Neighbour
HSV	=	Hue, Saturation, Value

Chapter 1

Introduction

1.1 Background and Motivation

Potentially, the human digestive system has a high risk of being affected by several types of diseases in certain age groups, although diet, family health-history, and other factor that will be discussed later on, also sets its mark in statistics. The human digestive system is prone to different types of diseases ranging from very serious and dangerous to less dangerous. One example of the former is gastric and colorectal cancer (CRC). CRC is the most common type of cancer and if detected in the later stages of evolution, it is lethal and not curable. According to Colon Cancer Survival Rates, it is stated that 5 years survival rate is 93% in stage one, and 8% in stage four [39]. Early detection is therefore crucial to prevent the unfortunate outcome. Following the research from the International Agency for Research on Cancer [38], it is stated that recent occurring cases have a probability of more than 65 % of being in countries with high rate development standards.

These numbers seem to suggest that our newly adopted lifestyle has affected our health negatively. The suspicion is further stressed by the fact that there is an increasing tendency of mortality rate by CRC in countries that becomes more developed. Keeping in mind that these are countries located in central Europe and have access to high nutrition standards and established health-care system.

Within the colorectal tract the developed polyps are common originators to the much unwanted colon cancer. With the early detection of these polyps, CRC will in most cases be avoided [57]. Although, if a polyp is not removed before further evolution, the risk of having a polyp diagnosed cancerous is 2.5% after 5 years, and 8% after 10 years. After a time of 20 years, the risk increases to a noteworthy and alarming 24%. Therefore target at the detection of colorectal polyps, which are the forerunner to CRC. Awareness of the fact that early polyp detection prevents CRC has in recent years lead to an increase in screening rates using colonoscopy [6]. Around 20% of colon cancer are derived from adenomatous polyps, which contains dysplastic cells that might further evolve into being cancerous. CRC can be avoided up to 5 years after treatment, although this relies heavily on the endoscopist's ability to detect polyps and remove them. Even so, according to the U.S Preventive Services Task Force (USPSTF), the standard for screening intervals of people between 50-75 years is every 10 years [16], which is similar to the Norwegian standard [18]. The overall procedure of polyp removal is also considered to be an uncomfortable and intrusive procedure for the patient, and thus, some people might not want to get examined at all.

Table 2.1 presents different systems using different methods, algorithms and partial systems as an approach to detecting polyps. These methods have been suggested to show encouraging results after being tested in a closed environment. While it still looks as if these numbers are acceptable, there are still some uncertainties on how well these approaches would perform in a real work environment, such as a hospital.

The general examination cost is also one of the main downsides. In Norway, the cost per examination is approximately 450$ and reflects how resource demanding the examination is. Scaling these procedures for the entire population also requires extensive budget. The way the colonoscopy is performed today in terms of the examination and the following analysis, is time inefficient, intrusive and uncomfortable for the patient. On a larger scale, a lot of hospital resources are used. The main focus of this project is to use previous standards and machine learning to design and implement a real-time polyp detection system that can be deployed using low computational resource computers. Overall, we believe this will help increase screening rates, reduce the cost and toll screenings place on the health care system, encourage further research and reduce CRC mortality rate.

1.2 Problem Statement

As explained introduction-wise in the section above, there are many factors to take into account, but there is a great potential for improvements in an automated polyp detection system. The purpose of this research is to design and implement an automated real-time detection system that meets the requirement to capture polyps during an examination procedure and make the endoscopist aware of them. Within this domain, very little research has been done, and in this research, we will present a system that achieves state of the art performance. This system is based on the idea of EIR [49]. EIR is named after the goddess of healing in Norwegian mythology and is an interdisciplinary research of a multimedia system that can be used as a tool in the detection of polyps in the GI tract. The challenges we encounter are:

- ***Is it possible to detect polyps live on computers with low computational resources? How will they perform?***

  The purpose of this research is to design and implement our new and innovative way of thinking by using the existing EIR system, to build a real-time detection system on a computer with low computational resources, that can contribute to answer unsolved research question. This will further have a noteworthy impact on helping people survive lethal diseases, improve their quality of life and at the same time reduce a total cost of the equipment. This is an important step since modern mobile devices are getting higher computational power and in future could also be used for polyp detection as well as other diseases.

- ***Will features and image resolution in combination with different algorithms improve performance with regard to FPS and accuracy?***

  Image quality and size requires computational processing power, specially considering search-based classification. It would be interesting to see if feature selection executed with specific algorithms would affect overall performance.

- ***How will unorthodox features perform in terms of classification compared the global image features of LIRE?***

  Recently, there has been much talk about artificial intelligence in the media. We are excited to experiment to discover what kind of performance we are able to achieve on a small form factor computer with low computational resources.

1.3 Limitation and Scope

Design and implementation of a real-time polyp detection system has several factors regarding limitation. There is a huge challenge classification-wise, which means there is still much work to be done in determining which features provides best detection results and how these features are combined to maximize accuracy. In our case, we look into which features are available, is low resource demanding and suited for real-time use and implement fast algorithms for classification that does not affect FPS to harshly. In this research, we will be using the ASU-Mayo data-set for our research purposes and is split into two main categories, test and train. These splits will also be used in our cross validation step. The features are extracted from the data-set we have available, which means that the achievable performance is also limited and dependant on our data-set and split. We also found a few misplaced images in given data-set, which will also affect the results.

There are several types of diseases of the GI tract, however our main focus is on polyps and how well the system detects these with regards do performance in real time.

This is particularly challenging since there are many different diseases that can occur in the colon, and it can also be leftovers that polyps can hide behind. There are also other challenges in term of lack of publicly available large scale data-sets for medical research purpose of the GI tract diseases. This leads to evaluation difficulties.

Hospitals use different equipment to collect data and it can be of different resolution, lighting conditions and other attributes. Our research partners in ASU Mayo, Vestre Viken Hospital Trust, Rikshospitalet and the Karolinska University of Hospital operate with different endoscopes for obtaining and collection their medical data. This leads to different quality of the images in term of attributes such as resolution and light conditions. This is a challenge which has to be resolved by developing a system which can cope with these constraints and differences. Although we strive for polyp detection with our system, we want to create a modular system which later can be generalized in the future to include the detection of other diseases.

We still need to emphasis on that our system should be generic, that means it should be easily extendable for detecting different diseases if required data-set is available in the future. This system should also be verified with regards to different use cases in order to prove that the system actually is generic and is efficient in terms of processing time with sizable amount of data.

1.4 Research and Method

This research has followed the design paradigm suggested and described by Association for Computing Machinery (ACM) Task Force on the Core of Computer Science [15]. It encourages new ways of thinking about computational work as a discipline. This has been done by exhibiting the disciplines content in a way that emphasizes the fundamental concepts, principles, and distinctions.

When it comes to the design, implementation and prototyping of the real-time polyp detection system, we have evaluated a system that still has room for improvement. To achieve desired outcome, we began by first study the focused literature of machine learning, image processing and deep learning. This for the sake of acquiring the knowledge and understanding of:

- How a machine experiences media, for instance audio or image. In our case we use videos, which is just a series of images that we extract and which makes up the data-set.
- How image processing empowers us to highlight the regions of interest and which features that can accurately help us with that.
- How the architectural layout of deep learning is set up and how it can be used to

our advantage.

These tools which are further improvements to EIR are evaluated with physicians by discussing the prototypes and the development of the system for them as users. Furthermore, we have designed and implemented software for training a classifier by using machine learning algorithms. Both the system and the classifiers have been coded from the ground up. The acquired data-set has been prepared with implemented software for feature extraction and their combination. This was necessary in order to evaluate the results and see how we could further improve the system detection phase.

We have experimented with different methods and changed or varied several parameters for object detection and made a visualization of the resulting output. Eventually, evaluation of the methods is done with well-known machine learning metrics and measurements for the training-time, running the respective classifiers, FPS, and the overall performance of the real-time detection system.

1.5 Main Contribution

This research touches upon many aspects of design, implementation and system development of a real-time object detection system and if such a system can operate on a small form-factor computer with low computational power. Our medical scenario involves the detection of polyps in the colon, which if not detected early and removed, will grow cancerous and deadly. In this research we evaluate different approaches to detect polyps real-time and further experimentation with promising alternatives to improve classification results. This also include the evaluation of global features, different algorithms for classification and which of these give better result in terms of classification and FPS. This research also experiment with unorthodox features, such as using Google Tensorflow to classify images based on the inception v3 model and experiment with these categories as feature to see if they are to any benefit for our medical scenario. Due to low computational resources, we have implemented an idea for a specific pipeline that compensates for low FPS during image processing in order to reach real-time classification while the patient is undergoing endoscopy and is being examined.

1.6 Outline

Our thesis is structured as follows:

Chapter 2: Background and Related Work

Chapter 2 contains a brief description of the medical scenario, currently used GI examination methods and related work. This chapter also consist a short EIR definition.

Chapter 3: Real-Time Polyp Detection

We in this chapter have given a summary of LIRE and it's global features, such as JCD, Tamura and CEDD. We further described our pipeline for feature extraction and the search through these features which is the backbone of our real time polyp detection. In addition, this chapter also describes popular phenomenons of machine learning and gives an educative explanation of different approaches implemented in our iterative system development.

Chapter 4: Iterative Development, Experiments and Results

This chapter contains the documentation of our development journey of the real-time system. We present the data-set used and also each build of our system as sectioned iterations for a more structured system description. We evaluate our experiments in each iteration and discuss the results shown in figures and tables and what they mean for the tested build as well as the next iteration. Each iteration will be concluded with a summary of the system build.

Chapter 5: Analysis and Conclusion

The final chapter of our thesis will consist of analysis of findings, our contributions and conclusion. We give a complete summary of the entire thesis, also revisit the problem statement and summarize over those, in order to give the reader a full overview over the build-process, experiments, evaluation and what remains as future work.

Chapter 2

Background and Related Work

2.1 Medical Scenario

The colon or also called the large intestine is the last part of the GI tract and of the digestive system. It can be divided into four main parts which are also visualized in figure 2.1:

- The ascending colon travels up the right side of the abdomen.
- The descending colon travels down the left abdomen.
- The traverse colon runs across the abdomen.
- The sigmoid colon is a short curving of the colon and is located just before the rectum.

The job of the colon is to remove water, salt and some nutrients forming stool. Muscles line the Colon's walls, squeezing its contents along.

2.1.1 Diseases in the GI tract

There are certain diseases which might occur in GI tract and affect the colon's ability to work properly. Below are the list of few of them:

- Ulcerative colitis

- Diverticulitis
- Irritable bowel syndrome
- Colonrectal Cancer

Treatment of these diseases differs in term of cost and time, and is mostly based on it's severity and cure.

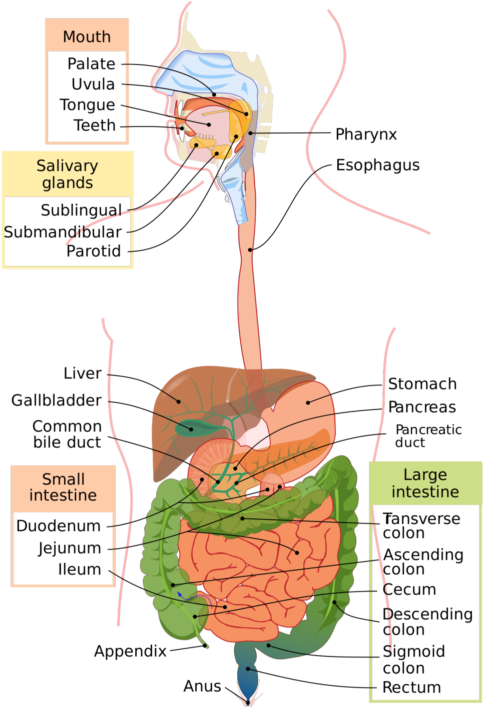


Figure 2.1: Gastrointestinal (GI) tract of a human. The GI tract (digestive tract, GI tract, GIT, gut, or alimentary canal) is an organ system which takes in food, digests it to extract and absorb energy and nutrients, and expels the remaining waste as feces and urine.[1]

Polyp in Colon Tract

In this research, we focus on polyp detection. A polyp is a small clump of cells that occurs on the wall of the colon as shown in figure 2.3b and is usually found in the GI-

[1] https://upload.wikimedia.org/wikipedia/commons/thumb/c/c5/Digestive_system_diagram_en.svg/2000px-Digestive_system_diagram_en.svg.png

tract, stomach, urine bladder and nose [56]. The wall of the colon is made up of several layers and a polyp can grow on the innermost layer also called Mucosa, as illustrated in figure 2.2. Most polyps are harmless but over the time it can develop into cancer that could be fatal if detected in later stages. Polyps often stick out of the tissue-wall as a small hill like structure. Polyps can develop for any age-group, however people older than 50 are at greater risk to get colon polyps [56]. The risk increases if the person smokes, overweight or if it is genetically in terms of other family members who have history of colon polyp.

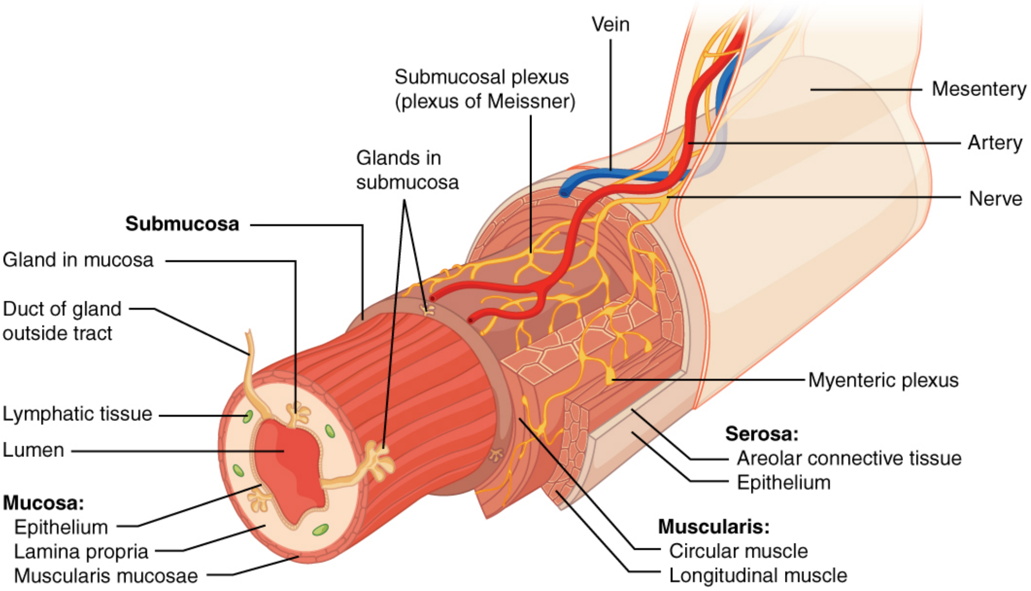


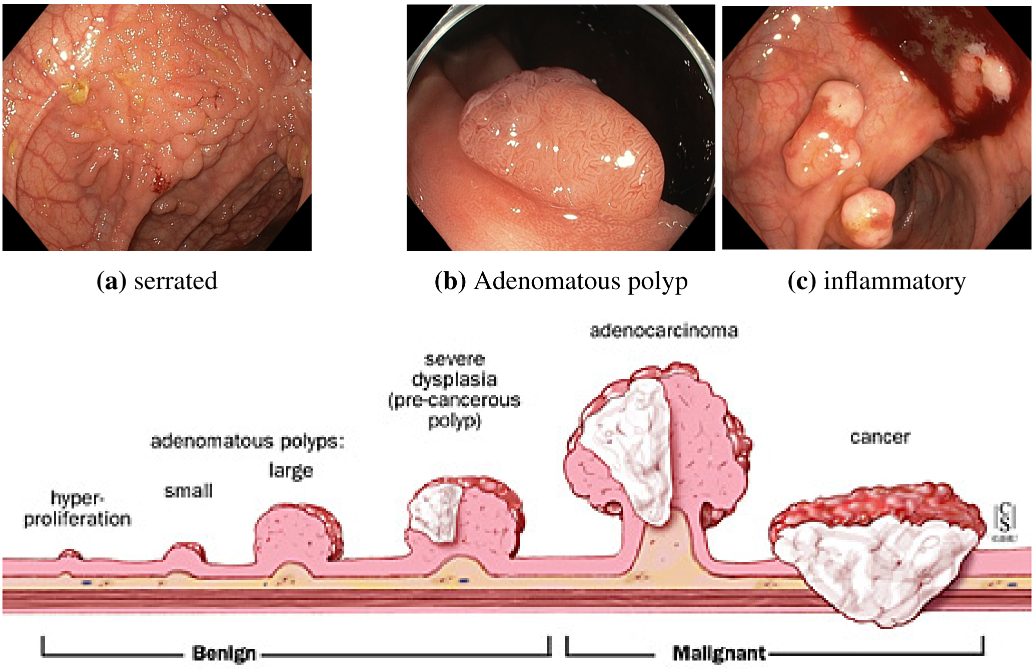
Figure 2.2: The mucosa is the innermost layer of the GI tract. that is surrounding the lumen, or open space within the tube.[2]

Colonoscopy or other regular screening method must be used to detect colon polyps because it often does not give signs of symptoms in early stages. If polyps are detected at early stages, they have not reached a cancerous state and can safely be completely removed from the mucosa.

There exist three main types of polyps i.e, Adenomatous, Serrated and Inflamma-

[2] `https://en.wikipedia.org/wiki/Gastrointestinal_tract#/media/File:Layers_of_the_GI_Tract_numbers.svg`

tory [57]. The most common type of polyp is adenomatous, meaning for patients who have developed polyps, the chances of having adenomatous is around 66% of all polyp cases. There is small chance of them developing into cancerous, but almost all types of malignant polyps are of type adenomatous [30].



(a) serrated

(b) Adenomatous polyp

(c) inflammatory

(d) Polyp progress [3]

Figure 2.3: a,b and c Images from [49] and they show different types of polyp, while image d shows growth and progress of a polyp

Inflammatory polyps shown in figure 2.3c are not as a significant threat like large serrated polyps and the may be caused by other diseases Crohn's disease or ulcerative colitis. These diseases may also increase the overall risk of colon cancer [63].

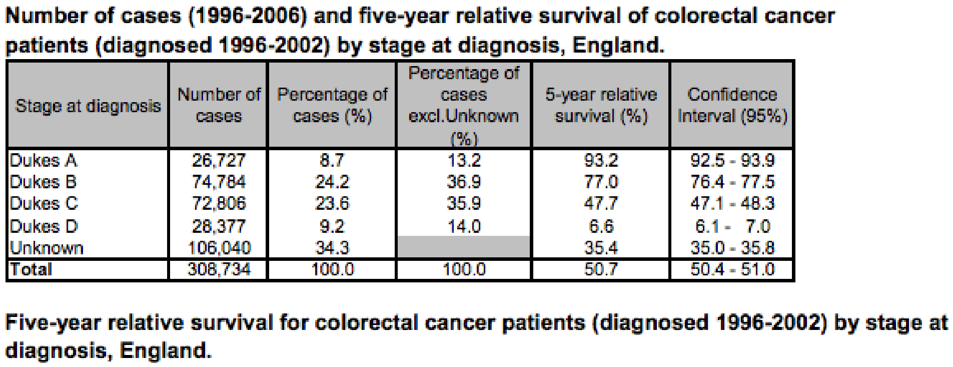
Adenomatous is an early stage of cell change. An image of adenomatous can be seen in figure 2.3b. Irregular margins are characterization of serrated polyps. Depending on size and location of the serrated polyp in the colon, it might develop into cancerous. Smaller serrated polyps also called hyper-plastic grows in the lower part of colon and they are most often not malignant, presented in figure 2.3a. Larger serrated polyps are flat and difficult to detect. It might also develop into precancerous [30] [22].

In our research, we make no distinction between polyp types because the doctor will remove any polyps that would be found under examination, because they can evolve into cancerous polyps.

CRC has up to five stages [45]. Stage A of the CRC is when polyp has not grown out of the innermost layer (mucosa) of the colon. In CRC Stage B, polyp grows out of the mucosa, but it has still not reached lymph nodes or other areas of the colon. Stage C can be described as growth of the polyp through the colon wall reaching neighbouring lymph nodes and organs. Polyp grows into thicker layer of muscle which contracts to force the contents of the intestine along. In CRC Stage D, the polyp has grown through or into the tissues surrounding the colon or rectum. The growth of polyp can be fatal in CRC stage 4. Polyp has grown into the surface of the visceral peritoneum and has grown through all the layers of the colon and starts attacking or already spread to nearby and distant organs such as lungs and liver or other inner structure [45]. The survival rate varies from stage CRC is detected as shown in figure 2.4. As pointed out in the figure, we observe the importance of early detection of polyps. Following the figure, we note that in stages A and B, also referred to as stage 0 and 1, the patient's 5 years relative survival rate is 93.2% and 77.0%. These numbers show how important early detection is, especially in relation to the last stage, where the survival rate has fallen to an alarming 6.6%.

The figure also indicates how many cases were reported for all individual stages, and we clearly see that the number of cases from stage A to stage B and C has increased by 170%. This may be due to symptoms that have potentially developed and the patient feels the urge to get a physical examination, even though the survival rate at that time drastically reduces to 47.7% in stage C. These figures clearly indicate that it is an advantage to routinely check for polyps routinely. According to [22], it has been shown that early polyp screening of large scale populations will improve prognosis and reduce events regarding CRC, which eliminates the possibility of polyps developing into cancer stages.

[3] http://www.hopkinscoloncancercenter.org/CMS/CMS_Page.aspx?CurrentUDV=59&CMS_Page_ID=0B34E9BE-5DE6-4CB4-B387-4158CC924084



Number of cases (1996-2006) and five-year relative survival of colorectal cancer patients (diagnosed 1996-2002) by stage at diagnosis, England.

Stage at diagnosis	Number of cases	Percentage of cases (%)	Percentage of cases excl.Unknown (%)	5-year relative survival (%)	Confidence Interval (95%)
Dukes A	26,727	8.7	13.2	93.2	92.5 - 93.9
Dukes B	74,784	24.2	36.9	77.0	76.4 - 77.5
Dukes C	72,806	23.6	35.9	47.7	47.1 - 48.3
Dukes D	28,377	9.2	14.0	6.6	6.1 - 7.0
Unknown	106,040	34.3		35.4	35.0 - 35.8
Total	308,734	100.0	100.0	50.7	50.4 - 51.0

Five-year relative survival for colorectal cancer patients (diagnosed 1996-2002) by stage at diagnosis, England.

Figure 2.4: Over 90% of patients diagnosed with the earliest stage of disease survived five years from diagnosis compared to only 6.6% of those diagnosed with advanced disease which has spread to other parts of the body[4].

2.1.2 GI Examination Procedures and Methods of Screening

According to the Mayo-clinic there are many people that will not experience any symptoms in the early stages of CRC, although when they first are experienced they will probably differ with relation to the size of cancer and the location in the large intestine.

Due to the relative absence of symptoms associated with CRC, the only sure way to detect it is to be examined. There are several methods to detect diseases in the GI tract. The most common include endoscopy, computed tomography and wireless capsule endoscopy. Each method is associated their respectively advantages and disadvantages. These concern the sensitivity and specificity of the method, costs and time efficiency/inefficiency, access to medical personnel, level of discomfort the patient will experience from one particular method of examination and if it is ethically justifiable.

Endoscopy

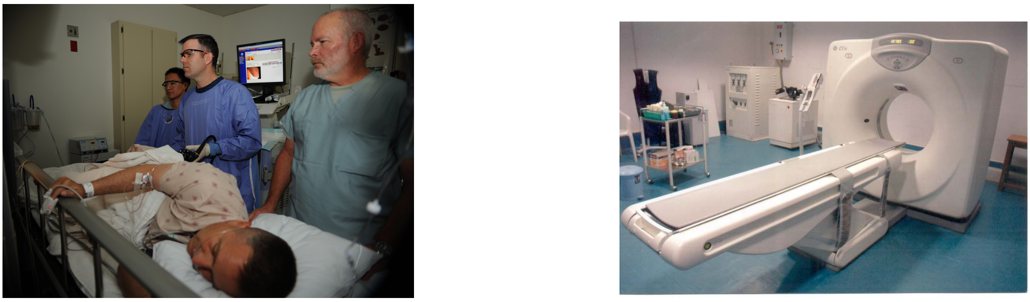
Today, endoscopy is the standard method of detection of CRC or other disease. Basically, an endoscope is a flexible lengthy tube with a camera and light attached to its tip. This device is inserted into the patients body cavity via the oral cavity or the anal cavity. The examination is called gastroscopy (entering orally) or colonoscopy (entering the anus), respectively. In addition, the endoscope has a built-in mechanism for transferring the recorded video to a computer for the doctor or surgeon to analyze.

The particular challenge with the endoscopy examination, apart from the general ones already listed in the previous section, is that the patient is not eager to undergo the examination because of the massive surgical invasion and discomfort.

The biggest advantage of using endoscopy as the screening method is that the physician can examine the cavity of the patient, the food tract and the duodenum if the endoscope is introduced through the oral cavity or if it is introduced through the anal cavity, the rectum and colon. Endoscopy is used for multiple purposes, that is to detect, diagnose and treat cancer. It is the best examination method and diagnostic tool we have to date for detection of cancer.

Although this method presents itself as a method of diversity it also has some limitations. Cost-wise this method makes a big financial impact as it is a very expensive process, but beside cost, one of the most considerable limitations is the fact that it's not possible to traverse through the small intestines of the patient and examine potential diseases or polyps. In addition, the experts are required to prepare for the procedure and in most cases sedate the patient. As for the patient, this process is usually uncomfortable and awkward and therefore undesired. Gastroscopy and colonoscopy are the most common and best examination approach so far to detect polyps and remove them, compared to the other methods like CT-scan. Detailed inspection of the entire colon is made possible as well as removal of any precancerous polyps. When it comes to the latter, the process of removing polyps is handled on the withdrawal of the endoscopy procedure. Preventive effect in colonoscopy is directly connected to the elimination of polyps and cancerous lesions. Since removal of polyps are dependent on the detection, the first essential action that needs to occur during the process of polyps detection on mucosa and get rid of any debris and other remaining waist, like digestive juices from the small bowel.

However, it has downsides and limitations. One of the most considerable limitations is to not be able to traverse through the small intestines with the endoscope and detect potential disease in that area. The downsides are the expenses, in some cases, the patient has to be sedated to perform the examination and is generally very uncomfortable and intrusive.



(a) Colonscopy [49]

(b) CT-scan

Figure 2.5: Screening methods

The main issue with colonoscopy is the relative CRC protective effect against the recurrence of colon cancer. This is especially the case if the colon is not carefully examined, where some factors involves the patients positioning, orientation and preparation before the procedure. On the other hand, recent studies have shown a more distinctive protective effect for CRC of the left colon (last part of the colon connected to the anus) [69] [37]. The difficulties of acquiring a proper protective effect in the right colon have raised some noteworthy explanations. One of the explanations regarding the patient is related to the patients biologically limitations such as genetics and how genes determine the possibly abnormalities in anatomy (colon anatomy), morphology of flat polyps or even a cancerous family history. Continuing on the factors, the patient might have done a poor preparation or does not make a sufficient effort to collaborate during the examination. In addition to these factors, the right colon has another set of limitations and difficulties that makes the process even more challenging for the surgeons; such as the right colon characteristics of deep folds and covering done by the substances from the small bowel and can make inspection not as straightforward.

Computed Tomography

Another screening method is Computer Tomography (CT). For a CT examination, the patient has to lie in a chamber. While in the chamber x-rays are sent through the patients

[4] `https://commons.wikimedia.org/wiki/File:Ct-scan.jpg`

body for the purpose of taking picture of certain tissue. The patient needs to fast 1-3 days as part of their preparation for the CT scan. On one hand, it is not as surgically invasive as endoscopy. The patient does not need to be sedated for example as he or she only need to more or less lie still without too much movement in the chamber. On the other hand, the x-ray radiation is harmful to the patient. Therefore, this method should only be used as a last resort. This method compared to other traditional colonoscopy methods is less invasive due to it's heavy reliance on the use of x-ray equipment like one shown on figure 2.5b , which patients usually favor. Compared to the endoscopy patients would not need to be sedated and the process is less time consuming. Since this screening method probably is not covered by insurance carriers, the procedure gravitates towards the individuals financial expense. The procedure itself also requires the patient to restrict themselves from any dietary for 1-3 days before the procedure and full bowel preparation is required prior to the tomography.

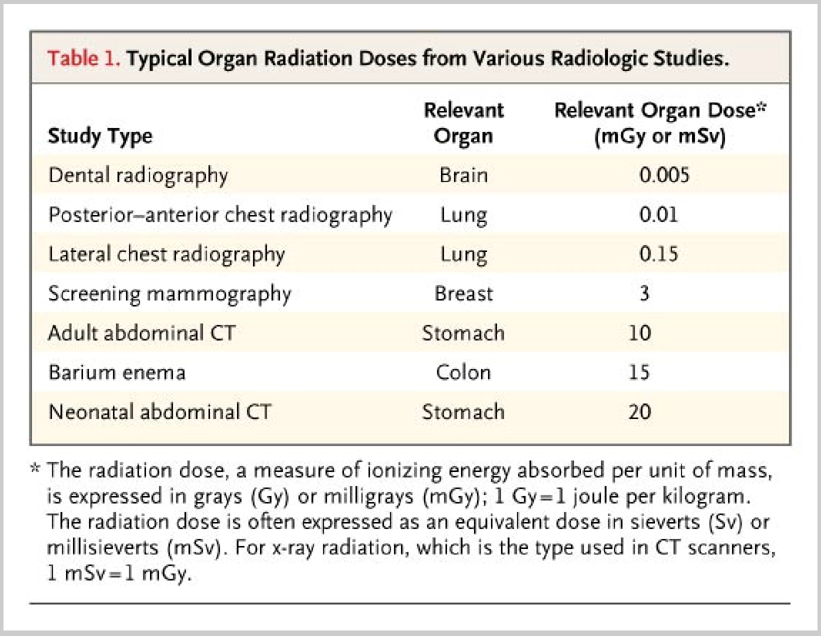


Table 1. **Typical Organ Radiation Doses from Various Radiologic Studies.**

Study Type	Relevant Organ	Relevant Organ Dose* (mGy or mSv)
Dental radiography	Brain	0.005
Posterior–anterior chest radiography	Lung	0.01
Lateral chest radiography	Lung	0.15
Screening mammography	Breast	3
Adult abdominal CT	Stomach	10
Barium enema	Colon	15
Neonatal abdominal CT	Stomach	20

* The radiation dose, a measure of ionizing energy absorbed per unit of mass, is expressed in grays (Gy) or milligrays (mGy); 1 Gy=1 joule per kilogram. The radiation dose is often expressed as an equivalent dose in sieverts (Sv) or millisieverts (mSv). For x-ray radiation, which is the type used in CT scanners, 1 mSv=1 mGy.

Figure 2.6: Organ Radiation [5]

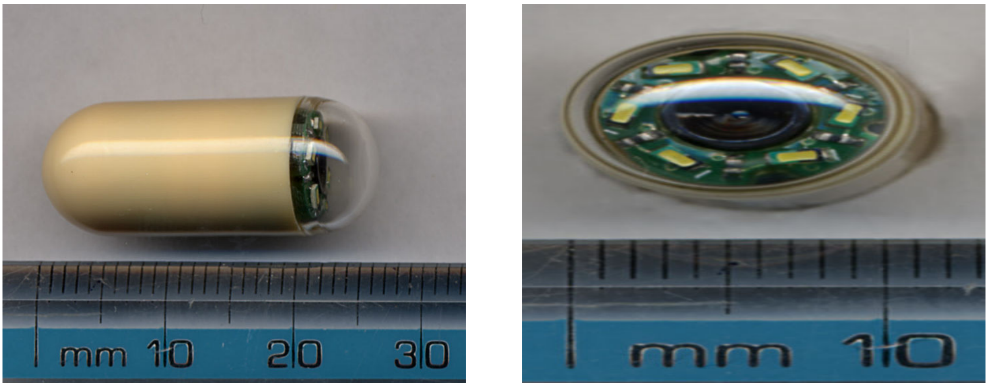
However, CT screening method is presented with some disadvantages, one being that it is hard to identify abnormal tissue or polyps with a limited diameter smaller than one centimeter and with this method physician does not have the ability to collect tissue samples undergoing the examination. The images presented are tomographic images, meaning the image is not a live representation of what we physically see, but rather slices of specific areas of the body obtained from a large series of two -dimensional x-ray images taken in different directions. Although the presence of discomfort is somewhat diminished, the patient is exposed to harmful radiation [24] under the examination.

Wireless Capsule Endoscopy

The recommended regular screening is not scalable for a large scale population and that is why we need a applicable way to reduce time consumption of endoscopy screening for physicians. In modern gastroenterology, Wireless Capsule Endoscopy (WCE) has gained great popularity among the available tools used for polyp detection [52, 27, 11]. In short, in WCE the patient swallows a pill that is about the size of a large vitamin pill. The pill is basically a camera with an image sensor, bleeding sensor, pH-sensor, battery, light source, antenna and wireless transceiver. This method can be compared to just swallowing a pill, which people already are familiar with. The level of discomfort for the patient is practically less compared to the other methods discussed. Thus, the WCE method overcomes issues that are associated with the other methods. It is more widely approved and accepted by patients who favour the less traumatic experience over traditional endoscopy. The challenge though will be reusing the capsules. Patients could hesitate to reuse the same capsules after one cycle through the body, even though the pill can also be thoroughly cleaned and sterilized after each use.

Apart from the level of comfort, this method introduces for the patients, there are many more benefits associated with the WCE method worth mentioning. These include effective use of the physicians time, that is the physician invests less time analyzing each frame of the video to detect polyp. The system will automatically tag/mark the frames in the video containing polyps. These tags will also lead to physician missing less polyps and better scalability of the system overall. Furthermore, it allows the physician to find the cause of bleeding or sources of gastroesophageal back-flow and abdominal pain [58, 66, 7, 42, 35].

The capsule continuously takes images as it travels through the GI tract (see figure 2.1), and at the same time transmitting data. The capsule also enables for medical personnel to analyze and examine the small intestines which doctors could not do with the current method of endoscopy. Capsule endoscopy produces a huge amount data and one of the challenges is to meet the necessary requirements of processing of the data received. For the capsule to completely finish it's route through the digestive system, the whole process might take up to 5 to 8 hours. As a result, the video has the following issues:



(a) Side view [5] **(b)** Front view [6]

Figure 2.7: Wireless-Capsular

- Large file-size
- Time consuming examination
- Expensive in terms of a specialist examination

For the lowest frame-rate with considerably low resolution, the uncompressed hours long video ends up as roughly 30GB in size. Therefore, comparing to a manual examination, the time it takes to analyze the video far exceeds the time of regular endoscopy. This also often leads to increased unwanted expenses. In addition to this the whole process takes

[5] https://commons.wikimedia.org/wiki/File:CapsuleEndoscope.jpg
[6] https://commons.wikimedia.org/wiki/File:CapsuleEndoscopeEnd.jpg

its toll on medical personnel, as a result of ineffective use of human resources. This suggests that we have to developing autonomic applications system that can automatically detect polyps, disease, and abnormality in tissue by implementing Artificial Intelligence (AI). Although the Wireless Capsule Endoscopy has some challenges, the benefits of the finished product are as followed:

- Less burden upon the specialist and medical staff
- Mass screening of the population is made possible
- Early diagnosis
- Early and effective treatment
- Reduction in death rate
- Reduce human-error factor

Current research faces following challenges as well to reach these goals:

- Lack of test data set
- Motivation among people
- Lack of suitable tools

Automated Computer Diagnosis

Computer Aided Diagnosis (CAD) is based on the concept that when computers have reached a level of trustworthy certainty through countless hours of training and precision tuning, they can assist medical professionals with a second opinion. The goal is to make machines diagnosis reach the level of the medical experts. By taking advantage of computers, the expert can increase their performance, which also leads to more efficient use of time and medical resources. Although the last and final decision lies with the medical professionals.

On the other hand, Automated Computer Diagnosis (ACD) further improves the idea of CAD. The purpose of using machines is automation, and by taking advantages of what machines do best we have the ability to contribute and improve the current medical standards. The process which involves of having the expert verify is now out of the question, although not entirely. This enables the health care system to evolve from the current methods of detecting polyps. The present solution is based on patients take initiative and have their colon examined, which requires the endoscopist to delegate time and effort just to see if there are any polyps present. With the ACD approach, we have the potential to use the wireless capsule which traverses through the entire GI tract and marks the video with what time a polyp was detected. In this point of time, the medical professionals has to make a decision on whether the patient needs further examination and removal of polyps or not. The performance level needed for the ACD system has to ideally be at least at the level of medical professionals. If the ACD system on the other hand would perform worse that the medical standard, it would be hard to justify its use and would not make sense to deploy.

2.2 Related work

This section will cover the related work, mainly covering polyp detection and real-time object detection. The state-of-the-art systems, also shown on table 2.1, have applied different approaches in colons-copies and polyp detection. The first approach from Wang et al.[64] is one of the most recent and prominent research in the field of polyp detection. As also shown on the table 2.1, different approaches uses different data-sets for testing and training their model, and measure polyp detection performance with different metrics. The last row shows EIR system, which is further described under section 2.8, is groundwork for our research. However, we will give a brief description of few of them below. These algorithms, methods and partial systems present at first glace promising results in their closed testing environment. However it is unclear how they will perform at hospitals, because many of the researches were conducted with small data-sets. Over-fitting can be a problem when tests are run on small data-sets.

Wang et al. [64] presents a fast polyps detection system ”Polyp-Alert”, taking advantage of previous edge-cross section visual features and a rule-based classifier [65]. The polyp-alert system calculates polyps in entire video instead of per frame. Their system was able to reach 97.7 % of detection rate and could detect 42 of 43 polyps in total 53 videos. The system used object tracking to track the polyps in preceding and subsequent frames and was able to reach 95.70 % accuracy and 97.70 % recall. The system could reach up to 10 FPS

Mamonov et al. [34] introduces an algorithms for two-class classifier to detect polyps in the colon. This algorithm is based on image geometric information extraction and creates a basis for a binary classifier to categorized into two classes; positive and negative, so less frames will be needed to manually inspected. The algorithm processes on one frame at a time and classify them as positive or negative based on a polyp has been detected or not. On a data-set of 18.738 frames, Mamonov et al. system could only reach 47% recall rate, however the sensitivity was around 81.25% with a specificity level of 90%. The data-set was gathered from five different patient videos and the findings was evaluated. The data-set contained 16 polyps and the input sequence varied between 2 - 32 frames, and a total of 16 sequences were tested. Similarly to the Wang et al. system, Mamonov et al. system was also based on detection per polyp and had 9.8% false positive rate.

Table 2.1: We see that some performance measurements are not available for all methods. Nevertheless, including every available information gives an idea about each methods performance [50].

Publ.System	Detection Type	Recall/ Sensitivity	Precision	Specificity	Accuracy	FPS	Data-set Size
Wang et al. [64]	polyp / edge, texture	97.70 %	N/A	N/A	95.70 %	10	1.8m frames
Wang et al. [65]	polyp / shape, color, texture	81.40 %	N/A	N/A	N/A	0.14	1.513 images
Mamonov et al. [34]	polyp / shape	47 %	N/A	90 %	N/A	N/A	18.738 frames
Hwang et al. [23]	polyp / shape	96 %	83 %	N/A	N/A	15	8.621 frames
Li and Meng [28]	tumor / textural pattern	88.6 %	N/A	96.20 %	92.4	N/A	N/A
Zhou et al. [70]	polyp / intensity	75 %	N/A	95.92 %	90.77	N/A	N/A
Alexandre et al. [3]	polyp / color pattern	93.69 %	N/A	76.89 %	N/A	N/A	35 images
Kang et al. [25]	polyp / shape, color	N/A	N/A	N/A	N/A	1	N/A
Cheng et al. [9]	polyp / texture, color	86.2%	N/A	N/A	N/A	0.074	74 images
Ameling et al. [4]	polyp / texture	AUC=95%	N/A	N/A	N/A	N/A	1.736 images
EIR [[49][48]]	abnormalities/30 features	98.50 %	93.88 %	72.49 %	87.70 %	300	18.781 frames

2.3 Classification

Amit P and Dr. R.C.Jain prescribe classification as “ The task of classification occurs in a wide range of human activity. At its broadest, the term could cover any context in which some decision or forecast is made on the basis of currently available information, and a classification procedure is then some formal method for repeatedly making such judgments in new situations” [54]. They further divide classification into two different types, unsupervised learning and supervised learning also known as guided learning. The first is a set of observations to detect or distribute data in different classes. The second, guided learning can be described as creating new rules based on existing and well-known classes for new observations of them. Nave Bayes classifiers, SVMs, Neural network approaches and binary classifiers are few of many classifiers used in supervised leaning. In our project we classify supervised leaning with machine learning methods such as Decision tree, Artificial network, K-nearest neighbour to predict improvised results. These and other methods we use have been listed and explained in detail in chapter 2.

2.4 EIR - Efficient computer aided diagnosis

As previously noted, early detection of polyps reduces the risk of death considerably. It has also been mentioned that the time it would take for mass scale examination is one of the main challenges in colonoscopy. In order to support mass scale examinations so we need to develop a system that help doctors in polyp-detection. A system which makes it possible to live-stream colonoscopy and also automatically do first line screening for GI tract, with the use of wireless video capsule endoscope. To further aid and scale such examinations we will work on EIR System. An effective system which is also capable of retrieving scalable information for medical data like videos and images [49, 43, 48, 44, 47, 50]. The EIR system consists of the following three subsystems.

- Annotation and knowledge transfer subsystem
- Detection and automatic analysis subsystem
  - Detection

  - Localization

- Visualization and computer aided diagnosis subsystem

The annotation subsystem gathers and transmit training data from the medical experts into the subsystem. The training data are important for good classification in medical field [49]. It is possible to add different global features for classification in EIR at once but not all features provide us with required information in detection. The more features we add, the more computational power is required for classification. It is therefore important at the beginning to decide what features would benefit us and are required. The modular designed subsystem, detection and automatic analysis of the diseases is divided into two subsystems, detection and automatic analysis. It is possible to extent this subsystem further into different diseases or their subcategories. The detection part detects irregularities in the frames. The job of this subsystem is to check whether there are some irregularities in the current frame or not. The output from the detection-part is input for the location-part subsystem. The localization part locates the accurate spot of the disease in the frames [49]. The output from the detection and analysis subsystem is viewed in visualization subsystem to medical expert for additional analysis. Figure 2.8 shows the entire EIR system architecture, and its data stream.

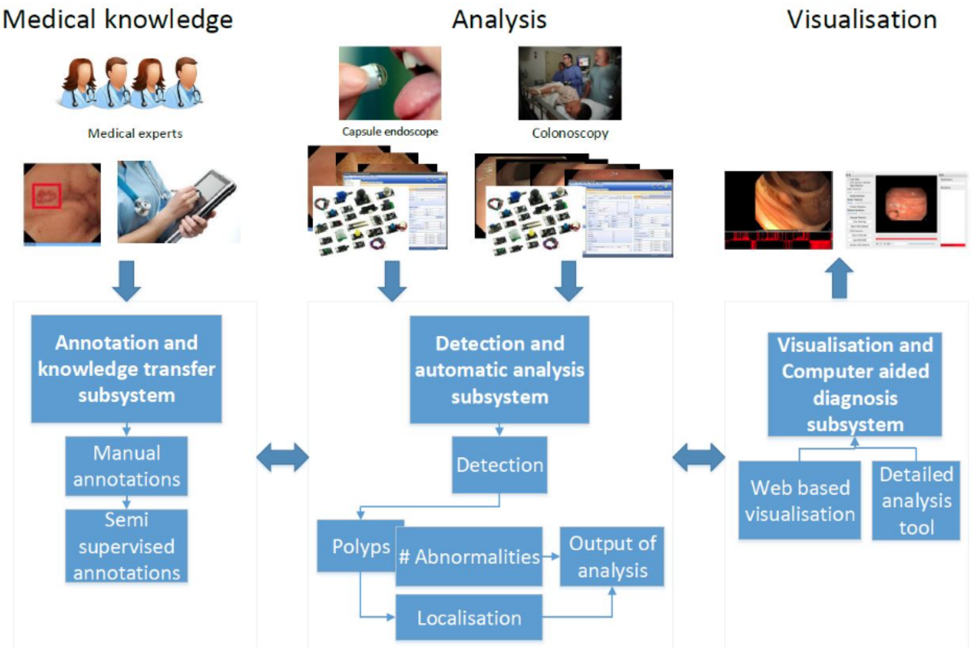


Figure 2.8: A complete overview of the EIR system. The system consists of annotation, detection and automatic analysis and visualization subsystems [49].

2.5 Summary

In this chapter, we have reviewed different diseases in GI tract, although focusing mainly on polyps. We have also discussed GI examination procedures such as gastroscopy, colonoscopy, computed tomography and other currently used screening methods at the hospitals. In future, computer will play a major role in screening and for medical purposes, so we also reviewed CAD, WCE and ACD. These new procedures need to perform at least at the same medical set standards to be able to fully replace the existing polyp detection procedures. EIR, a modern research in this field has shown promising results, and contains a complete pipeline for annotation, detection and visualization of the diseases in the GI tract. Object detection and classification by EIR is done with global image features. In our research, we will use the idea of EIR to implement a real-time polyp detection system on computers without GPU and generally low computational resources.

In the next chapter, we will discuss global and local features and how they potentially bring advantages to our system.

Chapter 3

Real-Time Polyp Detection

A polyp often appears in very different shapes and forms compared to the previous frame, although we are dealing with the same polyp. This phenomenon occurs due to the viewing angle and the polyps distance to the traversing camera, the amount of colon movement and insufflation and contraction from colon muscular systems. Therefore, we are experiencing that existing detection techniques on images containing polyps, require lengthy processing time due to the appearance of polyps and their complex characteristics. The explanation we have of why colonoscopy in some cases still fails to prevent most CRC related death, is that lesions sometimes goes under the radar and are not detected or effectively removed.

During the procedure of colonoscopy, having real-time feedback has the potential to not only alert the endoscopist, but also more efficiently assist the experts in the overall procedure to have them completely removed [13]. According to the research on Near real-time feedback during colonoscopy done by Iowa State University of North Texas [64], a detection rate of 90% comes with a false alarm rate of 1 false region per image. During a colonoscopy it is inadequate for a real-time feedback system to have high false alarm rate. This will in turn increase the total analysis time as part of the automated system, which is the opposite of what is intended. Having a system where analysis time is close to real-time will have a huge benefit for the endoscopist. The fast analysis time also implies that the system has low latency and movements feels smooth. As the endoscopist is examining the colon, the doctor gets additional assistance by the detection mechanism of the system to predict the present of the polyps. With extensions and further development in our system, the doctor will also be notified where the polyp is located in the frame. This is extremely beneficial when the doctor is fatigue or simply just overlooked a polyp. This even enables the doctors to verify their findings with the documentation that the system produces after the procedure, and can be utilized by universities in educating students to become future endoscopists.

In this chapter, we will describe the building blocks of our system and dive deeper into how they work and which part is useful for our purposes. The chapter is divided into four main sections, features used to describe the data-set for our system to execute classification, feature selection, Machine learning algorithms and the LIRE open source platform [33] [32]. LIRE is one of many essential parts of our system and stands both for feature extraction and ranking, which will be further described in section 3.5.

3.1 Global- and Local Features

A global feature is a kind of transitioning tool used to capture and describe the overall content of a medium and convey to the machine. In general, features are the extracted information from images in terms of numerical values, which for us humans are challenging to understand. In our case, we have tried and experimented with several features and even used a more unorthodox approach to find the right feature to increase classification results with our classification improvements described under section 4.8. If we were to ask the machine to look for similarity between two identical images with a feature looking for color, except for one of the images being in the black and white color spectrum, the machine would find some or none similarity at all. So, the feature selection is essential to try and make the machine to differentiate between smooth or rough surface, kind of texture, color or better yet, a polyp within a frame.

We also have local features, which mainly serves as the role of referring to distinct structures and patterns recognized in images. These can be edges, small image patches or even a point in the image. These noticeable patterns which are correlated with the image patch, more often than not, differs from the color, intensity and texture around the image patch. The local feature's purpose is to locate the image patch and try to highlight this area. The benefits and applications of local features are that they let us find image correspondences despite occlusion, presence of cluttering or changes in view conditions. When it comes to the area of application we fundamentally see it applied in image stitching, 3-D reconstruction and to compactly express image contents for classification and detection without any image segmentation [31, 48].

3.1.1 Feature selection

To find out which features to choose, Riegler et al. [49] ran different experiments where they tried various features with the help of EIR supported system. They also experimented different possible combinations of features to find promising results. These experiments were performed on one video to refrain from over-fitting on the data-set. This happens especially in scenarios where very rare and specific random features in the data-set is unique and different from the rest of the data samples or learning is performed too extensive [20]. We have used these features to extract information from images in our *indexer.java* program, and ranking with implemented search method in *searcher.java* to execute classification.

Riegler et al. further mentions that in these feature combinations, there are only insignificant variation in term of accuracy and detection. Although most of them give promising results as shown in table 3.1, which mean these can be used. Generally, CEDD achieved best results with 95.48% Recall, 76.86% Precision and 85.17% F1-score. The table also reveals that other features like JCD, Tamura etc. achieve similar positive results. Based on these results, we have chosen to use these features in our research.

Tamura

This feature is based on the psychological view of the human eye and how humans assume specific characteristics of an image texture. Tamura compare the perception of six core image characteristics such as coarseness, contrast, direction of texture, the likeliness of lines, roughness and regularity. These features have been researched with psychological measures taken from experiments done on human participants [61]. The three features

Table 3.1: Leave-one-out cross validation with different features

Leave-one-out cross-validation for all, by the EIR system supported, features [48].

Feature	**True Positive**	**True Negative**	**False Positive**	**False negative**	**Precision**	**Recall**	**F1 score**
JointHistogram	3.369	13.826	1.085	511	0.7563	0.8682	0.8084
JpegCoeffHist.	3.224	13.772	1.139	656	0.7389	0.8309	0.7822
Tamura	3.392	13.861	1.050	488	0.7636	0.8742	0.8151
FuzzyOppHist.	3.341	13.552	1.359	539	0.7108	0.8610	0.7787
SimpleColorHist.	2.736	13.563	1.348	1.144	0.6699	0.7051	0.6870
JCD	3.556	13.777	1.134	324	0.7582	0.9164	0.8298
FuzzyColorHist.	2.708	13.243	1.668	1.172	0.6188	0.6979	0.6560
RotInvtLIBP	3.479	13.829	1.082	401	0.7627	0.8966	0.8243
FCTH	2.846	13.671	1.240	1.034	0.6965	0.7335	0.7145
LocBinPattAOpp	2.412	13.349	1.562	1.468	0.6069	0.6216	0.6142
PHOG	2.879	13.806	1.105	1.001	0.7226	0.7420	0.7321
RankAndOpp	2.527	13.553	1.358	1.353	0.6504	0.6512	0.6508
ColorLayout	2.702	14.018	893	1.178	0.7515	0.6963	0.7229
CEDD	3.705	13.796	1.115	175	0.7686	0.9548	0.8517
Gabor	1.849	10.643	4.268	2.031	0.3022	0.4765	0.3699
OpponentHist.	2.246	14.157	754	1.634	0.7486	0.5788	0.6529
EdgeHistogram	3.548	13.737	1.174	332	0.7513	0.9144	0.8249
ScalableColor	3.231	13.684	1.227	649	0.7247	0.8327	0.7750
Late Fusion	3.710	13.894	1.017	170	0.7848	0.9561	0.8620

coarseness, contrast and direction, have been evaluated to achieve best results.

The most fundamental texture feature was observed to be the coarseness, which had an explicit correlation with scaling and repetition rates. The image can contain several scales where each scale has its own texture formation. Coarseness has the intention to identifying the occurrence of the largest formation of a texture and even the smallest texture at micro level.

The gray levels and their dynamic range in an image is captured by the contrast, alongside the polarization of the distribution of white and black. The dynamic range is found by measuring the standard deviation of the gray levels, and the black and white polarization is found using the kurtosis.

In an image, the texture direction found in a small area is the global property over the span of the texture region. This description of direction does not intend to discriminate between patterns or orientations, but rather measure the degree of the direction of the image's entirety. This is done by calculating the magnitude and angle of each pixel. By counting all pixels where the magnitude is greater than a given threshold and quantizing by the edge angle, it is possible to visualize the edge probability and reflect the degree of direction in addition to the sharpness.

CEDD

Color and Edge Directivity Descriptor (CEDD) is an image feature that extracts and combines the edge information with the color information to create one histogram containing all the values. Since the feature size is restricted to 54 bytes, which makes this a suitable feature to use for large scale use cases [8]. One of the noteworthy attributes of this feature is that, this features requires low computational resources for the feature extraction.

JCD

The Joint Composite Descriptor (JCD) fuses two compact composite descriptors (CCD) into one. It is possible to fuse multiple different features in one descriptior by taking advantage of CCD. JCD fuses CEDD and the fuzzy color and texture histogram (FCTH). FCTH uses the high frequency to band the haar wavelet transform in a fuzzy system, to form 8 texture areas. JCD is made of 7 textture areas, with each area made up of 24 sub regions that correspond to color areas.[68]

3.2 Feature Combination

EIR supports different global and local features and their combination. These combinations give different result based on included vectors, features and dimensions and this has been a hot topic for feature combination researchers. The refined combination of different vectors, features and dimensions helps finding the optimal classification and search outcomes. When to apply these feature combinations can also be tricky and affect desirable outcomes [12, 19, 67]. There are two ways of combining features, early fusion and late fusion. Early fusion is primarily combining different features outcome within a single representation. This representation has passed down to decision making step. Features are fused after decision making step in late fusion.

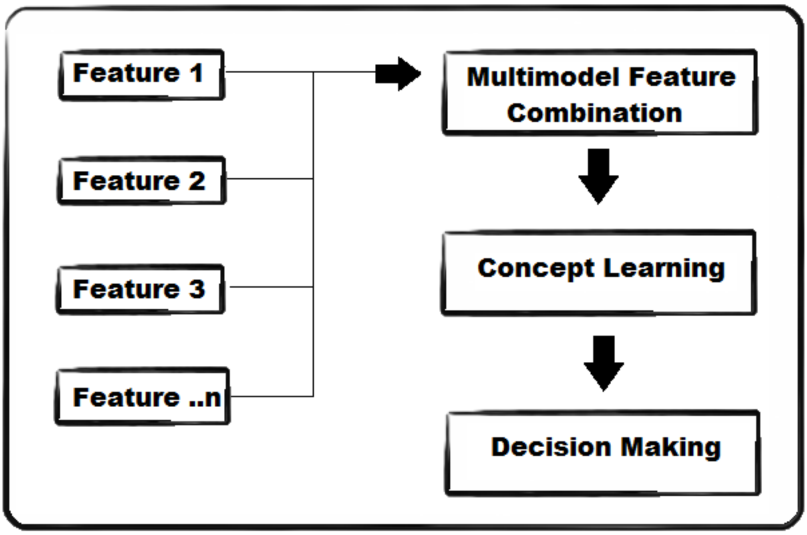


Figure 3.1: A prototype of the Job Information dialog

3.2.1 Early Fusion

The basic concept in early fusion is to first extract uni-modal features and analyze various uni-modal streams. The extracted features are fused within a single representation. A detail sketch of early fusion is shown in figure 3.1. The representation is one large vector combination of feature values, and this can be used to classify or searching tasks. Too much noise in the representation is included by useless features values and this is the negative aspect of early fusion [12]. In early fusion it is also difficult to combine features into a common representation.

3.2.2 Late Fusion

Just like in early fusion, the basic concept of late fusion also starts with extracting uni-modal features first. Instead of fusing all these uni-modal features together, they are forwarded to unique pre-classifier each. These pre-classifiers outcome are afterward fused together to yield a final detection score. A more detail sketch is shown in figure 3.2. An advanced and well selected combination strategy can give an improvement in the classi-

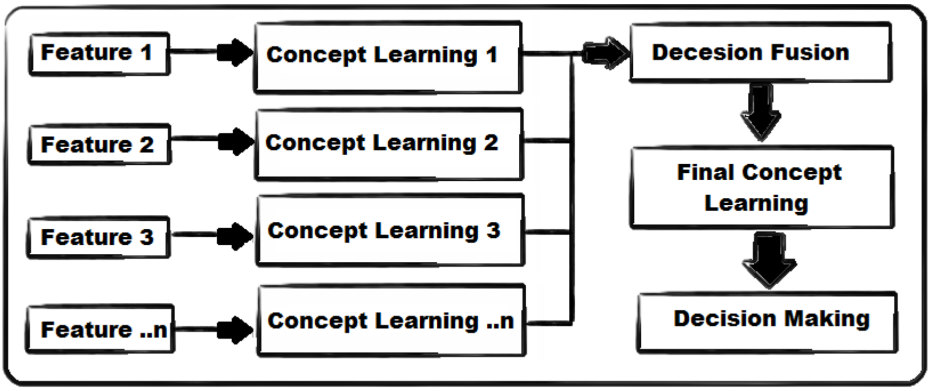


Figure 3.2: Late fusion sketch

fication outcome. Since each features are passed through unique pre-classifier and then to fuse output, it is passed through a new classifier. Because of these extra classifier step, this expensiveness in terms of the learning effort is the biggest disadvantage of late fusion. There also exists risk of losing correlation in mixed feature space if different features are fused together.

3.3 Machine Learning Overview

In computer science, we have the subcategory Machine Learning, which gives the computers the power to learn and execute/make decisions based on what was learned without explicitly being programmed to carry out certain actions. This research focuses on the four most prominent machine learning classifiers, being Instance-based algorithms, Deep Learning, Support Vector Machines (SVM) and Clustering. These methods have been researched quite a bit and are considered traditional approaches in machine learning. Supervised and unsupervised algorithms are two categories in machine learning. Supervised algorithms are based upon the fact that each input is associated with an output, which is also referred to as the desired output. Meaning we as programmers give the machine the right answer and based upon that the machine can estimate an output not far from the desired output by minimize the error. This approach also consists of labelling classes. In contrast we also have the unsupervised algorithms which are unlabeled. This results in the in-availability of calculating the accuracy of the model. This is one way of distinguishing between unsupervised learning from supervised learning. Since unsupervised algorithms do not need training data it is often hard to explain the algorithms outcome because the class is unknown or has not been labeled. This is the reason why unsupervised algorithms are used to understand the unlabeled data end explore it.

3.3.1 K Nearest Neighbour

The K-Nearest Neighbour (K-NN) is an instance based algorithms, meaning it learns from previously labeled training data and makes a prediction. As new data gets passed, the algorithm compares these new data points with training data points that has been recorded. By using similarity functions, such as euclidean distance, the algorithm can determine which class this new data point has more similarity with. These similarities are denoted and seen by the machine in terms of the data‘s features. The number of features used can be viewed as the domain which the computer understands each data. Therefore it is crucial to use features that can fully describe the data to the computer in the same way we humans experience it.

The K-NN algorithm specifically operates by using the training data to map data points and associating a central point to those data points that are close to each-other, making up one class. As shown on the figure 3.3, this central point is the mean of those data points. As new data gets input the algorithm predicts which class it belongs to based on distance to each central point [41, 26]. After assigning the new data to a class the algorithm then calculates a new central point, taking the new data point in consideration. This ultimately expands the class environment based on the data points and how much different one data point is to other data points from the same class.

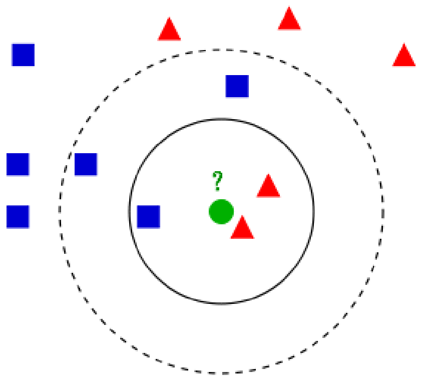


Figure 3.3: The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 (solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If k = 5 (dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle) [1].

3.3.2 Artificial Neural Networks

In Neural Networks [53], the existence of different types of network are numerous. The networks are usually characterized by their components, which are a set of nodes and the connection between them. In one specific type of network, the nodes are seen as artificial neurons (see figure 3.4). These network are called artificial neural networks (ANNs) and are comparable to the observed behavior of our brain. The artificial neuron itself is a computational model which is inspired by the brain-cell neurons. In reality, these neurons receive a signal through the synapses which are located on the dendrites of the neuron. The requirement for the the neuron to activate lies within the strength of the signal received. If the signal strength suffice in terms of surpassing a certain threshold, the neuron activates and emits a signal further down the line through the axon. The signal sent by the neuron may influence other neurons and determine theirs activation. The connection between nodes are assigned a weight which is one of the components in the calibration of the network through the back-propagation.

[1] https://upload.wikimedia.org/wikipedia/commons/e/e7/KnnClassification.svg

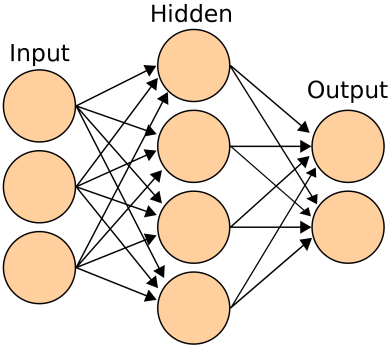


Figure 3.4: Each input point is a high-dimensional vector. The neural network is organized in a series of layers, where the input vector enters at the left side of the network, which is then projected to a hidden layer. Each unit in the hidden layer is a weighted sum of the values in the first layer. This layer then projects to an output layer, which is where the desired answer appears. [2]

The concept of back-propagation is similar to the calibration of any given equipment. The goal is to minimize the error so we get as accurate readings as possible. This approach is within the domain of supervised learning, given the fact that we have to provide the machine with desired output values which is also seen as the validation set.

3.3.3 Random Forests

Random forests or random decision forests [29, 21] are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

We start with a suitable collection of data including variables we would like to predict or understand and relevant predictors. We then can draw a random sample from our main data-set and build a decision tree on this random sample. This ”sample” can be half of the total data although it could be a different data portion of the master data-set. This process can be repeated by creating a second random sample and grow a decision tree over it. This prediction, which was made by this second tree, will usually be different

[2] https://upload.wikimedia.org/wikipedia/commons/thumb/e/e4/Artificial_neural_network.svg/2000px-Artificial_neural_network.svg.png

(at least slightly) than the first tree. This process can be repeated and more trees can be created. After each three is built, all of the data are run down the tree and predictions are compared for each pair[3]. If two cases predict the same terminal node, their prediction is increased by one. At the end, the predictions are normalized by dividing by total number of trees. Predictions are used to replace missing data and locate outliers.

Over-fitting is a common problem with different algorithms such as decision trees, and random decision forests correct this habit to their training set [17] (page 587-588).

3.3.4 Decision Tree

To make a conclusion about an item's target value, the decision tree learning applies a decision tree as its predictive mode [46]. These values are represented in the leaves, while observation of the items, which is made progress from towards the conclusion, are represented in the branches. This approach is used as data mining, machine learning and statistics where the target variable can form a discrete set of values called classifications trees; in these tree structures, branches illustrates conjunctions of different features that lead to class labels, leafs.

3.3.5 Ada Boost

Adaptive boosting ”Ada boost” [51], is a fast algorithm which focuses on classification problems and is intended for boosting weak classifiers into stronger one. Ada boost is simple and flexible algorithm.

3.3.6 Naive Bayes

Naive Bayes methods [36] are a set of supervised learning algorithms based on applying Bayes theorem with the naive assumption of independence between every pair of features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes lin-

[3] https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

ear time, rather than by expensive iterative approximation as used for many other types of classifiers.

3.3.7 Support vector machines -SVM

Another supervised learning method which is used for classification, outliers detection and regression is Support vector machines (SVMs) [10].

In SVM, a data points are viewed as a p-dimensional vectors, and to separate such points from each other and classify them (linear classify), there are many hyperplanes that help classifying data. Hyperplane with the maximum distance between classes are the best one to choose. As shown in figure 3.5, we can see that how a support vector machine would choose a separating hyperplane for two classes of points in 2D. H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin and would be desirable. The advantages with the SVMs are that, it is effective in high dimensional spaces and uses a subset of training points in the decision function, so it is also memory efficient. Disadvantages with the SVMs could be that, the number of features must be lower than the number of samples, the method will otherwise perform poorly.

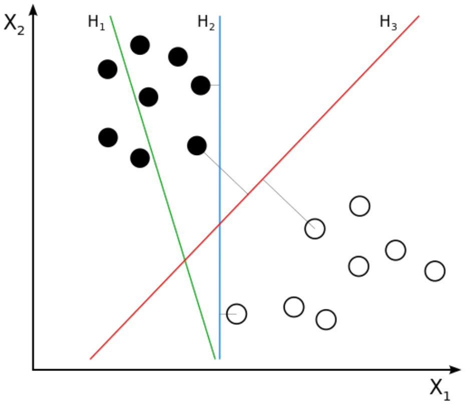


Figure 3.5: Graphic showing how a support vector machine would choose a separating hyperplane for two classes of points in 2D. H1 does not separate the classes. H2 does, but only with a small margin. H3 separates them with the maximum margin[4].

[4] https://commons.wikimedia.org/wiki/File3ASvm$_{s}eparating_{h}yperplanes_{(}SVG).svg$

3.4 Deep Learning

The improved classification improvements version introduced under iteration IV 4.8 is based on deep learning and focuses on to take advantage of the deep learning classification and use it for our medical scenario. In particular, we trained a model based on the Inception v3 architecture [59], which is a deep learning architecture designed for image classification, using the ImageNet dataset [14]. We use Google Tensorflow classification model, [2] which has the property to classify images and assign confidence percentages to each 1008 object categories also known as classes in Inception v3. This means that an image that contains an afghan-hound would likely have higher confidence percentage in class 1 and a lower confidence percentage in class 2, since class 1 represent Afghan hound and class 2 represent African chameleon, also shown in table 3.2.

Table 3.2: This table shows how the meta-data for the 1008 class probabilities are structured and prepared for training. N equal the number of images.

	Class 1	Class 2	...	Class 1008	Pos/Neg
Filename 1	Afghan-hound	African-chameleon	...	zucchini-courgette	1 / 0
Filename 2	0.0002	0.0004	...	0.0004	0
⋮	⋮	⋮	⋮	⋮	⋮
Filename N	0.0003	0.0004	...	0.0045	1

What we obtain later from the Inception v3 architecture is feature vectors with 1008 feature components for each image. This feature vector is then given to machine learning algorithms to train a model that can distinguish between the classes based on observed patterns in the feature vectors. It is therefore interesting to see if the inception v3 model that has previously been trained on edge detection and faces, among many others, to classify objects and look for any significant results.

3.5 The LIRE platform

LIRE [33, 32] is a Java library that provides a simple way to retrieve images and photos based on color and texture characteristics. LIRE creates a Lucene index of image features for Content Based Image Retrieval (CBIR) using local and global state-of-the-art methods.

3.5.1 Indexing

Indexes can be generated from each format of a global feature, regardless of if we want to create one index per feature or one index containing several features, where the latter is the used approach in this research. The structure of the index is based on fields and rows, meaning the first field in a row is the image path and the next is where the representation (meta-data) is generated, whether it be hash representation of the feature values or binary values. A feature vector can have many feature values, in fact one of our feature vectors we tested has over a thousand values. The number of rows are dependent on the number of images indexed, but the number of values in each row is decided by the amount of feature values the chosen feature has [48].

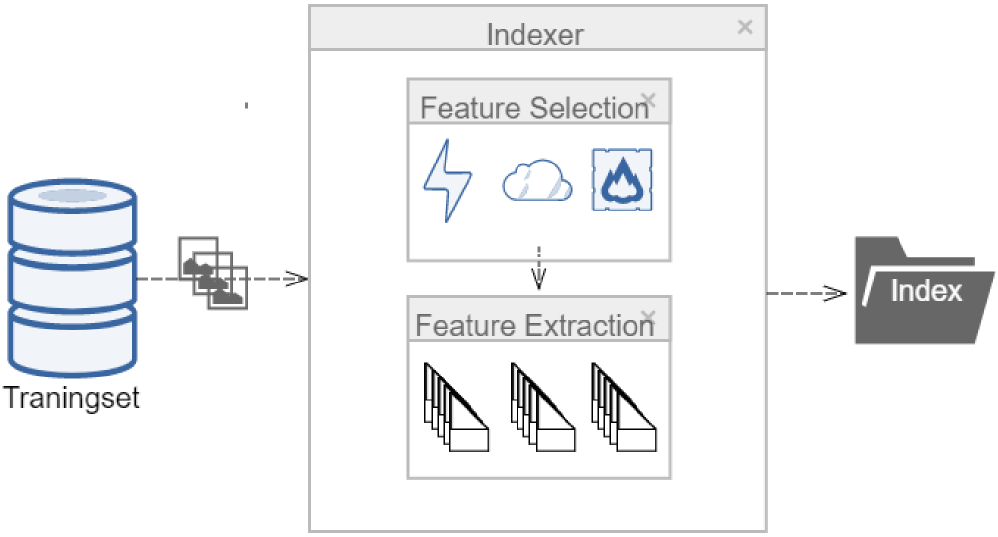


Figure 3.6: The figure shows the pipeline for feature extraction of the images in the training-set.

The approach using hashed values is based on Locality Sensitivity Hashing (LSH). The goal is to use several arbitrary hash functions and hash feature values. This results in similar images getting the same hash values and therefore be hashed into the same faction. In the image's feature space, Linear projection in random directions of hash functions makes this structure possible. In the sense of effectiveness, the created hash codes are not the optimal solution due to the need of many large hash tables to achieve reasonable quality, with regards to search. Due to the algorithm's increased speed these minor disadvantages are, for the sake of research, ignored [55].

Making the colonoscopy live-stream in the Java programming language grants us the advantage of utilizing the LIRE open source library, which is index and search based. As shown on figure 3.6, this method's premise is that we have our training data indexed, meaning we use features of our choice to extract information from each image in the training data. This will create a folder which contains the information of all the training data, although, at a significantly lower space cost, and only contains relevant information for our classification. Searching through the meta-data associated with each image makes this approach much more time - and resource efficient than the RAW image itself.

3.5.2 Search

Our search-based algorithm, searches through the indexed training-set and compares feature values to the feature values extracted from the received frame and is performed on each retrieved frame done in real-time. As a result, our developed search method returns a ranked list for each image given as input when the searcher method is called. This means that for each query image has created a term-based query from the hashed feature values of the query image at run-time. As illustrated in figure 3.7, we further obtain a list which is ranked based on the similarity between the query image and the created index of the training-set. This means that best ranked picture, first element in the list, has the best similarity rate to the query image the camera retrieved. Based on feature selection, the ranked list may be distorted or give a realistic representation, where the latter is the desired outcome. The list is created by calculating the dissimilarity or distance associated with the low level features. The distance function that computes the similarity for the ranking is called the Tanimoto distance [62].

As mentioned earlier, the images in the index folder with good match to the frame passed by the camera generates a lower value, and thus is ranked better in the list. In our

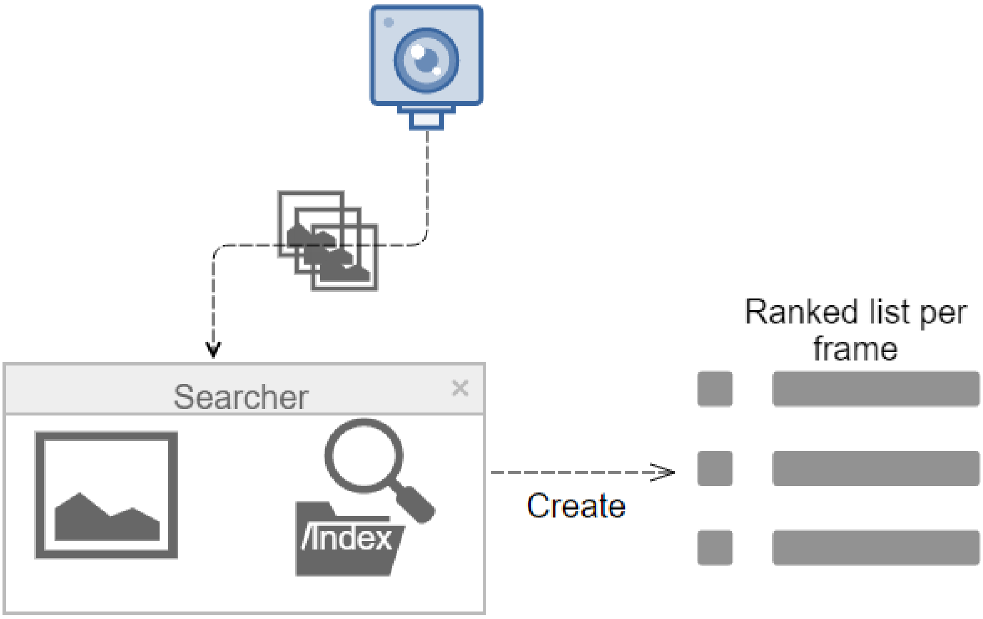


Figure 3.7: Pipeline for the image ranking

case, the lower number means better rank in the list. Further, the list is then sorted by rank and used for by the classification algorithm later down the pipeline. For us to be able to make decision and actually classify the frame as positive or negative, depends on two important aspects. Firstly, we have to consider which features would suit best our case with respect to the task at hand, and secondly, if combining features will improve classification in general.

3.6 Model Creation

In figure 3.8, we illustrate how the our real-time detection pipeline is structured. The pipeline for frame flow is initialized where the camera is enabled and captures frames. This video-feed is then split, one is shown to the endoscopist without any processing, like normal. The other video-feed sends the frame one at a time through our classification pipeline. Our system converts the frame into the data-type of Buffered-Image, making the frame easier to work with. Our searcher method then uses the frame to create a ranked list by searching through the indexed training-set and comparing the indexed images to the classifying frame and base the ranked scores on similarity. Our developed algorithm uses the scores in this ranked list to calculate which of the categories this ranked list was in favor to, and then predicts the class label.

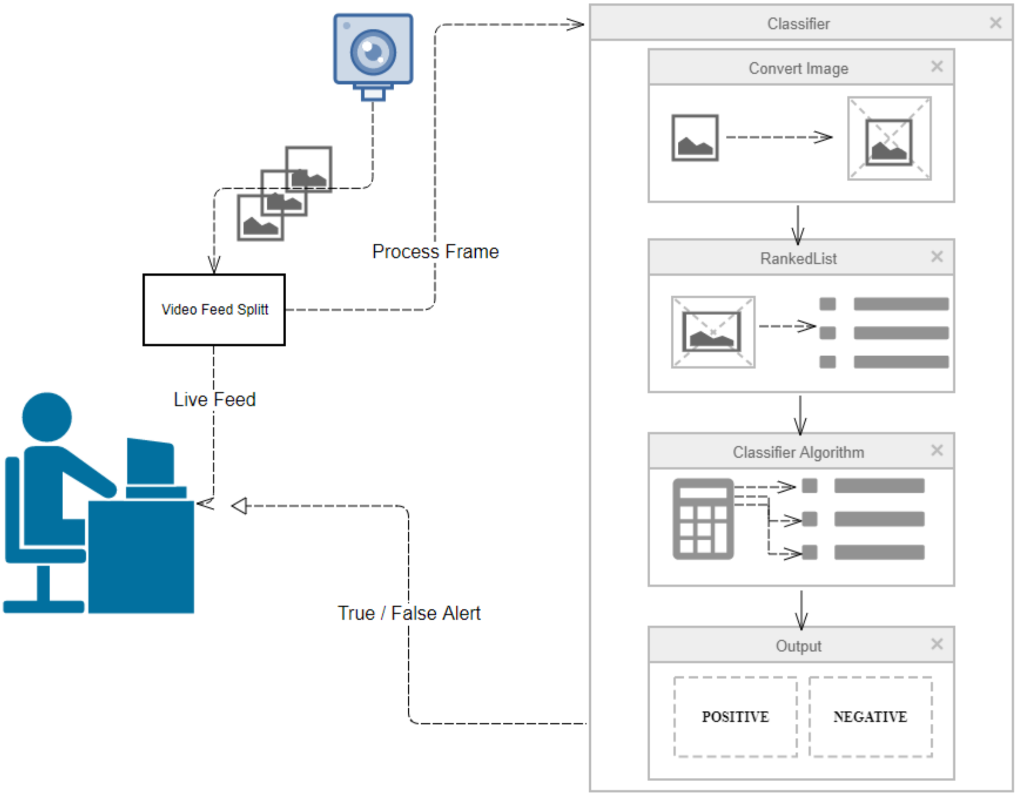
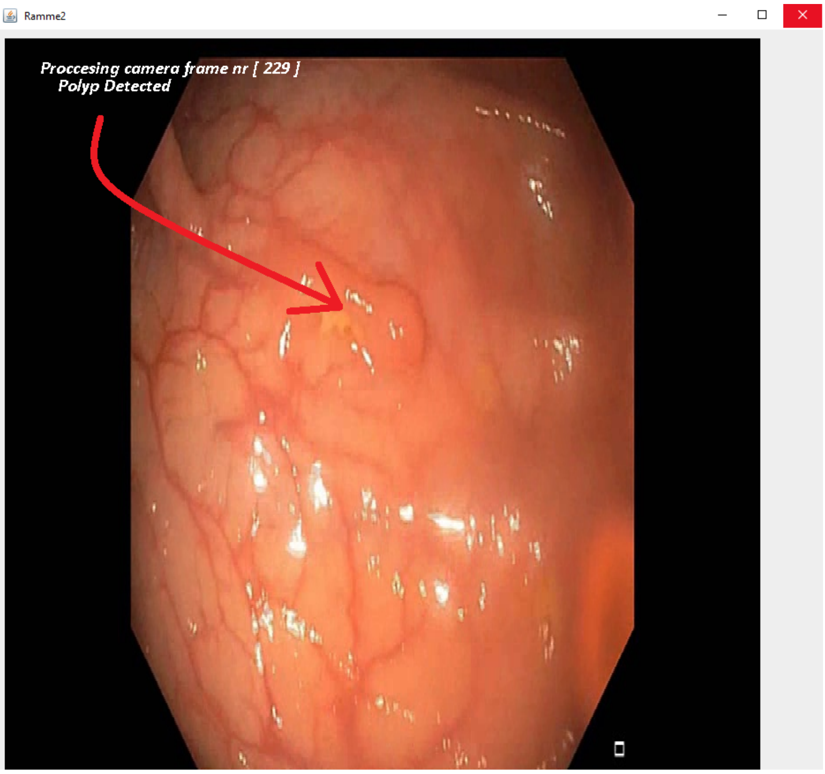


Figure 3.8: Main-Model of classification and our real-time detection system

One of the barrier we encountered as expected were low FPS, and these were caught up by skipping frames forwarded to the classifier from the video-feed. A polyp is visible in several frames and will in most cases be detected in the majority of these frames, due to the slow moving camera which enabling us to actually skip frames and not affect the ”Polyp-Alert” rate. By skipping frames, the endoscopist can see the live video, and at the same time get a live feedback on the screen from the classifier. The pace at which we skip frames depends on the FPS we achieve without any frames skipped, the slower the classifier, the more frames are skipped. The overall real-time classification pipeline ends with the classification feedback (positive or negative) is merged with the live video and visualizes to the endoscopist as shown on the figure 3.9.



The GUI which the endoscopist reviews provide a real-time feedback from the classifier.

Figure 3.9: Real-time Classification GUI

3.7 Summary

In this chapter, we discussed the LIRE platform with the use global and local features such as JCD, CEDD and Tamura. These three features exhibit intresing results in according to table 3.1 in Rieglers research [48]. We have decided to proceed with these features in our classification due to their overall score and possibility of their combination later on. We present the pipeline and architecture of our model and the parts which are the backbone of this real-time system. We give a graphic and description of how these feature values are extracted and indexed for further use, how the search method is used in order to create the ranked list, which is further used by the algorithms for classification purposes and the main system as a whole.

We have also discussed deep learning models such as Inception v3. Inception-v3 is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012. This is a standard task in computer vision, where models try to classify entire images into 1000 classes. In will be interesting to see if we can use inception v3 model to improve our classification on computers with not so high computational power.

Open source java library LIRE provides a simple way to retrieve images and their characteristics. It creates image indexes based on selected global or local features. Since EIR is build on LIRE and achieves a state-of-the-art performance, we will also build our real-time classification system around this open source library. Further we discussed our search-based algorithms, our system pipeline and model creation. In computer science, machine learning has recently become a popular topic, and we will see if we can utilize some of its algorithms such as kNN, ada boost or neural networks for improving classificiton of polyps.

Chapter 4

Iterative Development, Experiments and Results

4.1 Medical data

Classification and searching in large scale data-set requires a lot of resources and time. To achieve a system optimized to different approaches, including neural networks, it is important to have data set of a larger scale. Although, our approach does not require the same huge amount of images as a neural network classifier would. One of the challenges researchers faces are the lack of quantity in data-set to train and test the system, which we in our case did not. There are various reasons for this, it may be due to different hospitals use different type of technology to collect data, and it may be in different formats, sizes and resolutions. There are also privacy laws protecting collected patient data and its use for medical research purposes to also have in mind.

A large scale data-set for medical purposes is defined as if the amount of storage the complete data-set exceeds normal computer memory storage. A normal computer is of 128 GB of memory, and if we were to continue our 4K resolution experiment of the first system iteration, we would definitely require more space than we are currently in possession of; considering we are using a 500GB external hard-drive and a 1TB internal one. If we consider images on large image hosting sites like Flicker as a reference so the images can be of size of 3-33 MB and they are of up to 2048p. A data-set on large scale will then be between 4,000 to 110,000 photos to extract reasonable results.

We used ASU Mayo Clinic polyp data-set [60] to both for training and testing our system. This is the only available large scale annotated data-set for our GI tract use case. The data-set includes 38 videos with 36476 Images in total. The images are also of different pixels resolution 712 x 480, 856 x 480 and 1920 x 1080. Images are between 46 KB -1002 KB of size. The data is sorted into two main folders T1 and T2 [table 4.8.]. Both folders have two sub folders each Pos and Neg. The images have been sorted to positive and negative sub-folders respectively, this means we can be sure about a specific image containing a polyp or not. The tables 4.1 and 4.2 shows more details about the content of the ASU Mayo Clinic data-set. During iteration III, which is described in section 4.7, we use only with-polyps (wp) videos from table 4.1. We further divided these wp videos into small chunks for cross-validation. These chunks are visualized in figure 4.1. During our classification improvement described under section 4.8, for detection improvement, we used entire ASU Mayo data-set with 36476 images.

Table 4.1: Data-set T1 folder in details

Videoname	Neg	Pos	Resolution
NP_5	682	0	712 x 480
NP_6	838	0	712 x 480
NP_7	769	0	712 x 480
NP_8	712	0	712 x 480
NP_9	1843	0	712 x 480
NP_10	1925	0	712 x 480
NP_11	1550	0	712 x 480
NP_12	1740	0	712 x 480
NP_13	1802	0	712 x 480
NP_14	1639	0	712 x 480
WP_2	79	245	1920 x 1080
WP_4	0	910	1920 x 1080
WP_24	145	374	1920 x 1080
WP_49	110	391	856 x 480
WP_52	422	684	856 x 480
WP_61	130	209	1920 x 1080
WP_66	184	234	856 x 480
WP_68	70	189	1920 x 1080
WP_69	381	235	1920 x 1080
WP_70	25	385	856 x 480
Total	15046	3856	

Table 4.2: Data-set T2 folder in details

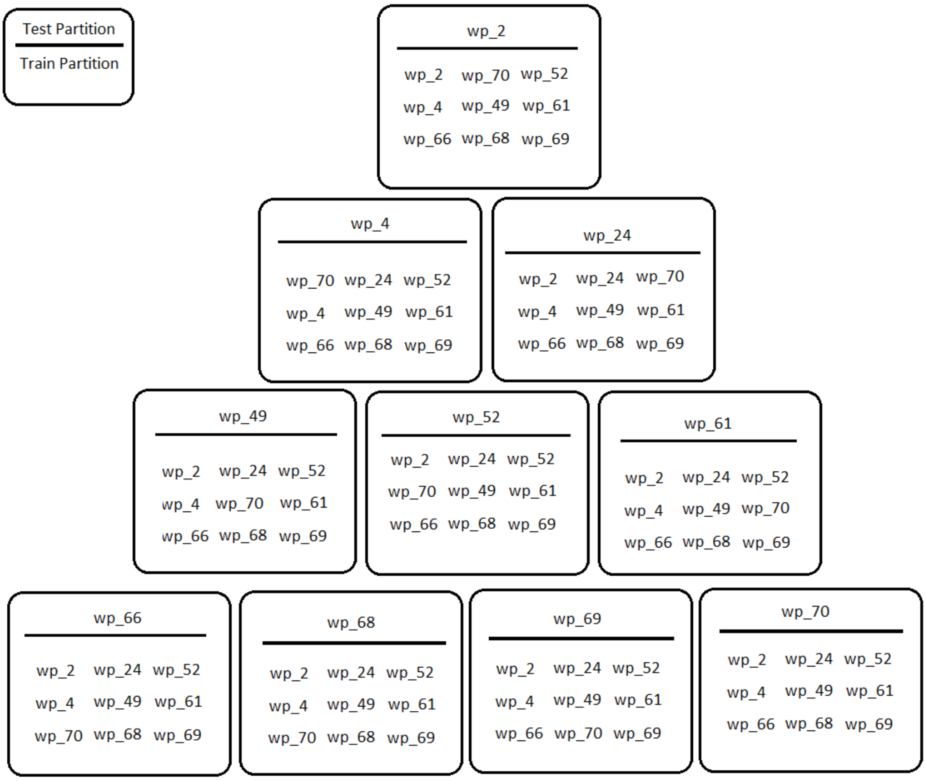
Videoname	Neg	Pos	Resolution
VD1	599	0	712 x 480
VD2	625	0	712 x 480
VD3	628	0	712 x 480
VD4	607	0	856 x 480
VD5	225	693	856 x 480
VD6	0	1218	856 x 480
VD7	110	445	712 x 480
VD8	111	335	856 x 480
VD9	106	290	1920 x 1080
VD10	1257	548	856 x 480
VD11	114	338	1920 x 1080
VD12	0	134	1920 x 1080
VD13	0	312	1920 x 1080
VD14	1815	0	712 x 480
VD15	1795	0	712 x 480
VD16	1627	0	712 x 480
VD17	1807	0	712 x 480
VD18	1835	0	712 x 480
Total	13261	4313	

4.1.1 Data-set

As shown on figure 4.1, we decided to use leave-one-out cross-validation to evaluate detection accuracy. The advantage of the leave-one-out cross validation is that we can check the robustness and generalization potential of the model in term of productivity. We left out one non-overlapping video for testing and rest of the data-set for training our model. These test and training videos were rotated and until all videos has been used for testing exactly once. Different features were used to extract information from images.

4.2 Machine Setup

During this thesis, we used two different machines while making progress through our iterations. Table 4.3 shows a more detailed list of hardware and software specifications used under this project. One of the main purpose of this project was to be able to run a real-time polyp detection, which consists of the entire pipeline illustrated in 3.8 on hardware with less computational resources, although with easier and much more mobile



Each box in this figure represents one split for Leave one-out-Cross-validation. Test video name is written above the line in the box which is then cross validated with videos below the line.

Figure 4.1: Leave one-out-cross-validation splits

deployment in hospitals than machine setups with GPU. We used computer 2 for iteration I and our classification improvement under section 4.8, however iteration II and III was completely executed on computer 1.

Table 4.3: List of HW and SW utilized during this research

Hardware and software used in this thesis. We utilized Open source projects like OpenCV and LIRE.

	Computer 1	Computer 2
Name	Intel nuc5i3ryh	macbook pro 15” mid 2012
Processor	Intel core i3-5010u (3M Cache, 2.10 Ghz)	Core i7 (I7-3615QM) 2.3 - 3.3 GHz
Harddisk	200 GB	256GB
Installed Memory (RAM)	8 GB	8GB
Gpu	None	Nvidia gt 650m 512mb
Operating system	Windows 10 Pro	macOS Sierra 10.10.5
OpenCV	2.4.11	2.4.11
Lire Lucene	1.0b2	
Java	Version 8 Update 101	1.8.0_121
Python	3.5.2	2.7.10
Cuda	None	8.0.83
Tensorflow	None	1.0.1

4.3 Evaluation Method and Metrics

In this research, we use leave-one-out cross validation for iteration III to determine and produce our results and also see how the system perform on an independent data-set. This results in having the system use 90% of the data-set on training and 10% on validation, where 10% in this case is one video. However, we will not guarantee that there will not be different results with different partitioning of the data-set, with regards to cross validation.

Because of the unbalanced data-set that was available for us, we had some difficulties with implementing a routine for evaluating classification, performance in therms of FPS, algorithm - and total frame time for the real time detection system. The 3 metrics we used were Precision, recall/sensitivity and F1 score. These are globally accepted and used metrics. Unbalanced data-set and some blurry images impacts the performance of the polyp classifier. Precision, F1-score and recall are the classifier characteristics that is used to determine our classifier and algorithm performance.

In pattern recognition, high positive precision means more favourable results were returned compared to any unfavourable ones by an algorithm. Positive precision calculation gives us an idea of how many true positives that was returned by the algorithm, with regards to the total number of positive recalls. This means that positive precision percentage represents the confidence level of the positive recalls and will for each positive recall represent the probability of the recall to be a true positive. The same calculation goes for the negative precision:

$$Positive\, precision = \frac{TP}{TP + FP}$$
$$Negative\, precision = \frac{TN}{TN + FN}$$

Recall is also called sensitivity and shows the fragmentation of all true positive instances that also are classified as positive. High positive recall, low FN, means that mostly all of the positive samples was classified as positive samples and thus captures by the algorithm. The same calculation goes for the negative recall:

$$Positive\, recall = \frac{TP}{TP + FN}$$
$$Negative\, recall = \frac{TN}{TN + FP}$$

In the best case, both a high recall and a high precision are desirable, although mostly high precision often cause low recall rate and vice verse. The F1 score is the harmonic mean between precision and recall and is to measure the quality of the classification system which consider both recall and precision:

$$F1 = 2\frac{Precision * Recall}{Precision + Recall}$$

4.4 System Development Iterations

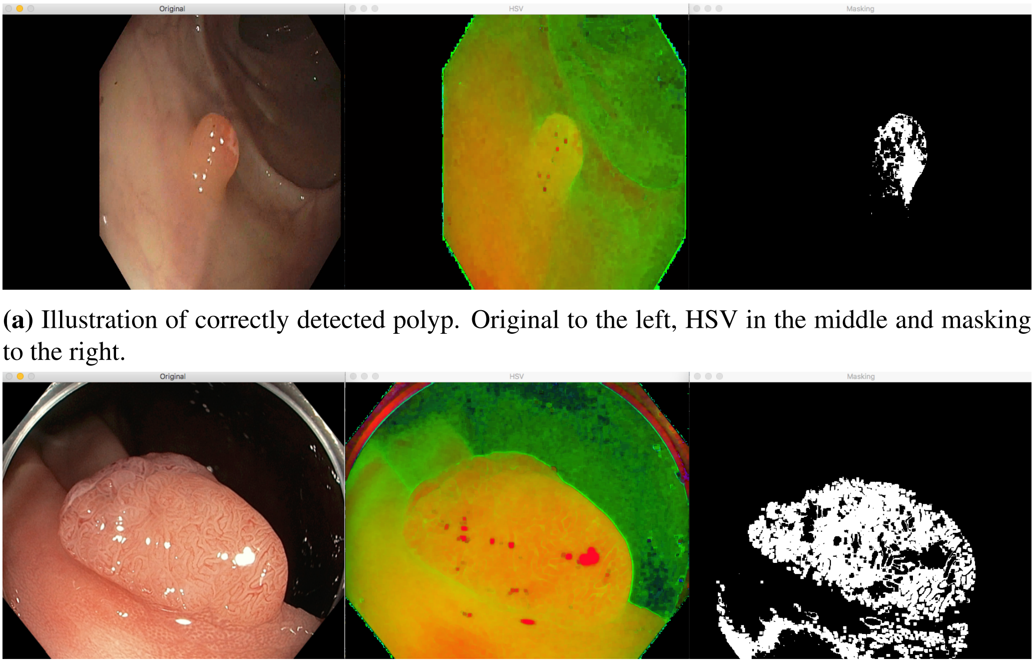
In this section, we will focus on presenting our development journey divided into four main parts, describing the phases for each iteration of our detection system. This will give the reader a deeper dive into our way of tackling this challenge, the model creation and a more detailed look at both architecture and pipeline. Each iteration section concludes with a brief explanation of the complete system iteration. Developing our system required plenty of careful planning on how we intent to make this a deploy-able system in hospitals. Not only with respect to the medical professionals use of the system, but also for the sake of optimizing and improving the initial advantages that comes with such a system. Preparing the data from its RAW video format to something more convenient is also something we had to take into consideration. By extracting each frame from the a single video and then separate each of the frames into different categories, depending on what the frame contains, we are (table of videos reference) able to obtain the data set. These frames could include polyp (positives), only colon tract (negatives) and other visual occurrences such as bleeding or inflammatory tissue. How we plan to create the system architecture is going to impact how the live stream performs, since when the procedure is live each frame that the system is receiving has to be processed immediately. In short, that means that the feature used to extract information from the training data also has to be used to extract values from the incoming frame on the fly. In our case, this is mandatory because the images that are indexed are in the domain of those features used. Therefore, to calculate the distance and get a realistic representation for similarity, the incoming frame also has to be in the same feature space.

4.5 Iteration I: Real-Time Color Detection

The system development process began by experimenting with the power of OpenCV and the LIRE platform to get a feel of how we could utilize these libraries to build our system. Initially, we implemented a system that detected different colors on the fly with OpenCV. The color detection was specifically based on the RGB color spectrum, where HUE values was used to specify saturation and brightness of the chosen color.

4.5.1 Model Creation

The detection of colors is not object based detection since color is not distinct to any object and can therefore change based on the exposure of brightness and the angle viewed from. This changes the HUE of the chosen color and will not be detected with such a detection system. In figure 4.2a, we illustrate that in some minor cases we are able to detect polyps with correct specified lower and upper Hue Saturation Value (HSV) parameters. On the other hand, we see a clear demonstration in figure 4.2b of what happens in cases where the camera has gotten too close to the polyp and have brighten up the area around the polyp as well. This phenomena in this iteration of the system is what makes polyp detection, or object detection in general not possible.



(a) Illustration of correctly detected polyp. Original to the left, HSV in the middle and masking to the right.

(b) Color detection did not separate between the polyp and the Mocusa. Illustration is in the same manner as in figure above 4.2a.

Comparison between an instance where the polyp was detected correctly using color detection 4.2a, and another where the surrounding areas of the polyps also was part of the polyp detection 4.2b.

Figure 4.2: Instance comparison of color detection

With machine learning, the computer learns what to look for, by being provided with negative - and positive images. The idea is like subtracting a negative image from a positive image to get the difference between the images. This way the computer learns what the object in the positive image looks like, although this is based on hundreds of images, and especially which features are selected to describe the images and the objects in them.

The process of implementing a our real-time system on a smaller scale was what got us to realize the complexity and level of difficulty of developing a real-time system. Especially, with regards to which factors that affect the performance of the system when it comes down to delay, how fast each frame is processed and how many FPS contributes to a smooth experience. Combining these elements from a real time detection system to classification and the difficulties form this field, elevates this task to a higher caliber.

4.5.2 Architecture and Pipeline

During this iteration, we created a python program where we initialize lower and upper boundary in HSV values for which color we are trying to detect. For this, we utilized OpenCV's image processing libraries, and in the first step, start by converting the image from RGB to HSV color-space. Secondly, we erode the image so we get a separation between the object edges that appear in the image. This is done by using a structuring element that scans over the image and compute a maximal pixel value for areas that overlap with the structuring element, and replaces that pixel with that maximal pixel value. High values are conveyed as bright areas and the opposite for dark areas. Therefore, we will have eroded around objects in the image.

Dilation, which is closely related to erosion, but is basically the opposite operation of erosion, is done on the eroded image where the dilation process feeds border to an object. This is done to reconstruct parts of the image. We then get a more noticeable and distinct separation between objects that initially was hardly separated. Lastly, we perform a masking of the image to view the selected parts of the image based on the HSV values given as parameters.

4.5.3 System Experiments and Evaluation

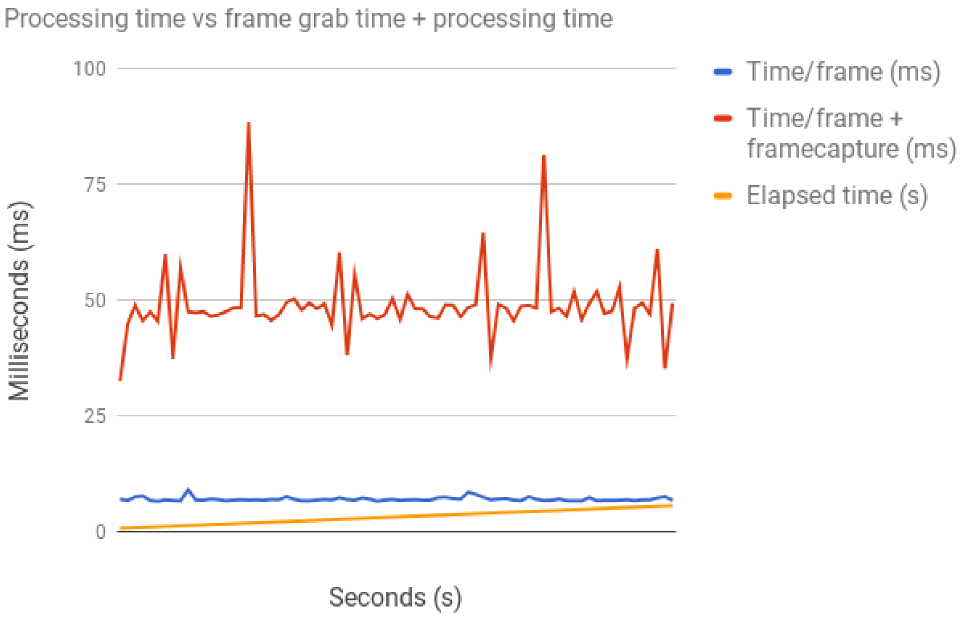
In table 4.4, we have measured the average time it takes to perform the image processing on the received frame from the camera in the second column. In the third column we measured the elapsed time for just grabbing the frame. This experiment reveals how much time it actually for the OpenCV method to grab the frame from the camera and process it, which compared to processing time for the frame itself alone is significantly higher. Figure 4.3 illustrates the differences between the two.

Table 4.4: Measured processing time table

Frame nr:	Time/frame (ms)	Time/frame + framecapture (ms)	Elapsed time (s)
1	7.012844086	32.48596191	0.703795
2	6.696939468	44.70705986	0.768378
3	7.49206543	48.9218235	0.834997
⋮	⋮	⋮	⋮
74	6.804943085	49.29590225	5.590582
AVG	6.984565709	48.88433379	

4.6 Iteration II: Prototype of Real-Time Object Detection

In the next system iteration, we had to completely change our approach to accomplish real-time object detection, by first and foremost implement a system that actually does classification. Not only where we faced with a real-time classification problem, but also with the fact that we have to make do with what little resources is available at a hospital in terms of GPU, CPU and RAM (as described in section 4.3). This is where we used the open source LIRE library which focuses on providing a easy way to retrieve images and photos based on color and texture characteristics. In our case, we designed and implemented a complete system for image classification to begin with. We implemented the system with segments of code for evaluation purposes, which can be seen in section 4.6.1.



Graph showcasing processing time per frame and also the time it takes to capture frame from camera

Figure 4.3: Frame capture and processing time

4.6.1 Model Creation

To create this model, we decided to use LIRE open source project for our classification purposes. This means that we used LIRE's built-in method for searching through images with GenericFastImageSearcher() along with LIRE's different features for extraction.

```
ImageSearcher imgSearcher = new GenericFastImageSearcher(2, JCD.class);
```

The features we choose to use for feature extraction are Tamura, JCD and CEDD, which are the best performing features according to [49], which is also shown in table 3.1. Since the system is meant to achieve real-time classification, we also continued using the OpenCV library for the image retrieval from a camera. The model that we have created consist of four main phases:

Phase I: Indexing and Feature Extraction

The initial phase handles the extraction of features of the data we use for training. The indexing program we developed, reads images from a folder and extract features based on which feature that has been selected beforehand. This folder is comprised of 15 images, where 6 images are positive and 9 images are negative. For our prototype purposes this is sufficient to determine if our classifier does indeed classify images. The data-set is viewable in table 4.5. Our indexing program creates an index folder that consist of the meta-data for each image, with respect to the feature. Since the index folder is much smaller in size compared to all the images, we also get faster performance when our system searches through the meta-data. When all images has been indexed, the preparation process of the data-set is done and is ready to be searched.

Table 4.5: Red Folder data-set

	Pos	Neg
Test	11	7
Train	6	9

Phase II: Frame Capture and Ranked List

The second phase of the program was creating the module that enables a camera and grabs whatever frame the camera is currently viewing and converting this to a type of BufferedImage, which is essentially an image with an accessible data buffer. It is therefore more efficient to work directly with BufferedImage. This was essential due to the fact that we needed to have a incoming frame which we could compare to the images in the index folder. Based on the selected feature, our Searcher.java creates a ranked list in a hash-map which is then used in our classification algorithm. The ranked list is score based, which further is defined as the distance to the incoming frame. This means that a image in the index folder that has high similarity with the incoming frame will have a higher ranking in the list, and thus better score.

Phase III : Classification Algorithm

The third phase, also called the classification phase, is where we have made our algorithm that handles the calculations and is the basis for the classification process. We have written the code shown in listing 4.1, that reads the ranked list made by the searcher and splits the elements of the list into two groups, one for negative images and one for positive images (this is done by reading the file-name). The indexed image with lowest score is the image with the best match to the incoming frame, and will have a high ranking. The algorithm sums the score of negative indexed images and positive indexed images and calculates the average. If the average for negative images are lower than the average for positive images, the classifier will classify the frame as a negative frame. In some cases, the ranked list will, for instance, contain of only positive indexed images. Then, we know for certain that the frame is positive, since the frame had best match with only positive indexed images. Although, the average for negative images will equal zero, and thus be lower than the average for positive indexed images. In this case, we have to be aware and implement a workaround for this scenario and still classifies correctly.

In the second algorithm as shown in listing 4.2, we created a similar algorithm to the averaging technique, but included a linearly increasing bias factor which has the purpose of spreading the scores from each other, like zooming in on a data point graph, as illustrated in figure 4.4. The idea is based on the fact that the classifier in theory would easier separate one class form another by having the data samples from their respectively class closer to each other and achieve better classification results. The classification performance is discussed under the evaluation section below.

Listing 4.1: AvgMethod

```
String p = "p";
for (Object o : hmap.entrySet()) {
  Map.Entry pair = (Map.Entry) o;
  String polarity = String.valueOf(pair.getKey());
  polarity = polarity.substring(polarity.lastIndexOf("\\") + 1, polarity.lastIndexOf(" "));
  if (polarity.contains(p)) {
    i++;
    tempP = ((Double) pair.getValue());
    posImg += tempP;
  } else {
    j++;
    tempN = ((Double) pair.getValue());
    negImg += tempN;
  }
}
posAvg = Math.abs(posImg / i);
negAvg = Math.abs(negImg / j);

if (posAvg == 0) {
    hmap.clear();
    return false;
 }
else if ( negAvg == 0 || posAvg <= negAvg ) {
    hmap.clear();
    return true;    }
else {
    hmap.clear();
    return false;   }
```

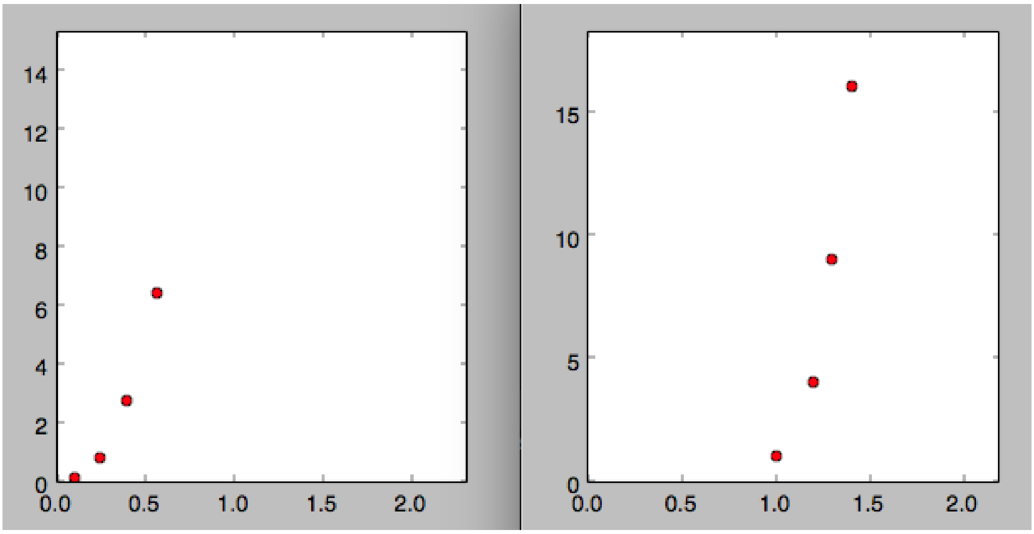


Illustration of how it would be easier to separate data points. On the plot to the left, we see how the data points are harder to separate due the distance between them. However, on the plot to the right we have used a bias factor to separate them for easier classification. X axis being feature x1 and y axis being feature x2.

Figure 4.4: Separating data points

Listing 4.2: LinearSpreadMethod

```
String p = "p";
double i = 0.1;

for (Object o : hmap.entrySet()) {
  Map.Entry pair = (Map.Entry) o;
  String polarity = String.valueOf(pair.getKey());
  polarity = polarity.substring(polarity.lastIndexOf("\\") + 1, polarity.lastIndexOf(" "));
  if (polarity.contains(p)) {
    tellerP++;
    tempP = ((Double) pair.getValue()) * i;
    posImg += tempP;
    i = i + 0.1;
  } else {
    tellerN++;
    tempN = ((Double) pair.getValue()) * i;
    negImg += tempN;
    i = i + 0.1;
  }}
posAvg = Math.abs(posImg / tellerP);
negAvg = Math.abs(negImg / tellerN);
if (posAvg == 0) {
  hmap.clear();
  return false;
} else if (posAvg <= negAvg || negAvg == 0) {
  hmap.clear();
  return true;
} else {
  hmap.clear();
  return false;
  }
```

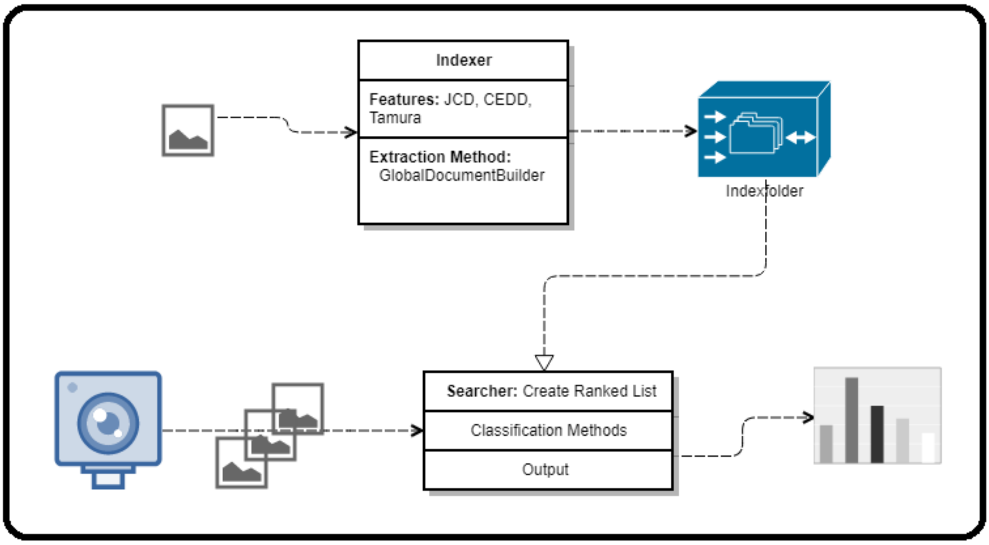
Phase IV : Output and Alert

As the system has completed classification of a frame, there is produced a output which alerts the endoscopist that our system has detected a polyp in the frame. As of this iteration, the output is in the form of text on the screen that the endoscopist is viewing as well as in the terminal, although for development purposes. What we assume would be a better alternative, might include sound or vibration, however, the alert is somewhat preference based and should be discussed with doctors to find a common ground. However, this alternative is left as future work in this thesis.

4.6.2 Architecture and Pipeline

In this section, we will describe architecture and the pipeline of Iteration II. The index folder gets created by our Indexer.java as shown in figure 4.5. The meta-data for each image in the train data-set gets created with GlobalDocumentbuilder(), and stored inside indexfolder. This is described in depth in section 4.6.1.

This newly created index-folder containing meta-data files are used in our classification to classify camera frames. In our classification segments as described in the section 4.6.1, the camera grabs a frame, and we convert them one by one into decoded buffered-frames. This decoded frame is then sent to the searcher where it is compared with indexed images and creates a ranked list with GenericFastImageSearcher(). This ranked list is then used in our classification methods, linearSpreadMethod and AvgMethod, to classify incoming frame. This flow through the pipeline occurs for each frame that the camera grabs. The output from these method are used to generate our diagrams.


Indexer
Features: JCD, CEDD, Tamura
Extraction Method: GlobalDocumentBuilder
Indexfolder
Searcher: Create Ranked List
Classification Methods
Output

The figure illustrates the architecture and pipeline of the red-folder.

Figure 4.5: Red-folder architecture and pipeline

4.6.3 System Experiments and Evaluation

In this section, we will run experiments on FPS, classification score and time consumption and discuss the figures presented along with their characteristics.

FPS Performance

During this prototype iteration, this experiment determines the effects of resolution, features and applied algorithm on the FPS. The FPS experiment in figure 4.6 reveals how much of an impact resolution has on the system. We can clearly observe that the 4K images (2976x3968) achieves a performance of 2 FPS with no difference among the features and algorithms. We suspect that this is due to the low initial FPS. However, looking at the downscaled images (744x992), we achieve an increased performance of minimum 200% for Tamura, and even higher with CEDD and JCD in combination with the averaging algorithm, confirming what we suspected about the 4K images. The index folder has not undergone any changes and the images there still remain in their native resolution. Although, under this experiment we therefore also down-scaled the training images and made a new index folder, down-scaledIndex. This was done to see if we get any performance increases by comparing down-scaled images to down-scaled indexed images. We got lower performance with down-scaled index and down-scaled images because of the lower quality in the images.

As of the results presented, the system does not reach the requirements of real-time and, as we already knew, we have to implement the skipping of frames in order to make the viewing experience stay live and achieve somewhat real-time polyp detection. It is crucial to mention that these are single threaded FPS performance, implying that optimized code with respect to threads available would score much better.

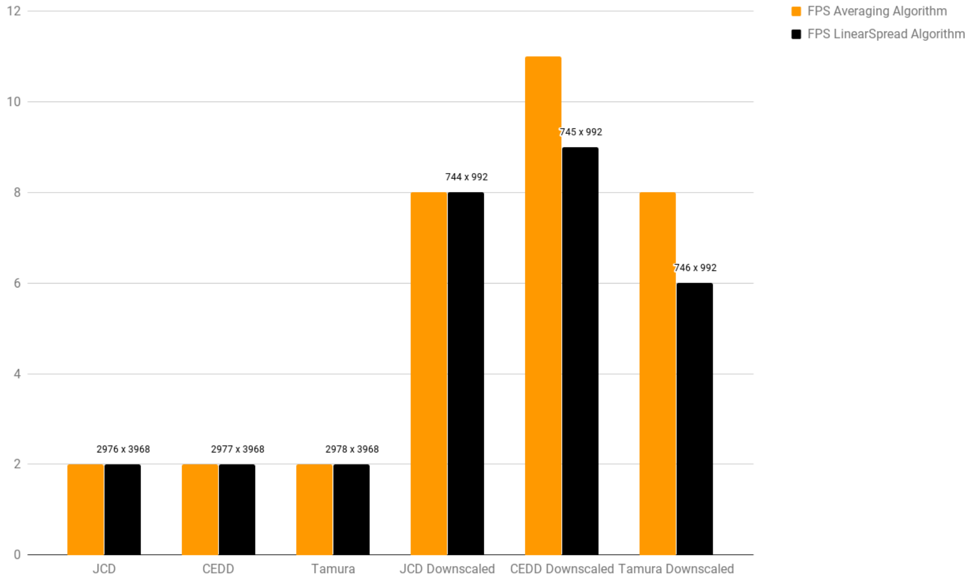
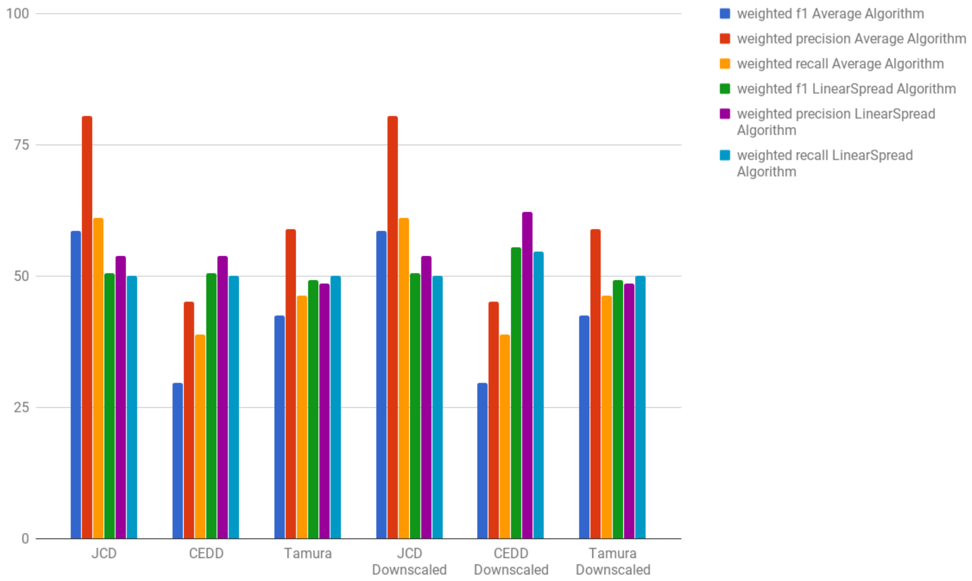


Figure 4.6: Graph shows how image resolution affects processing and increases FPS

Classification Score

In the figure 4.7, we have plotted the classification performance of our two algorithms, Averaging and LinearSpread. The three first column bars for each feature shows the weighted F1 score, the weighted precision score and the weighted recall score for the averaging algorithm. The next three columns shows the same characteristics for the other algorithm, LinearSpread. The last three features, are scores obtained while running the system with the down-scaled test images.

By examining the averaging algorithm in this figure, we notice that both CEDD and Tamura are outperformed by JCD, where their score hover around 30%-45%, JCD achieved a relatively higher score of around 60%. What we notice stands out for JCD, is the comparable higher weighted precision score, showing how much of what the classifier actually selected was relevant, which in this case was much higher.

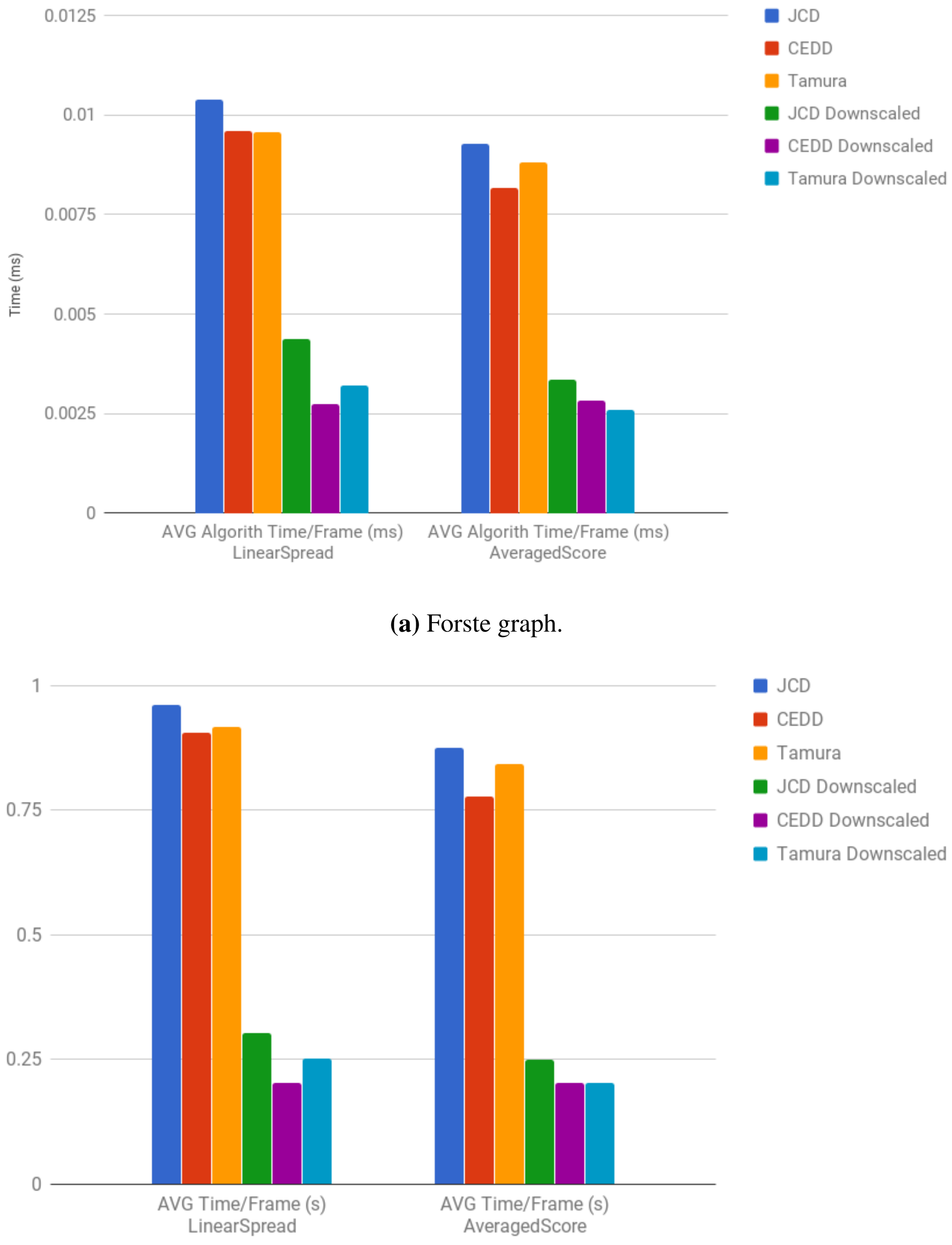


Performance of red folder classification with different methods. Graphs also shows native folder compared to down-scaled images.

Figure 4.7: Red-folder weighted score

Time Consumption

Figure 4.8 shows the elapsed time with the JCD, CEDD and Tamura features for both algorithms. In sub-figure 4.8b, we show the average elapsed time per frame, including processing of the frame itself. On the other hand, we in sub-figure 4.8a, show only how fast the algorithm executes. As seen in the legend, the three last histogram-bars are test-runs with down-scaled test images. From sub-figure 4.8a, we notice that the classification algorithm LinearSpread, is slightly slower than the Averaging algorithm. We suspect that this is due to the large amount of calculation required for each frame; This accounts for both down-scaled images and original. Comparing the time consumption of both the algorithms and the features respectively, we notice huge decreases in processing time between original images vs down-scaled ones, where down-scaled images is processed 200% faster. This applies to both sub-figures. The JCD feature, which is described in section 3.1.1 is more time-consuming compared to Tamura. However, this comes with a trade-off between time and detection.



(a) Forste graph.

(b) Comparison of elapsed processing - and algorithm time between resolutions in seconds (s)

Figure (a) shows average time in millisecond (ms) per frame for linaespread and averaged algorithms. Figure (b) shows total avg processing time per frame in seconds (s) for both linaespread and averaged algorithms.

Figure 4.8: Algorithm avg time

4.6.4 Iteration II: Summary

During this iteration, we developed a fully working system that indexes the training-set with different features which is then used for classification of the read frame in the validation-set. The classifier is also using our implemented algorithms. The pipeline, as seen in figure 4.5, has been constructed in such a way that the user can either choose to classify an already recorded video or use the camera to classify the received frames. This is valuable in cases where the wireless capsule has been used to record 5-8 hours video and having the opportunity to choose to classify a video would take a lot of weight of the health care system by not examining this video manually. The evaluation extension for our system has been tailored in order to measure the performance and further development. System also provided output on the terminal with evaluation of the frames and at the same time visualize the currently processed frame in the GUI.

Therefore, we conclude this system iteration with the remarks that several aspects that needs improvement, such as classification, processing time in terms of FPS and GUI-development, which was revealed through our experimentation. This prototype's original goal was to design and implement a system that does classification from images retrieved from a enabled camera.

4.7 Iteration III: Real-Time Polyp Detection

During this iteration, we will improve our classification algorithms along with FPS and fast indexing with threading. We will also implement our initial idea of making the system real-time for the endocopist where the system performance output is expected to be lower than 25-30 FPS.

4.7.1 Model Creation

The model presented under this section, is an extension of our previous implementation and will also include our real-time work-around idea concerning frame skipping. As shown in figure 3.8, this system will now split the video feed; one which the doctor can view and the other which will go through our classification pipeline. After classification, the output will be displayed on the video feed that the doctor reviews. What this workaround results in, is that we no longer repaint the processed frame after classification. In actuality, the endoscopist will view the video at the same speed the camera capture frames and the classifier will output the calculated result on the display the expert examines, although with slight delay. The delay will with certainty be measurable, but not as noticeable. For the doctors perception, the performance will feel real-time since the delay mentioned is around 200ms and is a fraction of a second. This is what we can achieve with the frame skipping implementation and fully utilize our low computational resources. The ratio at which we skip frames depends on which classifier algorithm we use and how fast this algorithm performs, the resolution of the frames captured by the camera and the computational power of the machine used, in this case, computer 1 shown in 4.3.

4.7.2 Architecture and Pipeline

As seen in figure 3.6, we have in this iteration implemented a threaded indexer based on how many cores and threads the computer has available. As a result, the indexing process of roughly 5402 images takes much lesser time and performs that much faster than the single threaded build from previous iteration. In this case, the total indexing time went down from several hours to approximately 45 minutes. Keeping in mind that with leave-one-out cross validation, the total amount of images indexed vary with respect to which video we leave out to validate with.

Considering the system in its entirety, the system has undergone some changes under the hood:

- Replacement of the classification algorithms and implementation of another, discussed in section 4.7.3.
- The indexer, as mentioned introductory, is now multi-threaded and performs exponentially faster.
- The data-set will be the one introduced in section 4.1.1.

- Added a GUI which presents the frames received from the camera and the detection results from our classifier.
- The implementation of the frame skipper.

4.7.3 Algorithm

During this iteration we implemented a new classification algorithm to improve detection rate. In previous iteration, our algorithms could accomplish a detection-rate of around 50% F1 score. The main purpose of this new implementation is to improve classification and elevate the polyp detection reliability.

In listing 4.3, we show the implemented algorithm which in mathematical representation is:

$$\frac{\partial}{\partial x}\ln(x) = \frac{1}{x}$$

For simplification this algorithm does not calculate the derivative, it rather calculates $\frac{1}{x}$, where x is the score associated to each indexed image respectively. An example of the calculation is shown in table 4.6 below:

Pay close attention to the rank and the score, since both are essential to why this algorithm operates as it does. As the rank decreases and score increases, the value of the function also increases since 1 divided over a higher number is less. Therefore, lower score, meaning higher relevancy, produces a higher function value. The summation of the positive function values is then compared to the summation of the negative function values and class with the highest value gets classified. In table 4.7, the function values for the two classes has been calculated and the frame is classified as negative, which in this case is right and desirable.

The classifications results are discussed under the section 4.7.4. As shown on figure 4.9, 4.11 and 4.12, we can see improvement in our new DivisionOverScore-algorithm by 20-25% compared to iteration II.

Table 4.6: Algorithm calculation for classification of a single frame

Rank	Score	Img_Name	Polarity	Formula (1/x)
1	55.37771911	IMG_20170710_124244_x (6).jpg	N	0.01805780404
2	59.02390787	IMG_20170710_124154_q (6).jpg	P	0.01694228722
3	67.78100912	IMG_20170710_124244_x (7).jpg	N	0.01475339499
4	67.96103291	IMG_20170710_124244_x (3).jpg	N	0.01471431432
5	68.72123749	IMG_20170710_124154_q (5).jpg	P	0.01455154238
6	70.59878936	IMG_20170710_124244_x (2).jpg	N	0.01416454884
7	71.7394095	MG_20170710_124154_q (4).jpg	P	0.01393933972
8	73.75523455	IMG_20170710_124154_q (3).jpg	P	0.0135583597
9	73.77115124	MG_20170710_124244_x (1).jpg	N	0.01355543438
10	75.08381263	IMG_20170710_124244_x (9).jpg	N	0.01331844994
11	76.42919824	IMG_20170710_124154_q (2).jpg	P	0.01308400484
12	80.46668367	IMG_20170710_124244_x (4).jpg	N	0.01242750359
13	80.6315741	IMG_20170710_124244_x (8).jpg	N	0.01240208952
14	81.67972498	IMG_20170710_124154_q (1).jpg	P	0.01224294034
15	85.36579738	MG_20170710_124244_x (5).jpg	N	0.01171429344

Table 4.7: Image function summation

Summation of function values for positive - and negative images from table 4.6

	Pos	Neg
Sum	0.0843184742	0.1251078331

Listing 4.3: Division Over Score Method

```
String p = "p";
double i = 1.00;
for (Object o : hmap.entrySet()) {
  Map.Entry pair = (Map.Entry) o;
  String polarity = String.valueOf(pair.getKey());
  polarity = polarity.substring(polarity.lastIndexOf("\\") + 1, polarity.lastIndexOf(" "));

  if (polarity.contains(p)) {
    double tempP = 1/((Double)pair.getValue());
    posImg += tempP;
  } else {
    double tempN = 1 / ((Double) pair.getValue());
    negImg += tempN;
  }}
if (negImg == 0.0 && posImg != 0.0) {
  hmap.clear();
  return true;
} else if (posImg == 0.0 && negImg != 0.0) {
  hmap.clear();
  return false;
```

```
} else {
  if (negImg <= posImg) {
    hmap.clear();
    return true;
  } else if (posImg < negImg) {
    hmap.clear();
    return false;
  } else {
    return false;
  }
```

4.7.4 System Experiments and Evaluation

As in the previous system evaluation section of iteration II, under this section we evaluate the system in the same manner, and analyze to see if there are any improvements in FPS, classification performance characteristics or time consumption.

FPS Performance

This time around we are using a different data-set than in the previous iteration, consisting of videos with varying resolutions. In figure 4.9 we present the FPS performance, where the classification pipeline uses Division-Over-Score as the classification algorithm. As we also noticed in the previous iteration, resolution plays a major part and affects the performance the most. Wp70 is the first video in the figure, with a resolution of 856 x 400 and a total of 410 frames. For this video, we see that in comparison the faster CEDD feature acquires a maximum of 7 FPS, JCD at 6 FPS and 5 FPS for Tamura. This trend was something we observed from evaluating the previous build of the system, and is also present in this evaluation. CEDD is the feature that scores the fastest in terms of time consumption and has a direct impact on the FPS.

Despite that we see some variety across all videos, the figure clearly displays, not only how much resolution affects performance, but also the effects of the bigger data-set. In our previous build, searching through the training data to find similarities was much quicker, and was the result of the considerably smaller data-set. Comparing the FPS results from both iterations, we notice that this recent build does not reach a peak of 11 FPS, which is caused by the data-set.

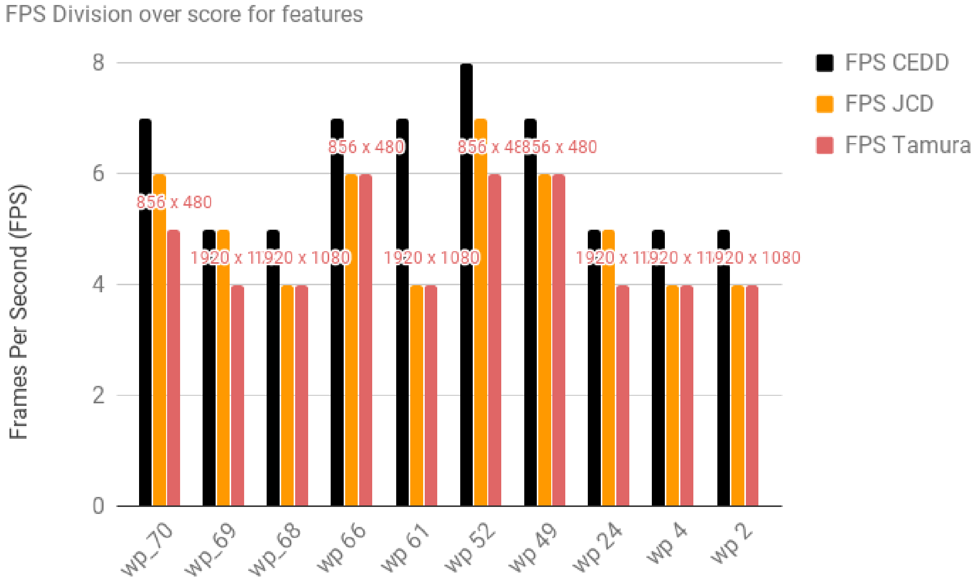


Figure shows frame per second (FPS) for different videos with different resolution from the data-set. FPS are different according to features selection.

Figure 4.9: FPS with different features

Classification Score

The classification characteristics have much improved since the previous build. Due to several videos and large data, we have split the presentation of scores into three histograms, where each graph show one of the three characteristics with three different features.

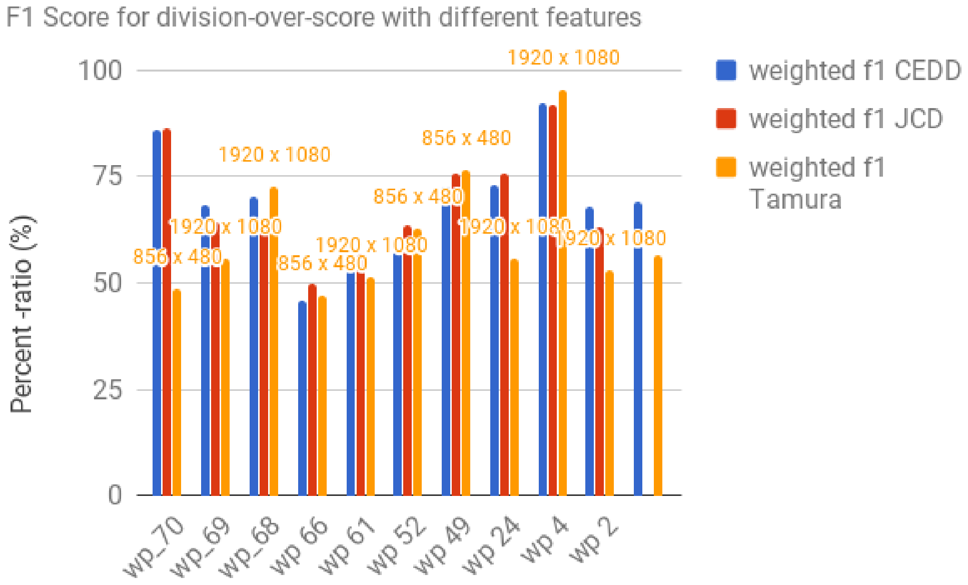


Figure 4.10: F1 score for iteration III

In figure 4.10, we notice a good amount of variety and on some videos JCD and CEDD scores much better than Tamura. However, there are cases where Tamura outperforms the two and it therefore remains to see which one achieves best classification result with regards to weighted precision and weighted recall. Although, as we observed in the FPS score 4.9, Tamura does not perform as fast as JCD and CEDD so it suggest that it might not be suitable for our real-time polyp detection. We also measure the algorithm time consumption and will conclude a clearer conclusion on that in the sections to come.

The figure 4.11 visualizes the weighted precision score of all videos. In the previous figure, F1 score 4.10, we notice that it is especially hard to classify polyps in video wp66, which is also present in the weighted precision figure. Precision represent the performance of the classifier in terms of the relevancy of the selected data samples. We suspect that the low precision score for wp66 in both figures, are due to the training-set not having images similar to the frames in this video, and thus classifies wrongly as a direct limitation of the training-set. On the other hand, wp4 has a precision score of 99.87% and the classifier performs near perfect. Although the high precision score for this video, wp4 only contain positive frames, meaning all frames has a polyp in them. Therefore, we have turned our attention how many positive and negative images are in the training-set. Which in our case means that increasing the number of negative images and balancing out the quantity between the two categories would potentially give improved classification. The differences between the features are not as visible is some cases. In this figure, CEDD and JCD usually pulls ahead and scores higher in almost all cases except a few, such as wp49 and wp66. This means that the weighted precision for these two features are more suitable for this scenario than Tamura. Although, we already addressed the limitation that comes with the training-set, and weighted precision might change for the better with a more balanced training-set.

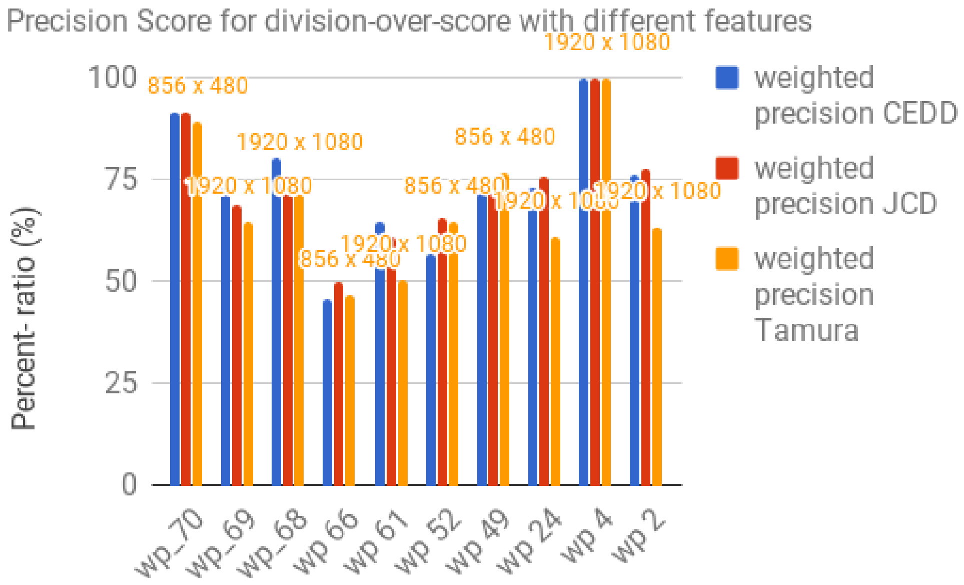


Figure 4.11: Precision for iteration III

The percentage in recall represents how many of the relevant items where selected. This characterizes how well the classifiers finds what is actually relevant, selects the datapoint and classifies it into the right category. Figure 4.12 show the weighted recall of all the videos with the three different features. We can already see improvements from the previous iteration with a general increase around 50% in some videos and features. With this algorithm, we achieve better weighted recall scores than the two previous algorithms and is much more suitable for classifying polyps. In terms of classification, the resolution of the video itself does not seem to play a major role, since we notice some videos achieve higher score than some 1080p videos. The performance of each feature vary and it would be hard to point out one that we would continue to use exclusively. Since this is meant to perform real-time, which feature we declare the best and most use-full depends on both the classification and last but not least the time consumption. In the next subsection we run experiments to characterize how each feature perform with regard to speed.

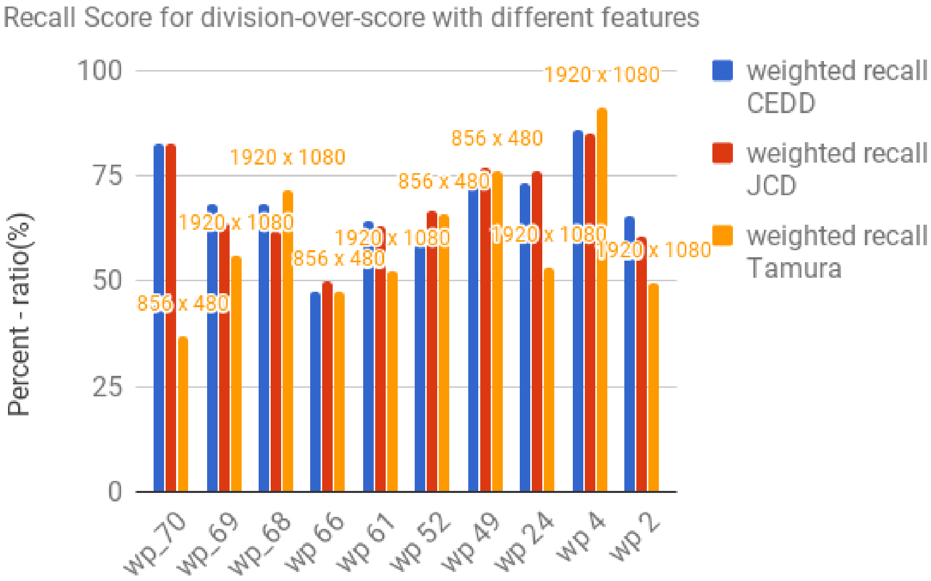
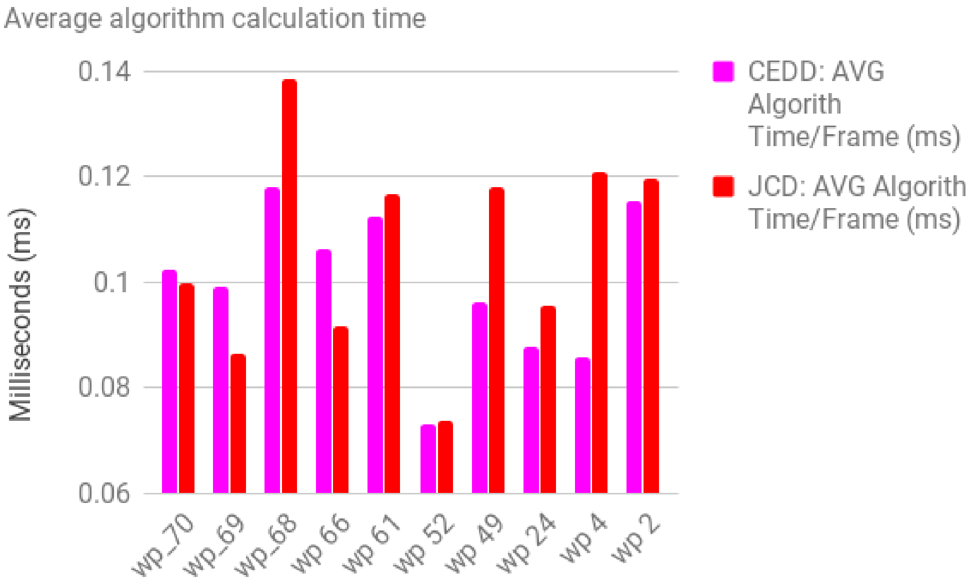


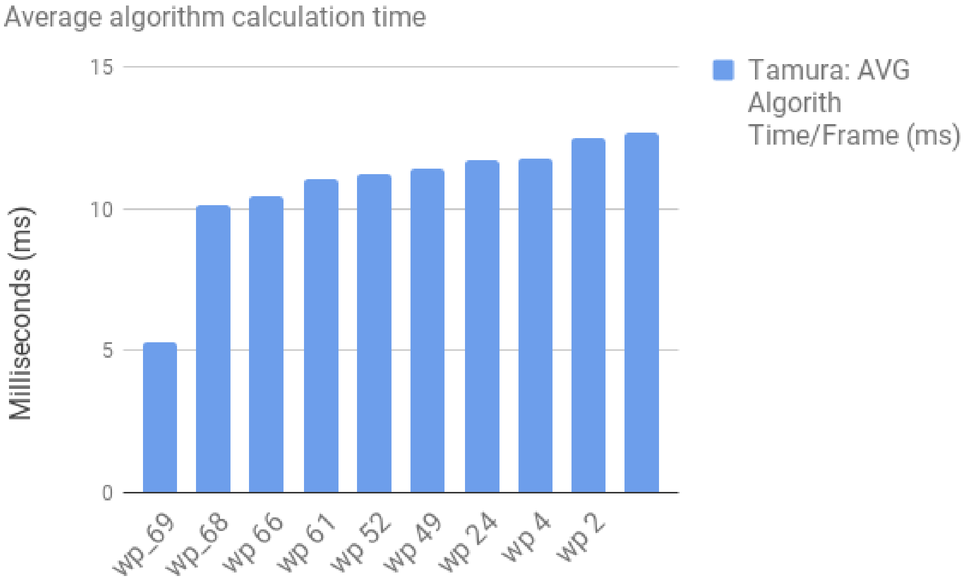
Figure 4.12: Recall for iteration III

Time Consumption

Aside from the polyp classification scores, we have measured the algorithm times with the different features and the gathered data is plotted in figure 4.13a. The reason for why the algorithm calculation time varies between the features, is due to the range of scores for each feature. Both JCD and CEDD have a lowest score of 100.0, where Tamura has a much wider score spectrum. From our experiments the lowest score that a data-sample got was around 500 000. The high score made a considerable impact on how fast the algorithm calculates to the final score for comparison. From this plot, we inspect that the blue line, CEDD, is generally faster than JCD although with the exception of two cases, wp70 and wp66. Due the large time consumption produced by Tamura we could not fit the elapsed time for Tamura in the same plot. For illustration purposes we included the timings for Tamura in figure 4.13b. Paying attention to the Y axis-es, we notice that Tamura is significantly slower in comparison. Therefore, we tried to simplify the calculation by minimizing the values by having the score divided by 100 000. However, the division itself has to deal with the large score, which was what we intended to avoid in the first place, and as a result did not reduce the algorithm time. Making a method that cast the number as a string and moving the comma a few spaces to the left and reversely casts the string back to a number, might skip the division, however, this was not something we got time to investigate further on.



(a) Alorithm time for JCD and CEDD



(b) Alorithm time for Tamura

Figures show total time consumption of both per frame and algorithm of different videos.

Figure 4.13: Frame and algorithm time consumption

Continuing the experiments, we have plotted the total processing time per frame in figure 4.14 and as we can see the large difference in algorithm time from the previous plots (see figure 4.13a and 4.13b) is not as drastically different. Nonetheless, it is clear that CEDD, the blue line, is the feature that is more suitable for real-time polyp detection, with a maximum difference of roughly 150ms at worst compared to Tamura, and 30ms compared to JCD.

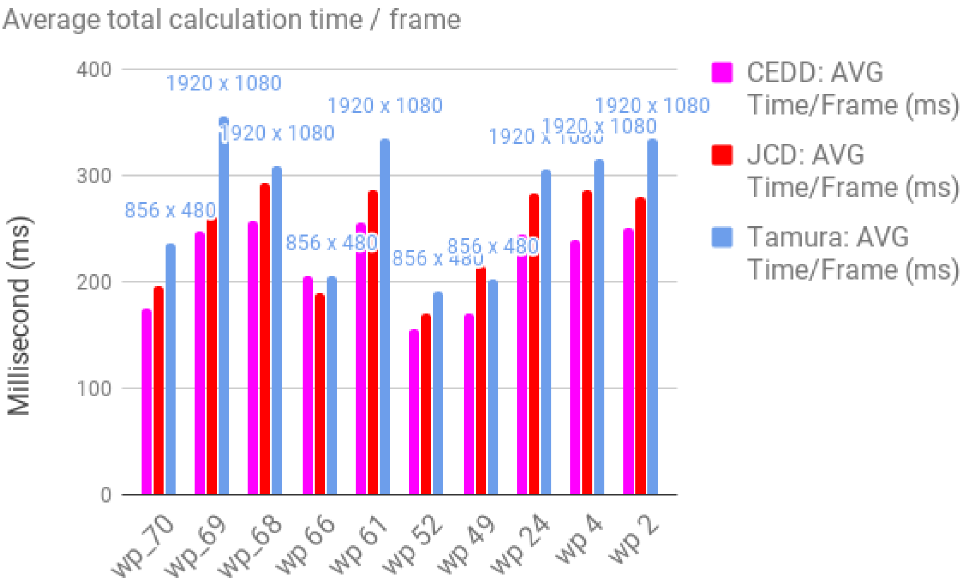


Figure 4.14: Total processing time per frame

4.7.5 Iteration III: Summary

Following iteration II into III, we have designed and implemented a fully working system with a more reliable polyp classifier than what was provided in the previous build. The classifier now scores 25% better weighted F1 score and generally higher in all classification characteristics. However, there is still room for optimized improvements. The training-set is indexed using multi-threading for a much faster performing indexer.

Iteration II did not have any alert protocol, meaning there was no direct feedback to the endoscopist about whether or not there was a polyp in the frame. This has now been implemented into the GUI which the endoscopist is examining the colon. Finding the best solution for alerting the doctor has not been out of this research's scope and in reality boils down to performing surveys of what endoscopist find more comfortable and from there analyze the findings to determine a common ground that can be implemented.

During iteration III, a workaround for the low FPS which we also encountered during iteration II was implemented. This solution is based on the idea of skipping frames mainly because the detection of polyps is not frame based. This means that even though our system detects, for descriptive purposes, 80 positive frames out of 100, it does not mean that the endoscopist will miss this one polyp. If one polyp is only visible in 4 frames and our system reads one frame and skips six, where 4 of those skipped frames is the polyp, then this solution would not be a reliable way of solving the low FPS problem. Since the endoscopist slowly traverse the equipment through the colon, the frame skipping idea is made possible. That said, our system evaluation is frame based and not polyp based. The pace at which the system skips frames is based on the achieved FPS rate in order to not skip unnecessary frames. In the future, these skipped frames could be stored and further classified for after-procedure feedback to the endoscopist.

The experiments and evaluation revealed promising results along with the limitations of the system and what potentially might be possible to achieve further down the line of development. Using only 1 thread, our system is able to achieve an average FPS of 6. The processing time per frame averages at around 220ms. Compared to [48],[figure 3.22 90/281] this faster than what EIR achieves with 1 thread, which is over 350ms for the java implementation and 240ms in C. By utilizing more cores and threads we would be able to achieve faster frame processing, although based on the premise that we also implement a buffer - or Que-model where frames got processed simultaneously or cued depending on how many threads are available.

4.8 Classification improvements

During previous iterations, we were able to reach 75% classification rate and to these classification characteristics of our classifier, we turn our focus towards classification which could in the future replace our java classifier algorithm implemented in iteration

III (described in section 4.7.3) into the pipeline for the entire real-time system. This section will rather focus solely on classification of polyps with goal of getting at least above 85% classification score, not the system in its entirety. We will take advantage of Google tensorflow [2] for feature extraction based on Inception v3, and various machine learning algorithms such as neural network [53] , KNN, and randomforest, discussed under section 3.3, for classification purposes through the Sci-Kit Learn library [40] .

Regarding the data-set, we also rotate our ASU-mayo data-set (4.1, 4.2) in order to artificially increase our data and see if this affects our classification characteristics not any shape or form. During this section, we use a mac-book pro machine (computer two in 4.3) along with ski-learn to take full advantage of different machine-learning algorithms.

4.8.1 Model Creation

During this iteration our main focus will be improving our polyp classification and reach better detection scores. To do so, we took advantage of different machine learning algorithms and this time with python programming language. The reason for our programming language choice this time was due to the many possibilities and machine learning tools available in the python. The most important algorithms we used are described under 3.3 in details.

4.8.2 Classifier Structure

In order to take advantage of the machine learning algorithms in the sklearn library, training the classifier required preparation of the entire data-set we have at disposal. Before the features where extracted, the data-set was sectioned into two categories, training and validation (t1 and t2), as seen in table 4.8. Both partitions have a positive folder called 1 and a negative folder called 0. As our implemented program reads the images inside both partitions, we write the image's file-name, ImageNet's Inception v3 1008 class categories as feature values and, at last, the folder-name to a file, which is either 1 (one) or 0 (zero) depending on which folder the image is inside. Table 3.2 illustrates how this text file is structured. When preparation of the data-set is done, we have one text file for images in the train folder and one text file for the validation images. Both text files will have a set of positive - and negative images.

The classifier has been implemented in a separate python file, where these files are loaded and split in such a fashion that the machine learns and by examples creates patterns for how the feature values for positive - and negative images looks like. This pattern is stored in a model, which can be saved and later loaded to avoid training the model over again. We implemented the model training in a clever fashion, where we have the opportunity to train several machine algorithms in a single run and compare the results. Since the features for the entire data-set takes a long time to load, this was extremely efficient in terms of experimenting with different algorithms and, last but not least, time.

4.8.3 Classifier Experiments and Evaluation

As in the previous system evaluation, under this section we evaluate the classifier in the same manner, and analyze the graphs in order to see if there are any improvements in classification performance characteristics, what might have impacted these results and what these numbers mean on in a greater context.

Classification Score

As we could see from the classification experiments in Iteration III, the differences between JCD and CEDD were minimal, although CEDD was somewhat faster. On the other hand, Tamura seems to perform well in some rare cases, which has something to do with how Tamura interpret the coarseness, contrast and direction of texture in an image. Therefore, we decided to experiment with early fusion of JCD with Tamura, which can be seen in figure 4.16. These figures show the classification characteristics of the chosen features with 8 different machine learning algorithms in order to evaluate which of the algorithms are better suited for polyp detection. For the sake of evaluation, we compare and analyze our JCD score using division-over-score algorithm from iteration III against these 8 algorithms using both JCD alone and fused with Tamura.

The histogram in figure 4.15 shows the score for each of the algorithms with JCD. These scores are randomly distributed since the algorithms are so different in how they function. However, six of eight algorithms score above 75% in weighted F1 score and weighted recall and all score are above 80% weighted precision. The top three performing algorithms are from high to low, neural networks, adaboost which is based on decision trees algorithm and, lastly, random forest. Neural networks achieve a score of around 88% in all characteristics and is our best performing algorithm and uses two hidden layers. The input layer has nodes equal to the number of features, which is 166 for JCD. The first hidden layer has 100 nodes and reduces to 50 in the second hidden layer due to decreasing number of nodes are activated during training. The last layer will have the same number of nodes as the number of classes, which in this case is negative and positive. Looking at the statistical measures of performance (TP,TN,FP,FN) in table 5.1, the high TN value indicates that negative images are easier to classify in their proper class compared to positive images. We therefore analyzed the entire data-set and noticed the high number of total negative images, which plays a role in why negative images are correctly classified.

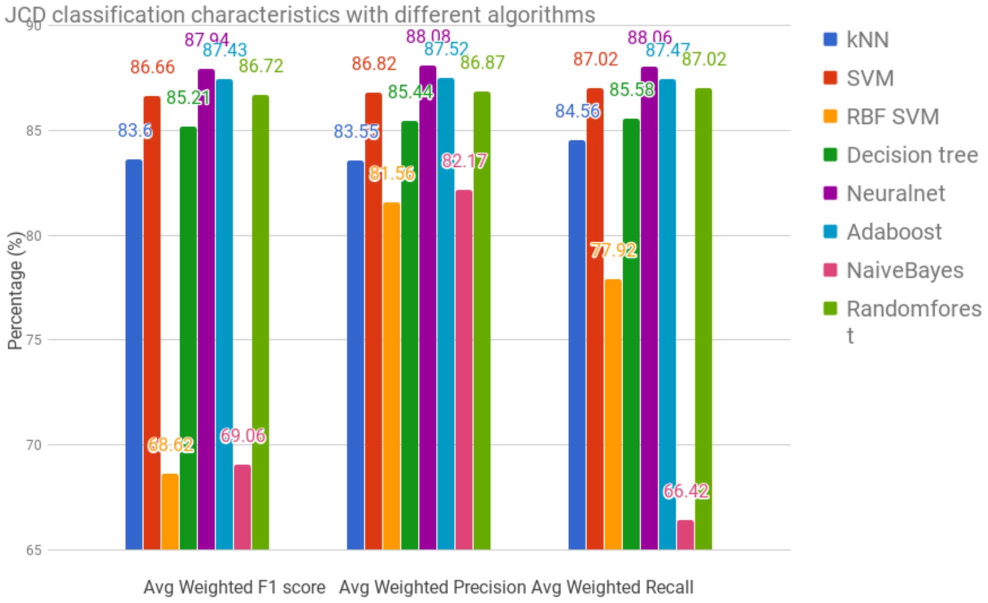


Figure 4.15: Classification characteristics results for different machine learning algorithms with the JCD feature. These characteristics are weighted F1 score, weighted precision and weighted recall, where neural networks is the best performing algorithm.

For the next experiment, we combined JCD and Tamura by using early fusion expecting to get slightly better scores in the classification characteristics. To our surprise, the result got slightly reduced. Looking at figure 4.16, we notice almost no difference at all, however analyzing the measurements in table 5.3, the calculation equals to a approximately 0.10% in reduction. The reason for the decrease in the scores is due to the fact that Tamura is not a high performing feature for the data as JCD and CEDD. This was also something that stood out in the experiments in iteration III. Nonetheless, these result are quite good and by including Tamura in training of the classifier, it might in the future perform better on another polyp data-set and show proof that it was beneficial. On the other hand, this experiment combined the two features with early fusion and does not guarantee that the result would not be better with late fusion, which they usually are. Support Vector Machine (SVM) algorithm performs, in this case, pretty high since SVM is a algorithm that is suitable for classifying multidimensional features with binary classes.

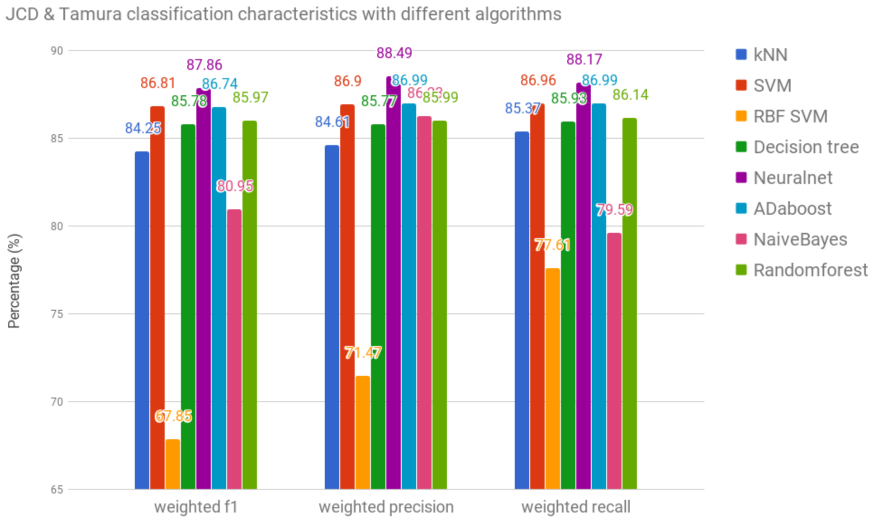


Figure 4.16: Classification characteristics results for different machine learning algorithms with early fusion of JCD and Tamura features. These characteristics are weighted F1 score, weighted precision and weighted recall, where neural networks is the best performing algorithm.

For this experiment, we used Inception v3 model [2][1] for classifying images in the ImageNet data-set into the 1008 class categories to classify our entire data-set. Polyps are not one of the 1008 classes, however by utilizing this model we will have 1008 features describing our every image in our data-set. This way the machine might have a better chance of classifying our data and achieve higher classification performance. In figure 4.17, we have displayed results using this many features and notice higher scores in the three characteristics in more algorithms that for JCD in fusion with Tamura. The results was not drastically different, and our expectation of achieving better polyp classification by using inception v3 categories as features, was not attained.

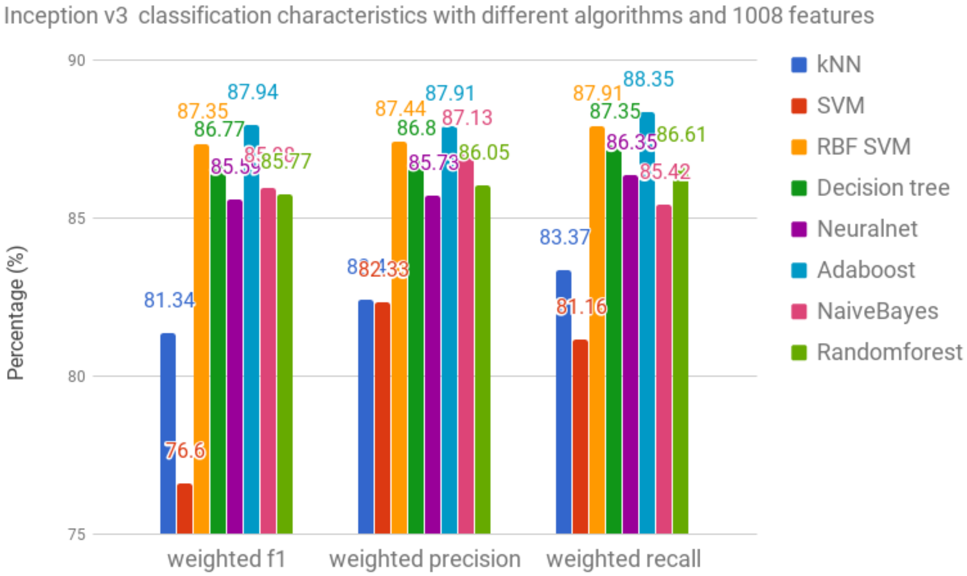


Figure 4.17: Classification characteristics results for different machine learning algorithms with 1008 features which in actuality is probabilities for the image to be on of the 1008 categories. These characteristics are weighted F1 score, weighted precision and weighted recall, where adaboosting decision trees is the best performing algorithm.

4.8.4 Rotation

The results from previous experiments under section 4.8.3 are huge improvements over the polyp classifier in iteration III. However, to increase the quantity of our data-set 4.8, we also created three different duplicates based on rotation. First duplicate is the original images rotated 90 degrees, second duplicate is rotated 180 degree and the last duplicate is 270 degrees. We ran tests with all features on original images, original + 180 degrees rotation and original + 90 + 180 + 270 degrees rotation. The experiment that tested all rotations was only tested with JCD and Tamura in early fusion. Examining the scores in the result tables under section 5.4, we do not notice benefits form rotating our data for classification. The achieved results are either around the same or worse compared to the unrotated approach during early experiments in iteration 4 4.8. These experiments were partly out of our research's scope and we did not emphasize further. Although, we ran some experiments and more in-depth research has to be done on this particular subject.

Table 4.8: The entire ASU Mayo clinic date-set contains the total of 36476 frames divided into two main folder. The first folder contain 17574 frames and second contains 18902 frames.

	Pos	**Neg**	**Sum**
Test	**4313**	**13261**	**17574**
Train	**3856**	**15046**	**18902**

4.8.5 Iteration IV: Summary

In this iteration IV section, we have presented our attempts to use supervised learning and improve polyp classification by using different machine learning algorithms in python as well as combining JCD with Tamura. Tamura was chosen due to the fact that the experiments in iteration III revealed that Tamura would be suitable for some polyps images where JCD and CEDD performed poorly. This iteration used the entire data-set we had available and was split into train and test. We cross- validated our experiments by training the model with the train data-set and validated with the test data-set and vice versa. We implemented code that extracts feature values of the entire data-set and creates two files.

One file that include the feature values for the train data-set and respectively for the test data-set. By comparing the polyp classification characteristics for JCD vs JCD in early fusion with Tamura, the experiments reveal that the results was highly similar. Therefore, we conclude that the data-set does not reflect how high the combination of feature might score.

This iteration also experiments with Google Tensorflow to use the Inception v3 model and classifies our entire data-set into the 1008 class-categories as features. Having 1008 features describing the data-set showed improved classification in most of the eight algorithms. Adaboost decision tree based algorithm performed the best, achieving a score of 87.94% weighted F1, 87.91% weighted precision and 88.33% weighted recall. Due to unbalance data-set, we notice higher F1, precision and recall scores for the negative images, which is a result of having more negative images in the entire data-set than positive. Lastly, we experimented with rotation of our data-set by 90,180 and 270 degrees in order to artificially increase our data-set.

4.9 Summary of chapter 4

In this chapter we have gone through a journey of system development iterations and presented each build into separate sections. Each system iteration section is build on the previous one. The first iteration, real-time color detection, was about getting to know with OpenCV and retrieves frames from the camera which gets processed. The detection is based on color, or rather HSV given as parameters which more or less is unique for the polyp. After the frame is captured, it is eroded using image processing tools in OpenCV to remove unevenness around the objects in the image. Dilation is then used to add back the edge to objects in the image, however not textures that disturbs the edges. By doing so, we accomplish reduce noise in the frame. Lastly, we do mask the frame based on the parsed HSV parameters. The HSV parameters is extracted manually using a separate script. This iteration was due to the fact that object detection using colors is in this case almost impossible. This is because color is not distinct to any object and the object's color will eventually change based on how close or far away the light source is to the object we are trying to detect. Due to this phenomena, this iteration was discontinued. Despite the discontinuity, we measured the frame processing time and how long it takes to retrieve the frame from the camera. These results show that the erosion, dilation and masking itself was quick fast and took averagely 6.98ms per frame. However, the the average frame capture including the processing takes 48.88ms. The color detection made us more knowledgeable with OpenCV and decided to take advantage of the library in the next build.

The next iteration was meant to be a prototype system. It is based on the LIRE platform and takes advantage of the fast image retrieval and feature extraction along side OpenCV, which is our tool to connect to the camera and retrieve frames. We use LIRE's build in method for indexing feature values with JCD,CEDD and Tamura. This step is a preparation step and is required for the system to classify frames real time. The real-time system enables the camera and our system retrieves the frames which is sent to the searcher. The searcher uses a method from LIRE that compares the retrieved frame with all images that was indexed beforehand. Based on which features the training-set set was indexed with, we create a score based ranked list where images in the index with highest similarity is ranked highest. The list will be sent to the classifier algorithm which uses the scores to decide which class the frame belongs to. We implemented two algorithms which is analyzed and evaluated based the main characteristics, weighted F1, weighted precision and weighted recall. One algorithm takes the score for all the positive - and negative images respectively and calculates the average, where lower score means better match. The other algorithm linearly increases the score of each element in the ranked list. The purposes for this is to make the difference between element i, i+1, i-1 bigger with a bias factor in order to increase the polyp classification. The used data-set was a small batch 500 images, 250 to be indexed and 250 to validate with. With 4K images the system achieves 2 FPS, although with downscaled image the FPS reached around 11. The classification characteristics was scored between 50%-65%. The measured algorithm time was almost halved when classifying the downscaled images.

In iteration III, which is the main model, we build on the skeleton which was the system under iteration II. By implementing a multithreaded indexer, training time was drastically reduced from over 12 hours to roughly 45 minutes on a system with 2 cores and 4 threads. The developed GUI displays the frame directly to the endoscopist while the frames are being processed in the background with the implemented frame skipping idea to keep feedback real time. Our new algorithm, division-over-score, performs much better than iteration II regarding classification characteristics. We used 10 videos and did leave-one out cross validation to evaluate the system in terms of FPS, classification characteristics, elapsed algorithm time and total processing time per frame on all 10 videos with all features. Classification performance had increased with over 25% in some runs compared to the previous iteration. CEDD was the fastest performing features and JCD was as fast, however Tamura as noteworthy slower. The system is fully working and could be deployed in the collaborating hospitals. These result are polyp detection done frame by frame, meaning out of 100 positive frame our system detects roughly 3/4 of those frame which result in the system detection this one polyp which is present in the 100 frames.

The last iteration, IV, is not a total rework of the entire system. It is a complete remake of the classification module which is intended to replace the classification pipeline in iteration III. This script was developed using the python programming language. We take advantage of the machine learning algorithms provided by the sklearn library and use eight of these algorithms to classify our images featured with JCD, JCD in early fusion with Tamura. We also use Google Tensorflow to classify our data-set based on the Inception v3 model which assigns 1008 probabilities to each image in our data-set, where these probabilities refer to their according classes. However, we use these 1008 category probabilities as features for our data-set in combination with the selected algorithms. Having 1008 features describing the data-set showed improved classification in most of the eight algorithms. Adaboost decision tree based algorithm performed the best, achieving a score of87.94% weighted F1, 87.91% weighted precision and 88.33% weighted recall. Due to unbalances data-set, we see higher F1, precision and recall scores for the negative images, which is a result of having more negative images in the entire data-set than positive. Lastly, we experimented with rotation of our data-set by 90,180 and 270 degrees in order to artificially increase our data-set.

Chapter 5

Analysis and Conclusion

5.1 Summary

During this research, we implemented a real-time detection system with polyps in the GI tract in mind. We have examined with different features and algorithms to improve polyp detection on computers with low computational resources. We have also developed and implemented the idea of skipping frames as a work around for low FPS on these computers. Our project started with previous research done on this very topic, such as EIR and Polyp-Alert. These are complete state-of-the-art systems, which in the near future could fully replace traditional gastroscopy methods deployed at the hospitals.

Our developed systems has been divided into four separate iterations, where each iteration is build on the previous implementation where the next has improved the lack of some aspects in the previous model. We began by looking into color detection, and so this thesis also discusses why color detection is not suited for object detection in the first iteration.

The second iteration focuses in it's entirety on the object detection phenomena and to implement a pipeline that combines two libraries in order to have a fully operating skeleton of the end goal. We use OpenCV to capture frames and implement methods for conversion of the frame. LIRE is the backbone of feature extraction and image-based search through our training-set. The ranked list created by LIRE is used in our algorithms for polyp classification. We evaluated the performance of our system at that stage and made notion on the aspects that needed improvement. These were improved FPS, higher classification score and a more refined GUI that displays feedback.

The main system developed is implemented as a pipeline which is able to retrieve images from a camera and use the indexed images in the training-set to classify the frame as either positive or negative with the implemented algorithm. The complete system is built on LIRE's tools for extraction of feature values and ranked list creation. We also use OpenCV, which is as library that enables the camera and provides us with tools used to grab and process frames. We have tailored the evaluation pipeline for characterizing our system and measuring the performance. Experimentation was done with three different features JCD, CEDD, and Tamura to evaluate which of these features would best suite polyp detection. Further, experimentation's on algorithms show that classification with the third algorithm, division-over-score, gave best classification characteristics over the algorithms used in iteration II, the best being JCD and CEDD. For the evaluations in iteration III, we used leave-one-out cross validation with 10 videos and have listed the results in both graphs and tables.

This last iteration is developed as a module which in the future could replace the implemented classification pipeline in iteration III and since polyp classification did not reach the heights we where expecting, we turned to python and Google's Tensor-flow Inception v3 model. This model is pre-trained and classification is based on the highest probability of 1008 classes. We use these 1008 probabilities as features for our entire data-set in combination with eight sklearn machine learning algorithms and see huge polyp classification performance. These experiments was also ran with JCD feature values and JCD in early fusion with Tamura. The classification characteristics scored in the top 80%, where the best performing algorithms being Neural networks with JCD and JCD fused with Tamura, and adaboosted decision trees for 1008 Inception v3 feature values. This thesis has touched many different aspects of object detection being features and how features describe images, development and implementation of a detection system which had the goal of process images real time, how our algorithm classifies images with the help of open source libraries, how classification could be improved using unorthodox features and lastly how ideas and work-around software can compensate for low computational power in small for factor machienes.

5.2 Main contribution

This thesis contributes with the realization of running real-time polyp detection on computers with low computers resources with the help of open source projects, such as LIRE, OpenCV and EIR. We have also introduced and implemented an idea on how low computational system can also achieve real-time processing by compensating with clever software development. To achieve this, our pipeline consists of training of the model, frame retriever from camera, frame processing, frame skipping as a work-around to compensate low FPS, real-time classification while visualizing a smoother video for the endoscopist. Due to the recent popularity of machine learning and its advantages in image processing, we have also implemented a stand-alone classifier and assessed it's performance for polyp detection. Our system achieves up to a promising 88.5% polyp detection pr frame with the help of different machine learning algorithms.

The implemented idea for reaching artificial real-time performance by skipping frames, is our main contribution to the research in this field. By taking advantage of this technique, already systems with low performing FPS can reach real-time processing and system reaching as high FPS as 300, such as EIR (see 2.1), can use this technique to look for several diseases during a run. This means that the system can process and classify several categories, not only positive or negative. That said, these days mobile smart phones are being manufactured with increasing processing power that with clever program implementation, these devices also will be able to manage real-time image processing locally and is why Google has newly launched Tensorflow compatible for mobile devices.

In the start of this research, our problem statement consist of following question which can be answer as following:

5.3 Problem Statement

As explained introduction-wise in the section above, there are many factors to take into account, but there is a great potential for improvements in an automated polyp detection system. The purpose of this research is to design and implement an automated live detection system that meets the requirement to capture polyps during an examination procedure and make the endoscopist aware of them. Within this domain, very little research has been done, and in this thesis, we will present a system that achieves state of the art performance. This system is based on the idea of EIR [49]. EIR is named after the goddess of healing in Norwegian mythology and is an interdisciplinary research of a multimedia system that can be used as a tool in the detection of polyps in the GI tract. The challenges we encounter are:

- ***Is it possible to detect polyps live on computers with low computational resources? How will they perform?***

  We have developed and implemented a pipeline that shows that it is possible to achieve a real-time image processing to a certain extent. We implemented our system without the use of multi-threading or installing any form for additional computational power, such as GPU. Although, the system was not up to pair with systems that take advantage of powerful GPU-processing power, we still managed to achieve promising and encouraging results. Our system shows that we in the near future will also have the possibility to take advantage of mobile devices (which nowadays are getting better and better build-in APU) for polyp detection and hopefully reduce mortality rates caused by cancer disease.

- ***Will features and image resolution in combination with different algorithms improve performance with regard to FPS and accuracy?***

  Our experiments revealed that resolution will have the biggest impact on FPS and we notice huge FPS increases as we down-scaled 4K images to 744 x 992. The algorithm times was approximately halved, which results in higher frames processed per second. The 4K images took slightly less than a second to process, from the time the frame is captures to the classifier has finished calculation and outputs the prediction. Analyzing classifications results, in theory the higher the resolution, the clear the object. This means that, classifying high resolution images will score better. Although, during iteration II and III, we discovered that Lire's similarity searches does not take resolution into consideration. Therefore, our experiments revealed that some 480p videos does in some cases achieved higher classification score than 1080p videos. Aside from resolution, some features are faster than others. This is due to CEDD's feature values are faster to extract and later on calculate in comparison to Tamura, which generally results in a higher FPS. JCD is almost as fast as CEDD, however the difference is noticeable.

- ***How will unorthodox features perform in terms of classification compared the global image features of LIRE?***

  This was an interesting topic and during this study, we implemented machine learning algorithms to improve our classification along with Google's Tensorflow Inception v3 model used as features to describe our data. Using these algorithms in combination with JCD and Tamura we got much better classification scores. The inception v3 features performed competitively with LIRE's global image feature, despite the fact that inception v3 was not train for the polyp data in this medical scenario.

5.4 Future Work

For future work, researchers can focus on the findings of the research. There is still potential for improvements for better classification algorithms and methods on computers with low computational resources. Towards the end of the research, we implemented recently trending artificial intelligence machine learning algorithms for classification and got promising results. In the future, researchers can experiment on implementing real-time video analysis along with sophisticated deep learning approaches for even more mobile devices, such as smart phones, tablets and small form-factor machines without advanced capabilities for scientific data-processing.

Other related research done on polyp detection focuses on Polyp-Alert, however during our research, we focused on frame by frame real-time polyp detection. This means that one polyp might have several positive frame in a video, depending on how long it is visible. Even though we did not achieved same high score as Polyp-Alert, we conclude that our system without any additional GPU power achieved encouraging results. There is still much that can be experimented on in regards to how endoscopist are alerted.

Our GUI is a simple solution, however as mentioned above, this has to be made in fully cooperation with the doctors and research on how they would feel comfortable to be notified as they carry out the procedure. Since we implemented an alternative classifier with deep learning approaches in python, there is also the possibility of building a wrapper and replace the java classifier with python and utilize the already existing GUI. As far as performance goes, researchers would have to invest some time to implement a multi-threaded build of our developed system and a more optimized code in order to fully utilize the devices already low power.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[3] Luís A Alexandre, Joao Casteleiro, and Nuno Nobreinst. Polyp detection in endoscopic video using svms. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 358–365. Springer, 2007.

[4] Stefan Ameling, Stephan Wirth, Dietrich Paulus, Gerard Lacey, and Fernando Vi-

larino. Texture-based polyp detection in colonoscopy. In *Bildverarbeitung für die Medizin 2009*, pages 346–350. Springer, 2009.

[5] David J Brenner and Eric J Hall. Computed tomographyan increasing source of radiation exposure. *N Engl J Med*, 357:2277–84, 2007.

[6] Prevention Centers for Disease Control. Cdc, et al. vital signs: colorectal cancer screening test useunited states. 62(44):881, 2013.

[7] Maxwell M Chait. Gastroesophageal reflux disease: Important considerations for the older patients. *World journal of gastrointestinal endoscopy*, 2(12):388, 2010.

[8] Savvas Chatzichristofis and Yiannis Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. *Computer vision systems*, pages 312–322, 2008.

[9] 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 *International Conference on Mass Data Analysis of Images and Signals in Medicine, Biotechnology, and Chemistry*, pages 62–72. Springer, 2008.

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[11] C. Goustout D. Adler. Wireless capsule endoscopy. 2003, (14-22).

[12] Venu Dasigi, Reinhold C Mann, and Vladimir A Protopopescu. Information fusion for text classificationan experimental comparison. *Pattern Recognition*, 34(12):2413–2425, 2001.

[13] Piet C De Groen, Michael Szewczynski, Felicity Enders, Wallapak Tavanapong, JungHwan Oh, and Johnny Wong. Real-time feedback during colonoscopy to improve quality: How often to improve inspection? In *Smart Education and Smart e-Learning*, pages 501–512. Springer, 2015.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[15] Peter Denning, Douglas E Comer, David Gries, Michael C Mulder, Allen B Tucker, A Joe Turner, and Paul R Young. Computing as a discipline: preliminary report of the acm task force on the core of computer science. *ACM SIGCSE Bulletin*, 20(1):41–41, 1988.

[16] Allen J Dietrich, Jonathan N Tobin, Christina M Robinson, Andrea Cassells, Mary Ann Greene, Van H Dunn, Kimberly M Falkenstern, Rosanna De Leon, and Michael L Beach. Telephone outreach to increase colon cancer screening in medicaid managed care organizations: a randomized controlled trial. *The Annals of Family Medicine*, 11(4):335–343, 2013.

[17] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[18] Hofstad B. Bretthauer M. Eide T. J. Gondal G., Grotmol T. and Hoff G. The norwegian colorectal cancer prevention (norccap) screening study. scandinavian journal of gastroenterology. 38(6):635642, 2003.

[19] Ajith H. Gunatilaka and Brian A. Baertlein. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE transactions on pattern analysis and machine intelligence*, 23(6):577–589, 2001.

[20] Douglas M Hawkins. The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12, 2004.

[21] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[22] Øyvind Holme, Michael Bretthauer, Atle Fretheim, Jan Odgaard-Jensen, and Geir Hoff. Flexible sigmoidoscopy versus faecal occult blood testing for colorectal cancer screening in asymptomatic individuals. *The Cochrane Library*, 2013.

[23] Sae Hwang, JungHwan Oh, Wallapak Tavanapong, Johnny Wong, and Piet C De Groen. Polyp detection in colonoscopy video using elliptical shape feature. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pages II–465. IEEE, 2007.

[24] Christine M Peterson Jay P Heiken and Christine O Menias. Virtual colonoscopy for colorectal cancer screening: current status. S133S139, 2005.

[25] J Kang and R Doraiswami. Real-time image processing system for endoscopic applications. In *Electrical and Computer Engineering, 2003. IEEE CCECE 2003. Canadian Conference on*, volume 3, pages 1469–1472. IEEE, 2003.

[26] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[27] Blair S Lewis and Paul Swain. Capsule endoscopy in the evaluation of patients with suspected small intestinal bleeding: results of a pilot study. *Gastrointestinal endoscopy*, 56(3):349–353, 2002.

[28] Baopu Li and Max Q-H Meng. Tumor recognition in wireless capsule endoscopy images using textural features and svm-based feature selection. *IEEE Transactions on Information Technology in Biomedicine*, 16(3):323–329, 2012.

[29] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[30] Teri A Longacre and Cecilia M Fenoglio-Preiser. Mixed hyperplastic adenomatous polyps/serrated adenomas: a distinct form of colorectal neoplasia. *The American journal of surgical pathology*, 14(6):524–537, 1990.

[31] Mathias Lux and Savvas A. Chatzichristofis. Lire: Lucene image retrieval: An extensible java cbir library. pages 1085–1088, 2008.

[32] Mathias Lux, Michael Riegler, Pål Halvorsen, and Glenn MacStravic. Liresolr: A visual information retrieval server. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 466–469. ACM, 2017.

[33] Mathias Lux, Michael Riegler, Pål Halvorsen, Konstantin Pogorelov, and Nektarios Anagnostopoulos. Lire: open source visual information retrieval. In *Proc. of MMSys*, 2016.

[34] Alexander V Mamonov, Isabel N Figueiredo, Pedro N Figueiredo, and Yen-Hsi Richard Tsai. Automated polyp detection in colon capsule endoscopy. *IEEE transactions on medical imaging*, 33(7):1488–1502, 2014.

[35] S. Kumar P.K. Sethy U. Goenka M.K. Goenka, S. Majumder. Single center experience of capsule endoscopy in patientswith obscure gastrointestinal bleeding. 2011.

[36] Kevin P Murphy. Naive bayes classifiers. *University of British Columbia*, 2006.

[37] Reiko Nishihara, Kana Wu, Paul Lochhead, Teppei Morikawa, Xiaoyun Liao, Zhi Rong Qian, Kentaro Inamura, Sun A Kim, Aya Kuchiba, Mai Yamauchi, et al. Long-term colorectal-cancer incidence and mortality after lower endoscopy. *New England Journal of Medicine*, 369(12):1095–1105, 2013.

[38] World Health Organization. *Global status report on alcohol and health*. World Health Organization, 2014.

[39] Jessica B OConnell, Melinda A Maggard, and Clifford Y Ko. Colon cancer survival rates with the new american joint committee on cancer sixth edition staging. *Journal of the National Cancer Institute*, 96(19):1420–1425, 2004.

[40] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[41] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[42] Hannah Pitanga Lukashok, Carlos Robles-Jara, and Carlos Robles-Medranda. Multiple intestinal erosions as a result of hemorrhage due to parasites: Case reports and review of the literature. *Diagnostic and therapeutic endoscopy*, 2011, 2011.

[43] Konstantin Pogorelov, Sigrun Losada Eskeland, Thomas de Lange, Carsten Griwodz, Kristin Ranheim Randel, Håkon Kvale Stensland, Duc-Tien Dang-Nguyen, Concetto Spampinato, Dag Johansen, Michael Riegler, et al. A holistic multimedia system for gastrointestinal tract disease detection. In *Proceedings of the 8th ACM on Multimedia Systems Conference*, pages 112–123. ACM, 2017.

[44] Konstantin Pogorelov, Michael Riegler, Pål Halvorsen, Peter Thelin Schmidt, Carsten Griwodz, Dag Johansen, Sigrun L. Eskeland, and Thomas de Lange. GPU-accelerated real-time gastrointestinal diseases detection. In *Proc. of CBMS*, 2016.

[45] S Popat, R Hubner, and RS Houlston. Systematic review of microsatellite instability and colorectal cancer prognosis. *Journal of Clinical Oncology*, 23(3):609–618, 2005.

[46] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[47] Michael Riegler, Mathias Lux, Carsten Griwodz, Concetto Spampinato, Thomas de Lange, Sigrun L Eskeland, Konstantin Pogorelov, Wallapak Tavanapong, Peter T Schmidt, Cathal Gurrin, et al. Multimedia and medicine: Teammates for better disease detection and survival. In *Proc. of ACM MM*, pages 968–977, 2016.

[48] Michael Riegler, Konstantin Pogorelov, Sigrun Losada Eskeland, Peter Thelin Schmidt, Zeno Albisser, Dag Johansen, Carsten Griwodz, Pål Halvorsen, and Thomas De Lange. From annotation to computer-aided diagnosis: Detailed evaluation of a medical multimedia system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 13(3):26:1–26:26, May 2017.

[49] Michael Riegler, Konstantin Pogorelov, Pål Halvorsen, Thomas de Lange, Carsten Griwodz, Peter Thelin Schmidt, Sigrun Losada Eskeland, and Dag Johansen. Eir-efficient computer aided diagnosis framework for gastrointestinal endoscopies. In *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*, pages 1–6. IEEE, 2016.

[50] 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.

[51] Raúl Rojas. Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting. *Freie University, Berlin, Tech. Rep*, 2009.

[52] Eitan Scapa, Harold Jacob, Shlomo Lewkowicz, Michal Migdal, Daniel Gat, Arkady Gluckhovski, Nurit Gutmann, and Zvi Fireman. Initial experience of wireless-capsule endoscopy for evaluating occult gastrointestinal bleeding and suspected small bowel pathology. *The American journal of gastroenterology*, 97(11):2776–2779, 2002.

[53] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[54] Amit Pratap Singh and RC Jain. Group identification based classification technique for aggregated common data to save memory space in digital forensic investigation (dfi).

[55] Erik Sintorn and Ulf Assarsson. Fast parallel gpu-sorting using a hybrid algorithm. *Journal of Parallel and Distributed Computing*, 68(10):1381–1388, 2008.

[56] Amnon Sonnenberg and Robert M Genta. Low prevalence of colon polyps in chronic inflammatory conditions of the colon. *The American journal of gastroenterology*, 110(7):1056–1061, 2015.

[57] Steven J Stryker, Bruce G Wolff, Clyde E Culp, Susan D Libbe, Duane M Ilstrup, and Robert L MacCarty. Natural history of untreated colonic polyps. *Gastroenterology*, 93(5):1009–1013, 1987.

[58] P Swain. Wireless capsule endoscopy and crohns disease. *Gut*, 54(3):323–326, 2005.

[59] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings*

of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2818–2826, 2016.

[60] Nima Tajbakhsh, Suryakanth R Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE transactions on medical imaging*, 35(2):630–644, 2016.

[61] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473, 1978.

[62] Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. 1958.

[63] Carlos Torres, Donald Antonioli, and Robert D Odze. Polypoid dysplasia and adenomas in inflammatory bowel disease: a clinical, pathologic, and follow-up study of 89 polyps from 59 patients. *The American journal of surgical pathology*, 22(3):275–284, 1998.

[64] Yi Wang, Wallapak Tavanapong, Johnny Wong, Jung Hwan Oh, and Piet C De Groen. Polyp-alert: Near real-time feedback during colonoscopy. *Computer methods and programs in biomedicine*, 120(3):164–179, 2015.

[65] Yi Wang, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C De Groen. Part-based multiderivative edge cross-sectional profiles for polyp detection in colonoscopy. *IEEE journal of biomedical and health informatics*, 18(4):1379–1389, 2014.

[66] F Wartel, V Maunoury, P Bulois, S Papadopoulos, B Filoche, and JF Colombel. Small-bowel ulcerations at wireless capsule endoscopy: go the whole way. *Gut*, 56(8):1132, 2007.

[67] Jian Yang, Jing-yu Yang, David Zhang, and Jian-feng Lu. Feature fusion: parallel strategy vs. serial strategy. *Pattern recognition*, 36(6):1369–1381, 2003.

[68] Konstantinos Zagoris, Savvas A Chatzichristofis, Nikos Papamarkos, and Yiannis S Boutalis. Automatic image annotation and retrieval using the joint composite descriptor. In *Informatics (PCI), 2010 14th Panhellenic Conference on*, pages 143–147. IEEE, 2010.

[69] Ann G Zauber, Sidney J Winawer, Michael J O'Brien, Iris Lansdorp-Vogelaar, Marjolein van Ballegooijen, Benjamin F Hankey, Weiji Shi, John H Bond, Melvin Schapiro, Joel F Panish, et al. Colonoscopic polypectomy and long-term prevention of colorectal-cancer deaths. *New England Journal of Medicine*, 366(8):687–696, 2012.

[70] Mingda Zhou, Guanqun Bao, Yishuang Geng, Bader Alkandari, and Xiaoxi Li. Polyp detection and radius measurement in small intestine using video capsule endoscopy. In *Biomedical Engineering and Informatics (BMEI), 2014 7th International Conference on*, pages 237–241. IEEE, 2014.

Appendix

The sources code can be downloaded from this github repository: `https://github.com/IcyFrequency/Master_Thesis`

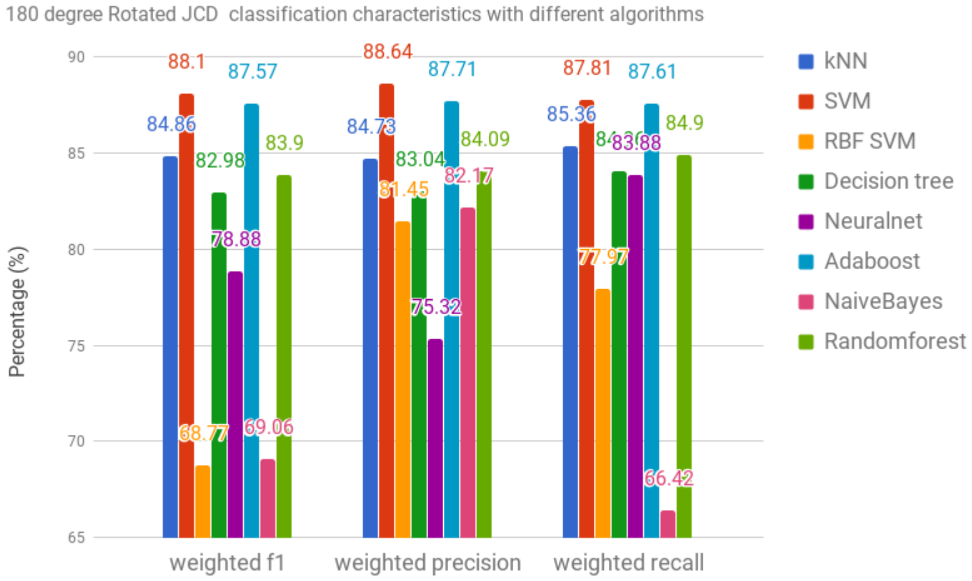


Figure 5.1: Diagram showing the percentage with 180 degrees rotated data with JCD.

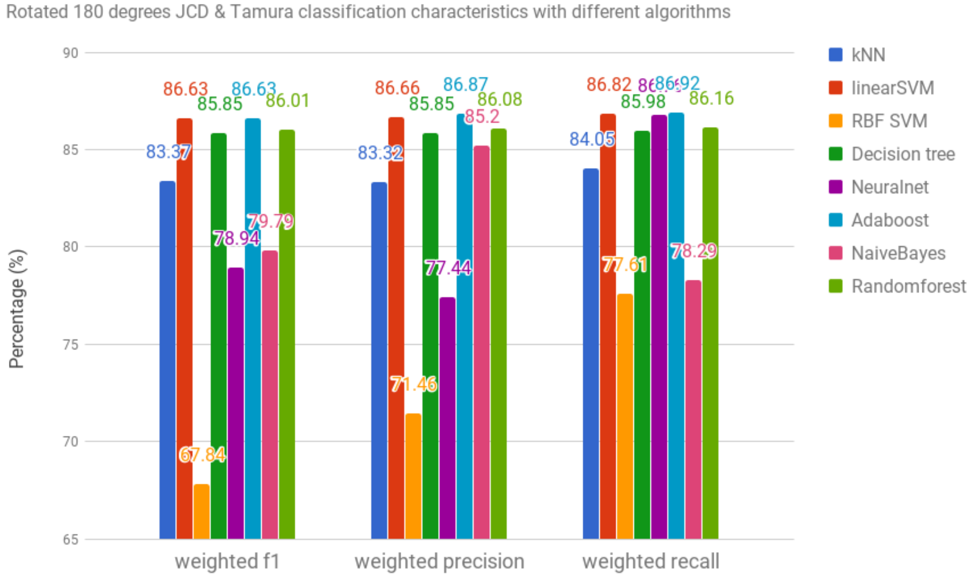


Figure 5.2: Diagram showing the percentage with 180 degrees rotated data and JCD in early fusion with Tamura

Table 5.1: JCD classification characteristics with ski-learn

Using the same feature values generated by LIRE, in the same manner we created the inception v3 text files we made these with JCD. These are results of the classification characteristics using ski-learn machine learning algorithms with the JCD feature values from LIRE.

Method	Split	Total	Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	17574	4313	13261	1694	12264	997	2619	77.63	77.63	79.42
kNN	Test/Train	18902	3856	15046	2684	14202	844	1172	89.15	89.05	89.33
	Avg weighted								83.6	83.55	84.56
SVM	Train/Test	17574	4313	13261	2340	12456	805	1973	83.29	83.4	84.19
SVM	Test/Train	18902	3856	15046	3031	13915	1131	825	89.79	90	89.65
	Avg weighted								86.66	86.82	87.02
RBF SVM	Train/Test	17574	4313	13261	40	13260	1	4273	65.44	81.01	75.68
RBF SVM	Test/Train	18902	3856	15046	84	15037	9	3772	71.58	82.07	80
	Avg weighted								68.62	81.56	77.92
Decision tree	Train/Test	17574	4313	13261	2060	12183	1078	2253	79.95	79.79	81.04
Decision tree	Test/Train	18902	3856	15046	3232	13744	1302	624	90.11	90.69	89.81
	Avg weighted								85.21	85.44	85.58
Neuralnet	Train/Test	17574	4313	13261	2673	12327	934	1640	84.89	84.79	85.36
Neuralnet	Test/Train	18902	3856	15046	3214	13905	1141	642	90.78	91.14	90.57
	Avg weighted								87.94	88.08	88.06
Adaboost	Train/Test	17574	4313	13261	2596	11979	1282	1717	82.62	82.42	82.93
Adaboost	Test/Train	18902	3856	15046	3334	13999	1047	522	91.9	92.27	91.7
	Avg weighted								87.43	87.52	87.47
NaiveBayes	Train/Test	17574	4313	13261	3439	9383	3878	874	74.73	80.56	72.96
NaiveBayes	Test/Train	18902	3856	15046	3582	7822	7224	274	63.78	83.67	60.33
	Avg weighted								69.06	82.17	66.42
Randomforest	Train/Test	17574	4313	13261	2240	12194	1067	2073	81.28	81.12	82.13
Randomforest	Test/Train	18902	3856	15046	3351	13956	1090	505	91.78	92.22	91.56
	Avg weighted								86.72	86.87	87.02

Table 5.2: JCD classification characteristics with 180 degrees rotated copied data

Using the same feature values generated by LIRE, in the same manner we created the inception v3 text files we made these with JCD. These are results of the classification characteristics using ski-learn machine learning algorithms with the JCD feature values from LIRE, although with 180 degrees rotation of copied data.

Method	Split		Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	35148	8626	26522	4178	24202	2320	4447	79.76	79.53	80.74
kNN	Test/Train	37804	7712	30092	5669	28222	1870	2042	89.61	89.57	89.65
kNN	Avg Weighted								84.86	84.73	85.36
SVM	Train/Test	35148	8626	26522	6625	23578	2944	2000	86.17	86.55	85.93
SVM	Test/Train	37804	7712	30092	6501	27356	2736	1210	89.89	90.58	89.56
SVM	Avg Weighted								88.1	88.64	87.81
RBF SVM	Train/Test	35148	8626	26522	110	26518	4	8515	65.64	80.79	75.76
RBF SVM	Test/Train	37804	7712	30092	184	30072	20	7527	71.68	82.06	80.03
RBF SVM	Avg Weighted								68.77	81.45	77.97
Decision tree	Train/Test	35148	8626	26522	3206	24728	1794	5419	77.4	77.63	79.48
Decision tree	Test/Train	37804	7712	30092	5265	28120	1972	2446	88.17	88.07	88.32
Decision tree	Avg Weighted								82.98	83.04	84.06
Neuralnet	Train/Test	35148	8626	26522	1	26511	10	8624	64.89	57.18	75.43
Neuralnet	Test/Train	37804	7712	30092	6587	28090	2002	1124	91.89	92.18	91.73
Neuralnet	Avg Weighted								78.88	75.32	83.88
Adaboost	Train/Test	35148	8626	26522	5192	23958	2564	3433	82.62	82.43	82.94
Adaboost	Test/Train	37804	7712	30092	6785	27978	2114	926	92.17	92.61	91.95
Adaboost	Avg Weighted								87.57	87.71	87.61
NaiveBayes	Train/Test	35148	8626	26522	6878	18766	7756	1747	74.73	80.56	72.96
NaiveBayes	Test/Train	37804	7712	30092	7163	15644	14448	548	63.78	83.67	60.33
NaiveBayes	Avg Weighted								69.06	82.17	66.42
Randomforest	Train/Test	35148	8626	26522	4012	25154	1368	4613	81.5	82.06	82.98
Randomforest	Test/Train	37804	7712	30092	4463	28306	1786	3248	86.14	85.98	86.68
Randomforest	Avg Weighted								83.9	84.09	84.9

Table 5.3: Early fusion of JCD and Tamura classification characteristics

As in table 5.1, the same experiments was also tested on JCD in early fusion with Tamura. This table show the classification characteristics of their performance.

Method	Split	Total	Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	17574	4313	13261	1670	12585	676	2640	78.99	79.84	81.12
kNN	Test/Train	18902	3856	15046	2684	14202	844	1172	89.15	89.05	89.33
	Avg weighted								84.25	84.61	85.37
SVM	Train/Test	17574	4313	13261	2475	12099	1162	1835	82.43	82.22	82.94
SVM	Test/Train	18902	3856	15046	3224	13917	1129	632	90.89	91.25	90.69
	Avg weighted								86.81	86.9	86.96
RBF SVM	Train/Test	17574	4313	13261	1	13260	1	4309	64.92	69.22	75.46
RBF SVM	Test/Train	18902	3856	15046	1	15045	1	3855	70.57	73.56	79.6
	Avg weighted								67.85	71.47	77.61
Decision tree	Train/Test	17574	4313	13261	2420	11989	1274	1890	81.5	81.26	81.99
Decision tree	Test/Train	18902	3856	15046	3035	13900	1146	821	89.75	89.97	89.59
	Avg weighted								85.78	85.77	85.93
Neuralnet	Train/Test	17574	4313	13261	2328	12603	658	1982	83.96	84.34	84.97
Neuralnet	Test/Train	18902	3856	15046	3479	13750	1296	377	91.49	92.34	91.15
	Avg weighted								87.86	88.49	88.17
Adaboost	Train/Test	17574	4313	13261	2185	12058	1203	2125	80.25	79.98	81.06
Adaboost	Test/Train	18902	3856	15046	3577	13909	1137	279	92.77	93.51	92.51
	Avg weighted								86.74	86.99	86.99
NaiveBayes	Train/Test	17574	4313	13261	3188	11309	1252	1122	86.17	86.27	86.09
NaiveBayes	Test/Train	18902	3856	15046	3567	10336	4710	292	76.09	86.2	73.54
	Avg weighted								80.95	86.23	79.59
Randomforest	Train/Test	17574	4313	13261	2378	12005	1256	1932	81.31	81.06	81.85
Randomforest	Test/Train	18902	3856	15046	3115	13922	1124	741	90.3	90.57	90.13
	Avg weighted								85.97	85.99	86.14

Table 5.4: Early fusion of JCD and Tamura classification characteristics 180 degrees rotated copied dat

As in table 5.1, the same experiments was also tested on JCD in early fusion with Tamura, although data was copied and rotated 180 degrees. This table show the classification characteristics of their performance.

Method	Split		Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	35148	8626	26522	4342	24825	1696	4284	81.88	82	82.99
kNN	Test/Train	37804	7712	30092	4512	27635	2456	3200	84.75	84.55	85.04
kNN	Avg weighted								83.37	83.32	84.05
linearSVM	Train/Test	35148	8626	26522	4898	24251	2270	3728	82.38	82.18	82.93
linearSVM	Test/Train	37804	7712	30092	6268	27917	2174	1444	90.59	90.83	90.43
linearSVM	Avg weighted								86.63	86.66	86.82
RBF SVM	Train/Test	35148	8626	26522	1	26520	1	8625	64.91	69.21	75.46
RBF SVM	Test/Train	37804	7712	30092	1	30090	1	7711	70.56	73.56	79.6
RBF SVM	Avg weighted								67.84	71.46	77.61
Decision tree	Train/Test	35148	8626	26522	4938	23921	2600	3688	81.68	81.45	82.11
Decision tree	Test/Train	37804	7712	30092	6066	27797	2294	1646	89.73	89.95	89.58
Decision tree	Avg weighted								85.85	85.85	85.98
Neuralnet	Train/Test	35148	8626	26522	4782	25019	1502	3844	68.66	65.65	84.79
Neuralnet	Test/Train	37804	7712	30092	5378	28115	1976	2334	88.5	88.41	88.6
Neuralnet	Avg weighted								78.94	77.44	86.76
Adaboost	Train/Test	35148	8626	26522	4310	24171	2350	4316	80.15	79.91	81.04
Adaboost	Test/Train	37804	7712	30092	7104	27819	2272	608	92.65	93.35	92.38
Adaboost	Avg weighted								86.63	86.87	86.92
NaiveBayes	Train/Test	35148	8626	26522	6396	22607	3914	2230	83.01	83.91	82.52
NaiveBayes	Test/Train	37804	7712	30092	7126	20981	9110	586	76.8	86.39	74.35
NaiveBayes	Avg weighted								79.79	85.2	78.29
Randomforest	Train/Test	35148	8626	26522	4776	24019	2502	3850	81.39	81.14	81.93
Randomforest	Test/Train	37804	7712	30092	6324	27735	2356	1388	90.31	90.67	90.1
Randomforest	Avg weighted								86.01	86.08	86.16

Table 5.5: Machine learning classification characteristics with inception v3 1008 features

Classification characteristics using Inception v3 1008 features with different machine learning algorithms. These weighted (wght) scores represent the classifiers performance in terms of polyp detection.

Method	Split	Total	Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	17574	4313	13261	1370	12791	470	2943	77.51	79.62	80.58
kNN	Test/Train	18902	3856	15046	1909	14340	706	1947	84.9	84.98	85.97
	Avg weighted								81.34	82.4	83.37
SVM	Train/Test	17574	4313	13261	844	13165	96	2943	76.41	83.71	80.39
SVM	Test/Train	18902	3856	15046	615	14862	184	3241	76.77	81.05	81.88
	Avg weighted								76.6	82.33	81.16
RBF SVM	Train/Test	17574	4313	13261	2608	12657	604	1705	86.16	86.43	86.87
RBF SVM	Test/Train	18902	3856	15046	2469	14328	718	1387	88.46	88.38	88.87
	Avg weighted								87.35	87.44	87.91
Decision tree	Train/Test	17574	4313	13261	2585	12550	711	1728	85.46	85.58	86.12
Decision tree	Test/Train	18902	3856	15046	2379	14346	700	1477	87.98	87.94	88.49
	Avg weighted								86.77	86.8	87.35
Neuralnet	Train/Test	17574	4313	13261	2367	12643	618	1946	84.43	84.85	85.41
Neuralnet	Test/Train	18902	3856	15046	2258	14229	817	1598	86.67	86.54	87.22
	Avg weighted								85.59	85.73	86.35
Adaboost	Train/Test	17574	4313	13261	2816	12555	706	1497	87.02	87.04	87.47
Adaboost	Test/Train	18902	3856	15046	2503	14353	693	1353	88.79	88.72	89.17
	Avg weighted								87.94	87.91	88.35
NaiveBayes	Train/Test	17574	4313	13261	3590	11696	1565	723	87.35	88.16	86.98
NaiveBayes	Test/Train	18902	3856	15046	2935	12935	2111	921	84.7	86.17	83.96
	Avg weighted								85.98	87.13	85.42
Randomforest	Train/Test	17574	4313	13261	2258	12710	551	2055	84	84.68	85.17
Randomforest	Test/Train	18902	3856	15046	2326	14296	750	1530	87.42	87.33	87.94
	Avg weighted								85.77	86.05	86.61

Table 5.6: Machine learning classification characteristics with inception v3 1008 features, this time 180 degrees rotated copied data

Classification characteristics using Inception v3 1008 features with different machine learning algorithms. Data has been copied and rotated 180 degrees. These weighted (wght) scores represent the classifiers performance in terms of polyp detection.

Method	Split		Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	35148	8626	26522	3932	24876	1646	4694	80.52	80.78	81.96
kNN	Test/Train	37804	7712	30092	4698	27682	2410	3014	85.43	85.27	85.65
kNN	Avg weighted								83.06	83.11	83.87
linearSVM	Train/Test	35148	8626	26522	1	26521	1	8625	64.91	69.21	75.46
linearSVM	Test/Train	37804	7712	30092	1	30091	1	7711	70.56	73.56	79.6
linearSVM	Avg weighted								67.84	71.46	77.61
RBF SVM	Train/Test	35148	8626	26522	5226	25344	1178	3400	86.28	86.56	86.98
RBF SVM	Test/Train	37804	7712	30092	4970	28658	1434	7742	79.22	78.5	83.78
RBF SVM	Avg weighted								82.62	82.38	85.32
Decision tree	Train/Test	35148	8626	26522	4782	25036	1486	3844	83.96	84.14	84.84
Decision tree	Test/Train	37804	7712	30092	4400	27940	2152	3312	85.09	84.86	85.55
Decision tree	Avg weighted								84.55	84.51	85.21
Neuralnet	Train/Test	35148	8626	26522	1	26521	1	8625	64.91	57.19	75.46
Neuralnet	Test/Train	37804	7712	30092	1	30091	1	7711	70.56	73.56	79.6
Neuralnet	Avg weighted								67.84	65.67	77.61
Adaboost	Train/Test	35148	8626	26522	5632	25110	1412	2994	87.02	87.04	87.47
Adaboost	Test/Train	37804	7712	30092	5006	28706	1386	2706	88.79	88.72	89.17
Adaboost	Avg weighted								87.94	87.91	88.35
NaiveBayes	Train/Test	35148	8626	26522	7182	23392	3130	1444	87.36	88.17	86.99
NaiveBayes	Test/Train	37804	7712	30092	5870	25870	4222	1842	84.7	86.17	83.96
NaiveBayes	Avg weighted								85.98	87.13	85.42
Randomforest	Train/Test	35148	8626	26522	4376	24580	1942	4250	81.39	81.33	82.38
Randomforest	Test/Train	37804	7712	30092	5014	27386	2706	2698	85.71	85.71	85.71
Randomforest	Avg weighted								83.63	83.6	84.11

Table 5.7: Early fusion of JCD and Tamura classification characteristics, rotated 90, 180 and 270 degrees of the copied data.

As in table 5.1, the same experiments was also tested on JCD in early fusion with Tamura, although data was copied and rotated 90, 180 and 270 degrees. This table show the classification characteristics of their performance.

Method	Split		Pos	Neg	TP	TN	FP	FN	Wght. F1	Wght. Precision	Wght. Recall
kNN	Train/Test	70296	17252	53044	8684	49652	3392	8568	81.88	82	82.99
kNN	Test/Train	75426	15242	60184	9020	55272	4911	6404	84.81	84.59	85.1
kNN	Avg weighted								83.4	83.34	84.08
SVM	Train/Test	70296	17252	53044	9684	48528	4516	7568	64.91	69.21	75.46
SVM	Test/Train	75426	15242	60184	12296	55896	4287	3128	90.35	90.55	90.22
SVM	Avg weighted								78.08	80.26	83.1
RBF SVM	Train/Test	70296	17252	53044	1	53043	1	17251	64.91	69.21	75.46
RBF SVM	Test/Train	75426	15242	60184	1	60182	1	15423	70.73	73.62	79.79
RBF SVM	Avg weighted								67.92	71.49	77.7
Decision tree	Train/Test	70296	17252	53044	9880	47840	5204	7372	81.69	81.46	82.11
Decision tree	Test/Train	75426	15242	60184	12124	55388	4795	3300	89.51	89.79	89.32
Decision tree	Avg weighted								85.74	85.77	85.84
Neuralnet	Train/Test	70296	17252	53044	8932	49380	3664	8320	81.98	81.98	82.95
Neuralnet	Test/Train	75426	15242	60184	11820	55856	4327	3604	89.63	89.75	89.54
Neuralnet	Avg weighted								85.94	86	86.36
Adaboost	Train/Test	70296	17252	53044	8620	48344	4700	8632	80.15	79.91	81.04
Adaboost	Test/Train	75426	15242	60184	14128	55616	4567	1296	92.53	93.24	92.25
Adaboost	Avg weighted								86.56	86.81	86.84
NaiveBayes	Train/Test	70296	17252	53044	12764	45236	7808	4488	82.99	83.87	82.51
NaiveBayes	Test/Train	75426	15242	60184	14256	41336	18847	1168	76.11	86.3	73.48
NaiveBayes	Avg weighted								79.43	85.13	77.84
Randomforest	Train/Test	70296	17252	53044	9720	48028	5016	7532	81.65	81.41	82.15
Randomforest	Test/Train	75426	15242	60184	12124	55664	4519	3300	89.84	90.05	89.68
Randomforest	Avg weighted								85.89	85.88	86.05

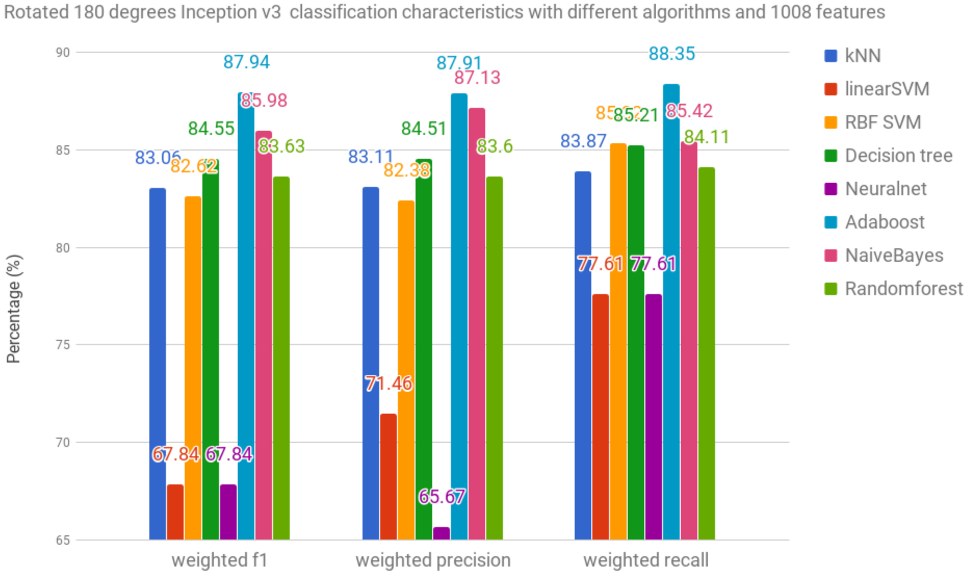


Figure 5.3: Diagram showing the percentage with 180 degrees rotated data with Inception v3

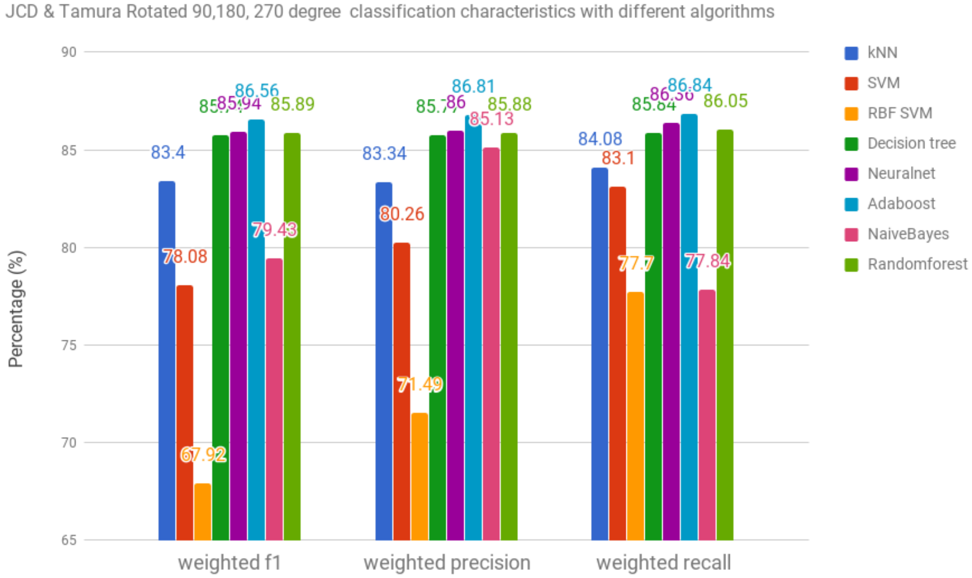


Figure 5.4: Diagram showing the percentage with 90, 180 and 270 degrees rotated data with JCD in early fusion with Tamura