Polyp Detection: Effect of Early and Late Feature Fusion

Salman Asskali

Thesis submitted for the degree of
Master in Robotics and Intelligent Systems
30 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2017

Polyp Detection: Effect of Early and Late Feature Fusion

Salman Asskali

Polyp Detection: Effect of Early and Late Feature Fusion

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

Printed: Reprosentralen, University of Oslo

Acknowledgments

I would like to express my deepest gratitude to my supervisors Pål Halvorsen and Michael Riegler for sharing their expertise and time. Without their continued support, I would not have been able to complete this thesis. I would also like to thank the aforementioned supervisors for giving me the opportunity to work on this project.

Finally I would like to thank to my family for all their understanding and for always being there to help when necessary, and my parents to whom I dedicate this thesis. I would also like to further extend my gratitude towards all my dear friends for helping me disconnect when needed.

Abstract

Computer systems, specially machine learning, has been changing the way we do most everyday tasks. A field in which technological advances have had a significant impact is medical diagnosis. Only decades ago, expert systems were working alongside professional doctors to help diagnose diseases or identify patients at risk. These expert systems, however, had hard-coded rules or functioned based on the opinion of people in the field of what was a relevant feature and what was not. Recent advances in machine learning, and the vast amount of data that has been made publicly available, have led medical diagnosis to move away from expert systems and start embracing these learning methods. The main difference between these two approaches is that in the latter, the system is not being told what is relevant and what is not, but rather it is able to identify those things by itself through large bodies of data.

In this thesis, we look at a specific component of these learning methods and how they affect performance in aiding medical systems. This component, called feature fusion, has two widely adopted variations: early fusion and late fusion. We seek to compare the performance of early and late fusion for medical diagnosis problems through image datasets, and provide some insight to data scientists on how our results can help their decision when building a practical system.

Contents

List of Figures

List of Tables

Part I

Introduction and Theory

Chapter 1

Introduction

1.1 Background and Motivation

Food consumption is an essential process for humans in order to survive and this is also the main factor why the digestive system is prone to many kinds of diseases that could threaten one's life. According to the data from Centers for Disease Control and Prevention (CDC), there are over 7 million appointments to the emergency room every year with a digestive system diagnosis and nearly 40 million visits to office-based physicians for digestive system symptoms[1]. Between 60 and 70 million people are affected by digestive diseases, a classification which includes a wide variety of diseases[2]. Timely diagnosis and detection are vital for the end result, procedures to be undertaken, and the level of treatment for the survival of patients. Gastrointestinal (GI) diseases can range from mild to extreme. The most well-known disturbances include symptoms of stomach pain, heart burn, diarrhea, constipation, nausea, and vomiting[3]. The common diseases are gastroesophageal reflux disease, peptic ulcer disease, inflammatory bowel disease, celiac disease and chronic infections[4].

GI diseases are not just rampant in the United States of America, but also growing in prominence in wealthy Asian countries. According to Gross, GI diseases are emerging and increasingly found in wealthy countries like Japan, South Korea, and China[5]. Furthermore, GI diseases in particular are becoming more common and have been linked to changing environmental factors brought on by industrialization, changes in diet, improved sanitation, and the increased use of antibiotics.

Once a GI disease is diagnosed to a certain patient, endoscopy is important to do next. An endoscopy is a procedure in which a doctor uses specialized instruments to view and operate on the internal organs and vessels of your body which allows surgeons to view problems within without making large incisions. For the traditional process, the surgeon inserts an endoscope through a small cut, or an opening in the body such as the mouth. An endoscope is a flexible tube with an attached camera that allows your doctor to see. Your doctor can use forceps (tongs) and scissors on the endoscope to operate or remove tissue for biopsy[6]. There is an alternative to traditional procedure which is the capsule endoscopy. According to Mayo Clinic staff, it is a procedure that uses a tiny wireless camera to take pictures of a digestive tract. A capsule endoscopy camera sits inside a vitamin-size capsule to be swallowed. As the capsule travels through your digestive tract, the camera takes thousands of pictures that are transmitted to a recorder on a belt around the waist [7]. Scheduled screening is difficult to undertake because such procedure is too expensive, patients are unwilling to undergo the process due to unpleasant sensations, medical expert consume large amount of time to finish the procedure, and scarcity for qualified medical professionals. Timely diagnosis and detection are vital for the end result, procedures to be undertaken, and the level of treatment for the survival of patients.

To help and measure endoscopic examinations, EIR[6][7] was developed at Simula Research Laboratory, named after a goddess with medical skills in Scandinavian mythology. EIR is an end-to-end efficient and scalable information retrieval system for medical data like videos and images, sensor data and patient records, i.e., EIR combines a content-based similarity search with statistical classifiers from the training data. The system supports endoscopists in the detection and interpretation of diseases in the GI tract, but can basically be expanded to any other use-case.

The primary goal of such Computer-aided diagnosis is to increase the detection of disease and reduce false positive. However, the use of EIR does not in any way substitute for a physician because the interpretation of the results has to be made by the radiologist or physician.

To enhance and to maximize the capability of the existing EIR system; the proponents of this project discuss the integration on several measures of fusion and LIRE[8], an open source application. Several research findings from other works and projects for the progress of endoscopy are covered as well.

1.2 Problem Statement

In this thesis, we are interested in investigating different feature extraction techniques, studying their strength and weaknesses, and their usefulness to assists in medical diagnosis through images.

The use of fusion is studied and implemented in our case to increase the performance of EIR system with the goal to be used in a live system, providing real-time feedback.

For this project we used an open Source Software called LIRE. This software is primarily used to extract features from images, and allow us to make a comparison of the two main feature fusion techniques: early fusion and late fusion.

For the fusion part we are using Scikit-learn which is an open source Python library that implements a range of machine learning. We are including different images from the medical scenario (GI tract), and using different features. In LIRE there are around 30 different features which are already implemented. The main goal with the project is testing early fusion and late fusion and comparing them by using different features and various classifiers.

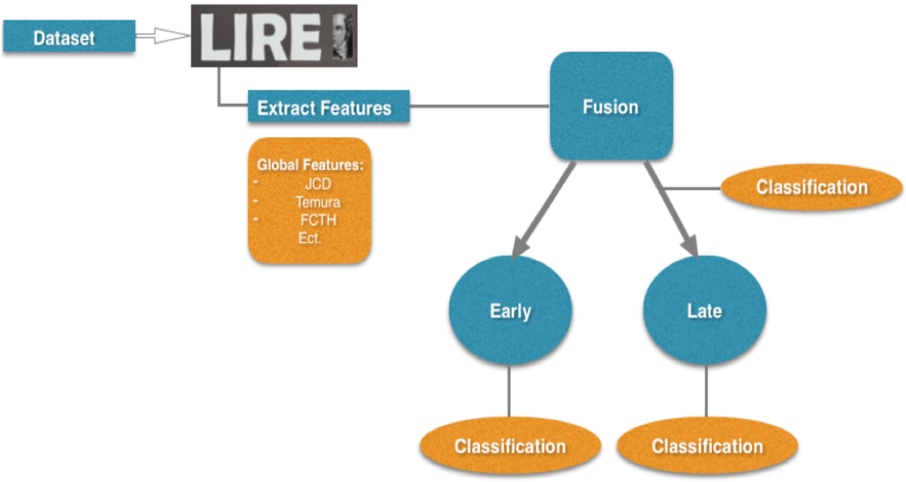


Figure 1.1: Pipeline for the project. Shows us the different steps

1.3 Limitations

It would be unjust to not acknowledge some inherent limitations in our study, mainly due to our time constraints. First, we explored different fusion techniques in image datasets. That means that there is some underlying structure common to all of these datasets (techniques like edge detection are known to be useful in images, but do not easily translate to other sources of data). As indicated in the introduction, there is also various diseases in the GI tract, but due to the datasets accessibility and the focus of this study we pay particular attention to the polyps. The centre of interest is the detection of polyps in the GI tract.

Second, did not explore dimensionality reduction (except for PCA) in depth. These techniques, specially those relying in nonlinear transformations, could have a significant impact on the results obtained.

Last, a reader who is up-to-date with current state-of-the-art methods would probably notice that we have not compared the efficiency of fusion techniques with feature learning algorithms based on deep learning. Again, this is definitely something worth exploring in the future, but we have not been able to include these results in our report.

1.4 Research Method

Our methodology focused mostly in looking at the performance of fusion techniques in terms of accuracy, precision and recall. We aim to exhaustively look at the impact early and late fusion has across multiple learning techniques. We seek to not only see if it is possible to have the best predictor improve performance through feature fusion, but also to see what feature fusion affect performance across multiple predictors individually.

To do so, we split our data into training and testing, recorded accuracy, precision, and recall each leaner obtained by using different sets of features. This procedure was repeated for early and late fusion.

The statistics obtained are reported in the experiment section.

1.5 Main Contributions

In this work, we conduct an in depth experimental study of different feature fusion methodologies. Upon thorough experimentation, our results indicate that late fusion tends to improve accuracy, but there is a trade-off on computational requirements.

Our study suggests different scenarios where machine learning practitioners might prefer one technique over the other, and demonstrate what these trade-offs are.

1.6 Outline

The following is an outline of the rest of this thesis:

- **Chapter 2 — Background**

  We introduce the relevance of machine learning and feature fusion in terms of the medical field. We explain how these techniques are making their way into medical diagnosis by drawing example from endoscopy, colonoscopy, and gastroscopy.

- **Chapter 3 — Tools and process**

  We define the tools that will be used through this thesis and explain their applications. LIRE, Weka, and scikit-learn are used for doing data processing and learning probabilistic models. Junyper Notebooks are used for data visualization and creating plots.

- **Chapter 4 — Experiments**

  We define the experiments and datasets we will use to compare feature fusion methodologies. We introduce and define the metrics on which we evaluate these methods: accuracy, precision, recall, F1, and F-beta.

- **Chapter 5 — Conclusion**

  We conclude with a summary of the study we conducted and the insights we obtained from the results.

  We go one step further, and suggest use cases for data scientists considering using early and late fusion methods, and describe the benefits and drawback of each according to the results we obtained.

Chapter 2

Background

2.1 Medical scenario

Since the medical field is diverse, we decided to specifically address the human gastrointestinal (GI) system (figure 2.1) because it is susceptible to various diseases which are visually distinguishable. This choice is also supported by the fact that the most common cancer types are located in the GI tract[2].

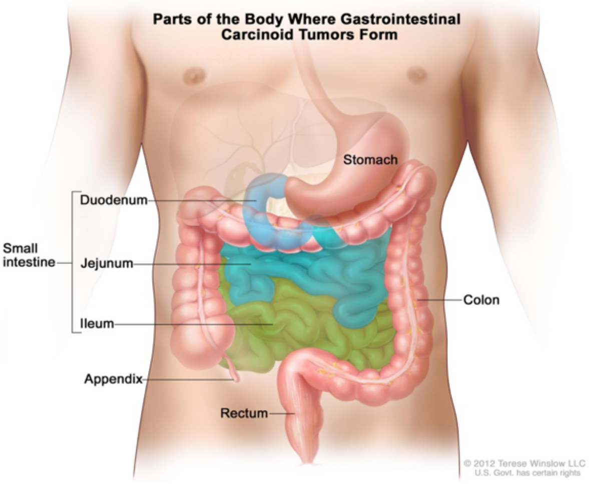

Parts of the Body Where Gastrointestinal Carcinoid Tumors Form
Stomach
Duodenum
Small intestine
Jejunum
Ileum
Colon
Appendix
Rectum
© 2012 Terese Winslow LLC U.S. Govt. has certain rights

Figure 2.1: A complete overview of the human GI tract[1]

The complex GI system can be affected by one of the most common cancers, which is Colorectal cancer (CRC). This is one of the the major health issues worldwide, and it is the third most common cancer after lung cancer and breast cancer[9]. If CRC is detected at an early stage, the prognosis is substantially improved, from a 90% 5-year survival probability in the early stage 1 to only 5-10% 5-year survival probability in the latest stage 4[10]. Several studies have shown that large population-based screening programmes improve the prognosis and even reduce incidences

[1] Figure is taken from: https://www.ncbi.nlm.nih.gov/pubmedhealth/PMHT0022856/?figure=1

of CRC[11], and the European Union guidelines recommend screening for CRC for all persons older than 50 years[12].

2.1.1 Endoscopy and Colonoscopy

GI endoscopies are common medical examinations where the lumen and the mucosa of the entire GI tract are visualized to diagnose diseases [11]. There are two different endoscopic examinations that is used frequently, colonoscopy(where the colonoscope is inserted into the rectum) and gastroscopy (inserted via the mouth). The endoscopic system is made of an endoscope, a flexible tube with a charge couple device (CCD) chip and two bundles of optical fibers at the tip. The endoscope is connected to a video processor and a light source, and the video signals are transferred to a screen for the doctor to analyse. Such endoscopies are demanding and invasive procedures, and can be of great discomfort for patients. They are performed by medical experts (endoscopists), have to be performed in real-time and do not scale well to larger populations due to labor-intensive expert involvement. Furthermore, colonoscopy is not the ideal screening test, many polyps are hard to detect and in average, 20% of polyps are missed or incompletely removed, i.e., meaning that the risk of getting CRC later on largely depends on the endoscopist's ability to detect polyps [13]. The proponents therefore aim for a system that detects endoscopic findings in videos of the GI tract(EIR).

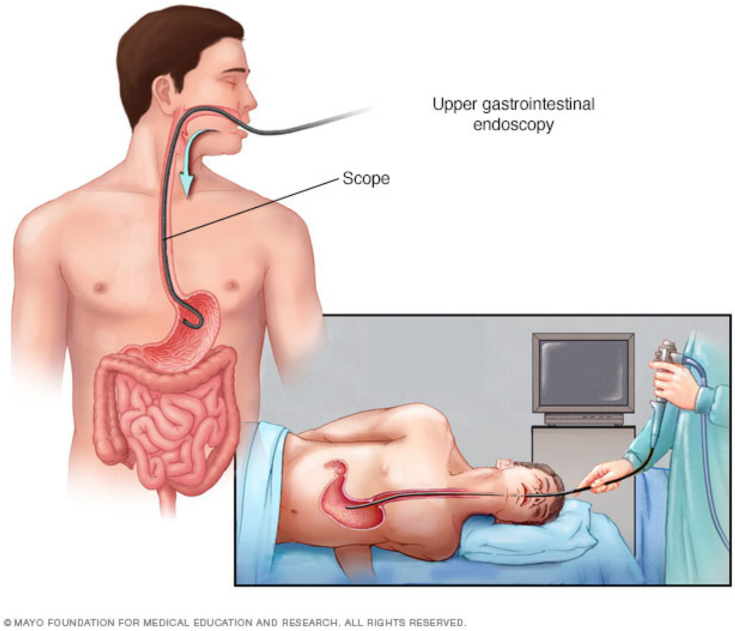

Upper gastrointestinal endoscopy
Scope
© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

Figure 2.2: Endoscopy illustration^2

Once a polyp is detected, the morphology needs to be assessed to determine whether or not the polyp has a risk of malignant transformation. There exist mainly three classification systems for polyp assessment, two for characterization of the surface and one for the shape. The Kudo and the Nice-classification are both used to characterize the surface structure of the polyp. The Kudoclassification[13] is based upon chromoscopy requiring supplementary staining of the mucosa with a colorant, while the Nice-classification[14] is based on electronic color filter on the scope. The Paris classification is used to describe the shape of the polyp[15]. Despite these classifications, endoscopists assess polyps quite differently, and a standard computer algorithm for interpretation may therefore reduce the differences in the assessment[16].

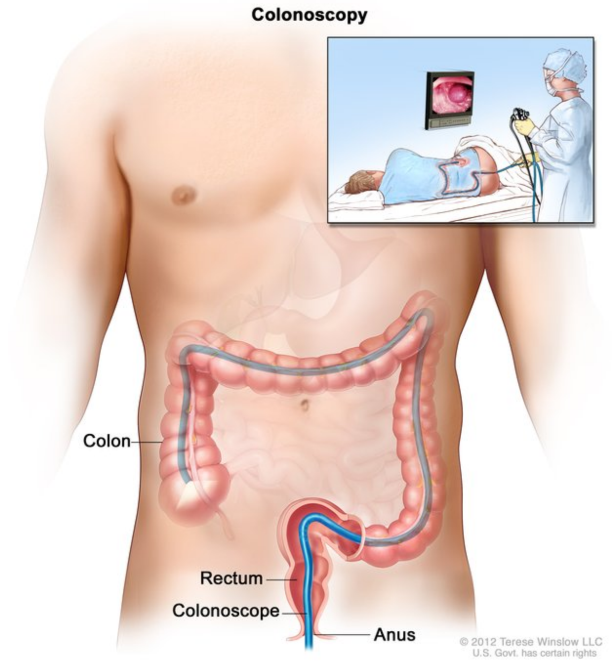


Figure 2.3: Colonoscopy illustration[3]

2.1.2 Gastroscopy

Gastroscopy is also refereed to as an upper gastrointestinal endoscopy. It has a light and a camera at one end called endescope. This is a flexible tube which is inserted via the mouth. The camera transmits images of the internal oesophagus, stomach and duodemum to a monitor.

Gastroscopy can be used to an array of medical purposes like it investigates problems like difficulty in swallowing (dysphagia) or persis-

[2] http://www.mayoclinic.org/tests-procedures/endoscopy/multimedia/endoscopy/img-20007299

[3] https://www.niddk.nih.gov/health-information/diagnostic-tests/colonoscopy

tent abdominal pain; diagnoses various conditions such as stomach ulcers or gastro-oesophagus reflux disease (GORD); and treats conditions such as bleeding ulcers, a blockage in the oespohagus, non-cancerous growths (polyps) or small cancerous tumours.

This procedure takes less than 15 minutes and may take longer if it's being used to treat a condition. The throat will be numbed with a local anaesthetic spray before the procedure. Or, a sedative can be chosen, depending on the patient's preference. With these options, a patient will still be awake, but will be drowsy and have a reduced awareness about what's happening. The doctor undertaking the procedure will place the endoscope in the back of the patient's mouth. It will then be guided down your oesophagus and into the stomach. This procedure is not painful, though it may be unpleasant and uncomfortable at times[3].

2.1.3 Wireless Capsule Endoscospy

The multi-sensor WVC is swallowed in order to visualize the GI tract for subsequent detection and diagnosis of GI diseases. Thus, people will be able to buy WVCs at the pharmacy, and connect and deliver the video stream from the GI tract to the phone over a wireless network. The video footage can be processed in the phone or delivered to the system, which finally analyses the video automatically. In the best case, the first screening results are available within eight hours after swallowing the WVC, which is the time the camera typically spends traversing the GI tract.

The current WVCs have a low resolution of 256 × 256 with 3-30 frames per second (adaptive frame rate with a feedback loop from the receiver to the transmitter). They do not have optimum lighting, making it more challenging to analyze small details in the images. Nevertheless, ongoing work tries to improve the state-of-the-art technology which will make it possible to use the methods and algorithms developed for colonoscopies also for WVCs [19].

Figure 2.4: Endoscopy capsule(Pillcam)[4]

In the case of the colon, accuracy of existing methods is far below the required precision and recall, and the processing of the algorithms does not scale in terms of big data. Each type of disease or irregularity requires interaction between medical researchers dictating what the system must learn to detect, image processing researchers investigating detection or summarization algorithms, hardware developers to develop/produce/research sensors, and distributed processing researchers in order to scale the big data analytics of the sensor data. For other scenarios, like in the upper part of the GI tract, there will be similar challenges and corresponding interaction between research disciplines. There are large challenges with respect to accuracy (precision and recall), scale of the processing and hardware data quality because of different manufacturers.

2.1.4 Automated Computer diagnosis

A possible solution to achieve real-time instead of near real-time performance is the SAPPHIRE middleware and software development kit for medical video analysis[17]. The toolkit has been used to build the EM-Automated-RT software[18]. EM-Automated-RT does real-time video analysis to determine the quality of a colonoscopy procedure, and it is able to give visual feedback to the endoscopist performing the procedure.

This is done to achieve optimal accuracy of the inspection of the colon during the procedure. Nevertheless, it is limited to the assessment of the endoscopist's quality, and does not automatize disease detection itself.

A dominant trend to speed up processing of CPU-intensive tasks is to offload processing tasks to GPUs. Stanek et al.[17][18] indicate that utilizing a GPU can be the right way to achieve real-time performance. In other areas this has already been explored to a certain extent. For example, some do applied it in sport technology[19][20], where GPUs were used to improve the video processing performance to achieve live, interactive panning and zooming in panorama video.

In summary, actual computer-aided diagnostic systems for the GI tract do not provide real-time performance in combination with a sufficient detection or localization accuracy.Therefore, the proponents present a system focusing on both high accuracy detection and real-time performance.

2.2 Related Work

The proponents of this thesis are researching feasible solutions to maximize the capability of the EIR system by fusing methods like early and late fusing. Such integration is to increase EIRs performance. The processes and sub-tasks that have undertaken to complete this endeavor are acquisition, processing, and annotating of video sequences. This section deals with relevant samples of the most interesting and relevant existing research

[4] http://agimedical.com/wp-content/uploads/2015/04/pillcam.jpg

and experimentation related to the tasks of this project. The proponents collected and analyzed information that can be used in creating an effective GI diseases diagnosis.

Liu et al.[21] describe a very advanced annotation tool called Arthemis. Arthemis is part of an integrated capturing and analysis system for colonoscopy, called Endoscopic Multimedia Information System (EMIS). EMIS provides functionality for collecting and archiving endoscopy videos. The use of an annotation tool for endoscopy videos is further researched by Lux and Riegler[8]. This demo paper focuses on common interaction methods for experts to annotate videos by recording speech and drawing onto the video. The paper aims at gathering information about the recorded videos in an easy and simple way, so that the annotation effort is minimally invasive for the daily routine of the experts. The related work in the field of annotation shows that it is crucial to integrate the annotation tool in a minimally invasive way within the environment of the experts. It is very important to provide them with a solution which is very easy to use and, at the same time, very easy to deploy in a restrictive medical environment. The annotation subsystem in EIR builds up on technologies and methods from[22]and[23] to reach optimal annotation performance.

Automatic analysis systems for the GI tract. Detection of diseases in the GI tract has mostly focused on polyps. This is most probably due to the lack of data in the medical field and polyps being a condition with at least some data available. Automatically analysis of polyps in colonoscopies has attracted research attention for a long time and several studies have been published [28]. However, there is a need for complete scalable real-time detection systems, both for computer aided diagnosis during colonoscopy examinations and for analysis of huge amounts of data from VCEs. Furthermore, all of the related works are limited to a very specific use-case, which in most cases is polyp detection for a specific type of camera. Several algorithms, methods and partial systems have been proposed and have achieved at first glance promising results in their respective testing environments. However, in some cases, it is unclear how well the approach would perform as a real system used in hospitals. Most of the research conducted in this field uses rather small amounts of training and testing data, making it difficult to generalize the methods beyond the specific dataset and test scenarios. Therefore, overfitting for the specific datasets can be a problem and can lead to unreliable results. The approach from Wang et al. [29] is the most recent and best working one in the field of polyp detection.

Mamonov et al.[24] presented an algorithm for a binary classifier to detect polyps in the colon. The method is called binary classification with pre-selection, and it aims at reducing the amount of frames that need to be manually inspected. The sensitivity of the algorithm with regards to single input frames is significantly lower and only reaches 47%.

The best working and complete system in the well-researched polyp detection field is Polyp-Alert[25], which is able to give near real-time feedback during colonoscopies. The system can process 10 frames per second and uses visual features and a rule based classifier to detect the edges of polyps. Further, they distinguish between clear frames and polyp frames in their detection. The researchers report a performance of 97.7% correctly detected polyps based on their dataset which consists of 52 videos recorded using different colonoscopes. Unfortunately, the dataset is not publicly available, and therefore, an exact detection performance comparison is not possible. Compared to EIR system, this system seems to reach higher detection accuracy, but it appears that EIR system is faster in terms of processing time per frame and can therefore detect polyps in real-time. A comparison using the same hardware and full length videos is currently to be carried out together with the developers of Polyp-Alert. Furthermore, EIR system is not designed and restricted to detect only polyps, and can be expanded to any possible disease if there was a correct training data.

Deep learning is probably the most promising approach everybody needs to explore further in EIR, and it is already very relevant for similar problems detecting for instance breast cancer[26], polyp detection[27]or lung cancer[28]. Nevertheless, such approaches are challenging to use in EIR use-case. First, training is very complicated and time-consuming. EIR system has to be fast and understandable since it deals with patient data, where the outcome can differentiate between life and death. This can lead to serious problems in the medical field since it is very difficult to evaluate them properly[29]. Furthermore, one of the biggest challenges is that they require most of the time a lot of training data. In the medical field, this is a very important issue since it is hard to get data due to the lack of experts time (doctors have a very high workload) and legal and ethical issues. Some common conditions, like colon polyps, may reach the required amount of training data for deep learning while other endoscopic findings, like for example tattoos from previous endoscopic procedures (black colored parts of the mucosa), are not that well documented, but still interesting to detect[30]. Nevertheless, for certrain use-cases, such as presentend in [26] also a small amount of training data can lead to reasonable results. **Tajbakhsh et al. [27]** presented a combined algorithm for a binary classifier to detect polyps in the colon, which was trained and tested on a 35, 000 frames dataset with only 20 different polyps. The proposed polyp detection method first selects multiple possible polyp locations in a frame using machine learning of local polyp features such as color, texture, shape, and temporal information in multiple scales. A generated set of locations is then processed by a number of convolution feature specialized neural networks and followed by results aggregation and frame binary classification. The detection performance of the method is 0.002 false positive per input frame at 50% sensitivity.

Late Fusion. Also known as decision fusion has features which are managed by different classifiers. These classifiers are used in sorting or labeling results. Such process is a part of the first classification steps and the output of each classifier is combined to acquire an accurate concluding result. Since each feature is administered by a separate classifier, late fusion is very pricy in terms of learning effort. Furthermore, to combine the pre-classified features one or more additional classifiers are needed. Another challenge is the possible loss of information that comes naturally if different features are combined[31]. The combination of the output of the pre-classifiers is a very important step and can be performed in different ways. Which method is the best depends on the dataset, the features that are used and the metrics that are used to calculate the distances between the different features[6]. In [31], the authors try to learn tag relevance for social image retrieval by using multiple features in an unsupervised learning environment. They state that visual content cannot be discovered by using only one visual feature. To prove this claim, they tested different approaches of feature combination methods to find the best one for their task. The second one is called UniformTagger, which is a combination of different base learners that are combined in a uniform way. The conclusion is that learning using a combination of different features can outperform single-feature-based learning systems. The second observation is that late fusion approaches lead to a better performance than early fusion approaches [32][33]. Based on the indication that late fusion is better suited for multimedia data, we decided to use it for feature combination in our work. We are going to classified each feature and use them separately, and combine them afterwards.

2.3 EIR

According to the study of Riegler et al.[6] conducted in 2016, the main objectives of the EIR system are (i) easy to use, (ii) easy to extend to different diseases, (iii) realtime handling of multimedia content, (iv) being able to be used as a live system and (v) high classification performance with minimal false negative classification results. It can be split into three main parts: the annotation sub-system, the detection and automatic analysis sub-system, and the visualization and computer aided diagnosis sub-system. All three parts are important to achieve a holistic system that can support doctors in disease detection and diagnosis in the GI tract. [6]

The Annotations Subsystem

The main purpose of this subsystem is to collect training data for the detection and automatic analysis subsystem. In the medical field, medical personnel have little or no access to multimedia data, and they don't have the time to make use of these approaches. This is mainly due to legal issues and their daily workload. Most patients are also reluctant to carry out this procedure. Moreover, these video procedures are very time-consuming, and the quality of the annotation outcome depends largely on the expertise and dedication of the medical personnel. [20]. For example, the VCE procedure has about 216,000 images per examination, which could take an experienced endoscopist about 1 hour to view and analyze the entire data contained in the video [21]. Aside from getting data to enable automatic screening from the EIR system, the annotation subsystem makes it feasible to use an annotated video in a medical video archive for documentation of the procedures or even for teaching purposes.

Detection and Automatic Analysis Subsystem
This subsystem automatically collects, analyze and localize findings in the part of the body being studied. It is designed in a modular way so it can be extended to different diseases or their subcategories. The detection part analyzes videos or images to find out if there is anything abnormal to be found, then there is the localization part which determines the location of the detected malady. It uses the results of the detection as its input.

The Visualization and Computer-aided diagnosis Subsystem
The primary purpose of the visualization subsystem is to provide or display the results of the automatic detection and analysis subsystem to medical experts who will then use the output for proper diagnosis using computer-aided diagnosis. This tool uses the output of the localization and system detection part to create a web-based visual which makes use of a video sharing platform that enables doctors to watch videos, archive them, annotate, and share information.

2.4 Summary

There have been so many contributions, studies, and innovations undertaken for the sake and progress of computer-aided diagnostic tools in the field of medicine. It is an indication of men's ever changing survival technique in the ever evolving world. EIR system is one of the many things devised for diagnosis of internal organs, specifically in the GI tract. The first parts of this paper talked about certain progresses in this system and what the proponents of this thesis see as a potential contributor to the expansion of EIR's capabilities. Early and late fusions are to be added to the EIR. And the best thing to work out is which of the two is best? Both have different things to offer that will be essential to the evolution of the EIR system.

Part II

Tools and Implementation

Chapter 3

Tools and Processes

This chapter presents the tools that's been used throughout this study. Different softwares and tools has been utilized to reach our goal. This contains the various stages of testing the combinations of features, extracting the features and fusing them.

Table 3.1: Tools and software used

Software	**Name**	**Version**
Visual information retrieval	LIRE [1]	1.0b4
Data mining	Weka[2]	3.8
Machine learning library	Scikit-learn[3]	0.18.1
Data Manipulation	Pandas[4]	0.20
Tool	**Name**	**Version**
Notebook	Jupyter[5]	4.1.1

3.1 Extracting Features

The major component of content based image retrieval is feature extraction. Feature extraction is done using colours, textures or shapes. For texture extraction, structural, statistical and spectral approaches are used. The features that's used are already implemented in LIRE. LIRE provides us with 30 features. For color feature extraction, color histograms like Local Color Histogram (LCH), Fuzzy Color Histogram (FCH) and Global Color Histogram (GCH) are used.

The extracting of the features is done in LIRE(Lucene Image REtrieval), a powerful tool for extracting visual features from bitmap images and storing them in a Lucene for later retrieval[8]. LIRE is an open-source java-based Library that's used for content based image retrieval. This type

[5] http://www.lire-project.net

[5] http://www.cs.waikato.ac.nz/ml/weka/

[5] http://scikit-learn.org

[5] http://pandas.pydata.org/

[5] http://jupyter.org

of software attracts developers and researchers because of its simplicity and a good overview. Numerous types of features are already provided by LIRE, one can also develop features and use the search engine that's offered by this software. This allows more focus on developing the feature itself, which saves a lot of time. LIRE provides in this case everything from parallel indexing, hashing and linear search.

Currently the following image retrieval features are included in LIRE:

1.MPEG-7 descriptors scalable color, edge histogram and color layout[8].
2. Color histograms in HSV and RGB color space.
3. Tamura texture features coarseness, directionality and contrast[34].
4. Color and edge directivity descriptor, CEDD[35]
5. Auto color correlation feature as defined by Huang et al.[36]
6. Fuzzy color and texture histogram known as FCTH[37]

3.2 Features

In machine learning and pattern recognition, features are individual measurable properties of a phenomenon being observed[34]. Features are characteristics, variables or observable phenomenon that can capture a given visual property of an image locally for objects or regions, or globally for the entire image, and can then be quantified and recorded. The success of machine learning depends largely on finding the right features. In some cases, features might be explicit in the input, while in others, you may have to come up with a method of feature extraction. For a domain like CBIR (content-based image retrieval), development of useful features is a research area unto itself.

3.2.1 Global features

Global features also known as global descriptors are generally used in image retrieval, classification, and object detection. Global descriptors detect if an object is present in an image or video. Global features generalize and describe the entire image with single vector, and they include contour representations, texture features, and shape descriptors. Co-HOG and Histogram oriented gradients (HOG) are examples of global features. Global features are used for low-level application and where rough segmentation of an object is available.

Joint histogram

According to Stricker et al. (1995)[38], color histograms are commonly used for content-based image retrieval due to their efficiency and robustness. However, a color histogram only registers an image's general color configuration, so images with very different looks can have similar color histograms. This problem is especially precarious in huge image databases,

Table 3.2: The different global features that is provided by LIRE and supported by EIR. All of them have been extracted and used in this project

Feature	**Dimensions/bins**
Joint Histogram	576
Jpeg Coefficient Histogram	192
Tamura	18
Fuzzy Opponent Histogram	576
Simple Color Histogram	64
Fuzzy Color Histogram	125
Rotation Invariant LIBP	36
FCTH	192
Local Binary Patterns and Opponent	288
PHOG	630
Color layout	33
CEDD	144
Gabor	60
Opponent Histogram	64
Edge histogram	80
Scalable color	64
Rank And Opponent	576
JCD	168

where many images have the same color histogram. Some professionals suggest a substitute to color histograms called a joint histogram, which includes additional information without sacrificing the robustness of color histograms. The proponents created a joint histogram by selecting a set of local pixel features and creating a multidimensional histogram. Each entry in a joint histogram comprises the number of pixels in the image that are described by a certain combination of feature values. Stricker et al. (1995)[38] define a number of different joint histograms, and assess their performance for image recovery on a database with over 210,000 images. Their standards joint histograms beat color histograms by an order of scale.

Jpeg coefficient histogram

With JPEG coefficient, the DC and AC coefficients of JPEG images are encrypted using a stream cipher and scrambling encryption, respectively. Then, the encrypted images are transferred to and kept in a server, which can also offer recovery service. When accepting an encrypted query image, the server deprived of any knowledge of the plaintext content may obtain statistically its AC coefficients histogram. By computing the distances between the histograms of encrypted query image and database image, the server may produce the coded images closest to the query image to the authorized user[39].

Tamura

Texture is a very vital feature particularly used in numerous image handling problems. People are used to some texture based perceptual structures to decide between textured images or regions. These Perceptual structures are extremely needed for two reasons; they will be best in terms of feature selection and will be appropriate to all kinds of textures. Some of the significant perceptual structures are coarseness, contrast, directionality and busyness. Tamura proposed a new perception-based approach to content-based image classification and retrieval. The proposal is based on multiple representations: Original Image Representation and Autocorrelation Function Representation. The computational measures for textural features are calculated both on original and auto-correlated images. In order to authenticate these features measures, applied them for texture classification and retrieval on brodatz images. For texture classification, features calculated on Multiple representation correctly categorized the best identical class among the remaining class in contrast with original representation based features and autocorrelation representation based features. K-Nearest Neighborhood classifier is used for this classification task. For texture retrieval, Multiple representation based features salvaged more number of appropriate images in comparison with features derived from autocorrelation representation. Gower co-efficient of resemblance is used to find the feature similarity between images in retrieval task. Thus this work achieved good classification rate of 93.5% and better retrieval rate by using these estimated features on our approach[40].

Fuzzy opponent histogram

Image category recognition, such as fuzzy color histogram, is significant to access graphical information on the level of items and scene types. The intensity-based descriptors have been widely used to increase brightness invariance and discriminative power. As numerous descriptors be, a structured indication of color invariant descriptors in the context of image category recognition is mandatory. The uniqueness of color descriptors is measured experimentally using two benchmarks from the image and the video domains[41].

Simple color histogram

According to Rosebrock (2014)[42], a simple color histograms by definition disregard both the shape and texture of the object(s) in the image. This means that it does not have the concept of the shape of an object or the texture of the object. Also, histograms also ignore any spatial information like where in the image the pixel value came from. An extension to the histogram, the color correlogram, can be used to encode a spatial relationship among pixels.

Fuzzy color histogram

A conventional color histogram (CCH) is sensitive to noisy interference such as illumination changes and quantization errors. Also, CCHs' large aspect or histogram bins require large calculation on histogram comparison. To address these concerns, Han and his team presented a new color histogram representation, called fuzzy color histogram (FCH), by considering the color similarity of each pixel's color connected to all the histogram bins through fuzzy-set membership function. A original and firm approach for calculating the membership values based on fuzzy c-means algorithm is presented. The proposed FCH is further subjugated in the submission of image indexing and recovery. Experimental outcomes obviously show that FCH produces better retrieval results than CCH. Such computing methodology is justly necessary for image retrieval over large image databases[43].

Rotation invariant local binary pattern

Rotation invariant local binary pattern is based on distinguishing that certain local binary patterns called 'uniform' are important assets of local image texture, and their manifestation histogram shows to be a dominant texture feature. Ojala et al. (2002)[44] derived a comprehensive gray scale and cycle invariant operator demonstration that allows for identifying the 'uniform' patterns for any quantization of the angular space and for any spatial resolution, and present a process for merging multiple operators for multiresolution analysis. The proposed method is very robust in terms of gray scale variations, since the operator is by description invariant against any monotonic transformation of the gray scale. Another benefit is computational simplicity, as the operator can be realized with a few operations in a small neighborhood and a lookup table. Outstanding experimental outcomes gained in true difficulties of rotation invariance, where the classifier is qualified at one specific rotation angle and tried with samples from other rotation angles, validate that good judgment can be attained with the occurrence statistics of simple rotation invariant local binary patterns.

Fuzzy color and texture histogram (FCTH)

Chatzichristofis et al. (2008)[37] proposed a paper that bdeals with the extraction of a new low level feature that combines, in one histogram, color and texture information which was termed Fuzzy Color and Texture Histogram. FCTH resulted from the combination of three fuzzy systems. FCTH size is restricted to 72 bytes per image, rendering this descriptor suitable for use in large image databases. The proposed feature is appropriate for accurately retrieving images even in distortion cases such as deformations, noise and smoothing. It is verified on a large sum of images selected from proprietary image databases or randomly retrieved from common search engines. To assess the performance of the proposed feature, the averaged normalized altered retrieval rank was used.

Local binary patterns and opponent histogram

Texture is a vital feature for image examination, classification or segmentation[45]. Since more and more image segmentation problems involve multi-and even hyperspectral data, it turns out to be essential to define multispectral texture features. It was suggested that a natural extension of the classical Local Binary Pattern (LBP) operator to the condition of multispectral images. The Local Multispectral Binary Pattern (LMBP) operator is based on the report of total collections in the multispectral image space and on an extension of the typical univariate LBP. It permits the calculation of both a multispectral texture structure coefficient and a multispectral difference parameter circulation. Outcomes are verified in the case of the segmentation of brain tissues from multispectral MR images, and compared to other multispectral texture features.

Pyramid histogram of oriented gradients (PHOG)

Bosch et al. (2007)[46] proposed the Pyramid Histogram of Oriented Gradients (PHOG). It is designed to classify images by the object categories they hold. It has three areas. First, introduce a descriptor that signifies local image shape and its spatial design with a spatial pyramid kernel. These are planned so that the shape corresponds between two images can be measured by the distance between their descriptors using the kernel. Second, generalize the spatial pyramid kernel and study its level weighting parameters (on a validation set). This considerably progresses classification performance. Third, show that shape and appearance kernels may be united (again by learning parameters on a validation set).

Color layout

MPEG-7, formally known as Multimedia Content Description Interface which includes standardized tools facilitating structural, thorough descriptions of audio-visual information at diverse granularity levels and in different parts. It aims to support and facilitate a wide range of applications, such as media portals, content broadcasting, and ubiquitous multimedia. Chang et al. (2001)[47] presented a complex overview of the MPEG-7 standard. It was first deliberate the scope, basic terminology, and potential applications. Next, discussion of constituent components and compare the relationship with other standards.

Color and edge directivity descriptor (CEDD)

Chatzichristofis et al. (2008)[35] proposed the color and edge directivity descriptor (CEDD) that deals with a fresh low-level feature that is extracted from the images and can be used for indexing and recovery. It incorporates color and texture information in a histogram. CEDD size is restricted to

54 bytes per image and changing this descriptor fit for use in large image databases. One of the vital attributes of the CEDD is the low computational power desirable for its extraction, in contrast with the requirements of most MPEG-7 descriptors.

Gabor

Gabor filters have been effectively useful to a wide range of image handling jobs. Weldon et al. (1996)[48] studies the design of a single filter to segment a two-texture image. An original efficient algorithm for Gabor-filter design is presented, along with approaches for estimating filter output statistics. The algorithm lures upon previous results that showed that the output of a Gabor-filtered texture is modeled well by a Rician distribution. The total amount of the output power is used to choose the center frequency of the filter and is used to guesstimate the Rician statistics of the Gabor-filtered image. The method is added generalized to include the statistics of post-filtered outputs that are generated by a Gaussian filtering process succeeding the Gabor filter. The new method typically needs an order of magnitude less calculation to design a filter than a previously proposed method. Experimental outcomes prove the efficacy of the method.

Opponent histogram

The opponent histogram according to Transactions on Data Hiding and Multimedia Security V is a 3-D histogram based on the channels of the opponent color space YCbCr. This color space was designed to the color models of the first two channels are shift-invariant with respect to light intensity. The third channel possesses intensity information and has no invariance properties.

Edge histogram

To form the histogram, just simply concatenate the measurements into one long vector. To do the concatenation is acceptable as long as you keep track of the way you map the bin/direction combo into a slot in the 1-D histogram. This long histogram of concatenations is then most often used for machine learning tasks, like training a classifier to recognize some aspect of images based upon the way their gradients are oriented.

Scalable color

The Scalable Color Descriptor is derived from a color histogram defined in the Hue-Saturation-Value (HSV) color space with fixed color space quantization. It uses a Haar alter coefficient encrypting, permitting scalable image of description, as well as complexity scalability of feature extraction and matching actions.

Joint composite descriptor (JCD)

Compact Composite Descriptors (CCDs) is set of low lever features that can be used to describe various types of multimedia information. The set contains descriptors for three different types of images. A group of two descriptors (CEDD and FCTH) combines color and texture information in order to describe the visual content of the real-world color images. At the same time, another descriptor (BTDH) combines brightness and texture characteristics in order to describe the visual content of grayscale images (as well as radiology medical images). The recently proposed SpCD combines color and spatial distribution characteristics to describe artificially generated images (computer graphics, color sketches etc.)[49].

3.2.2 Local features

Local features are a pattern or distinct feature seen in an image. These are generally used for object recognition and identification. Local descriptors recognition of image involves finding the identity (i.e.) recognizing a person or an object in an image. Local features describe image patches (critical points of the image), points, edge, and texture. They are associated with image patches that differ from their surrounding by colour, intensity or texture. The actual representation of the entire image is insignificant in local feature extraction. Local features describe images with multiple points on the image thus making them more robust. SIFT, HOG, SURF, LBP, BRISK, FREAK and MSER are good examples of local features. Local features are generally used for high application such as object recognition.

With local features, you can find image correspondence undermining presence of clusters, changes in view conditions, or occlusions. The properties of local features also make them suitable for classification. The combination of local and global features improves the accuracy of recognition with the side effect being computational overheads. Feature detectors and descriptors are selected by considering the requirements of your application, and nature of the data.

3.3 Fusion methods

EIR supports various global features and their combination. When used in different scenarios, different combinations of the supported features would lead to more accurate results than those gotten in others. There has been a non-trivial problem which has posed as a research topic over the years, and this is the combination of various features which contain features spaces with different dimensions and sizes.

A sophisticated and unique combination of features can lead to more accurate classifications and search results in a system. However, feature combination does not come without its pitfalls. These pitfalls if not taken into proper consideration can lead to a decline in the performance of the

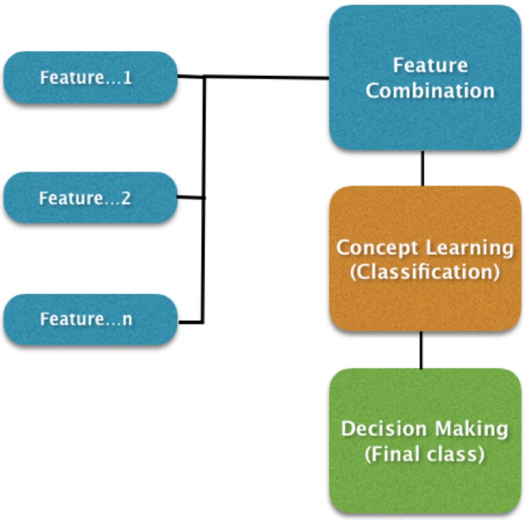

Feature...1
Feature...2
Feature...n
Feature Combination
Concept Learning (Classification)
Decision Making (Final class)

Figure 3.1: Pipeline of early fusion

system [50][51][52]. Features can be fused in two different ways: Early fusion- also called feature, and decision or late fusion.

3.3.1 Early fusion

Early fusion fundamentally fuses values of different features into a single representation before they are used in a decision-making steps. The early fusion combines uni-modal features after being extracted, into uni-modal representation. This is one method. Another method is to combine or join all features with one long feature vector, but at the same time applying features selection methods such as principal component analysis. This approach reduces a large set of variables to a smaller, conceptually more coherent set of variables which contain the same information as the original larger representation. Applying this method, however, can lead to a reduced length of the vector. Another problem is that the combination of very diverse features can be challenging. An example is, attempting to combine an audio feature which has several hundred dimensions with a video feature which has several thousands of dimensions. In this case, it is advisable to the trio of feature selection, features reduction, and normalization - on the data before fusion.

3.3.2 Late fusion

In late fusion, each feature is processed by its classifier. In other words, extracted features are processed separately and only the results are combined. (The output of each classifier is combined to arrive at a final result) Because each feature is processed in a different classifier, decision fusion (another name for late fusion) is very costly regarding learning efforts. The combination of the output of the pre-classifiers is a critical step,

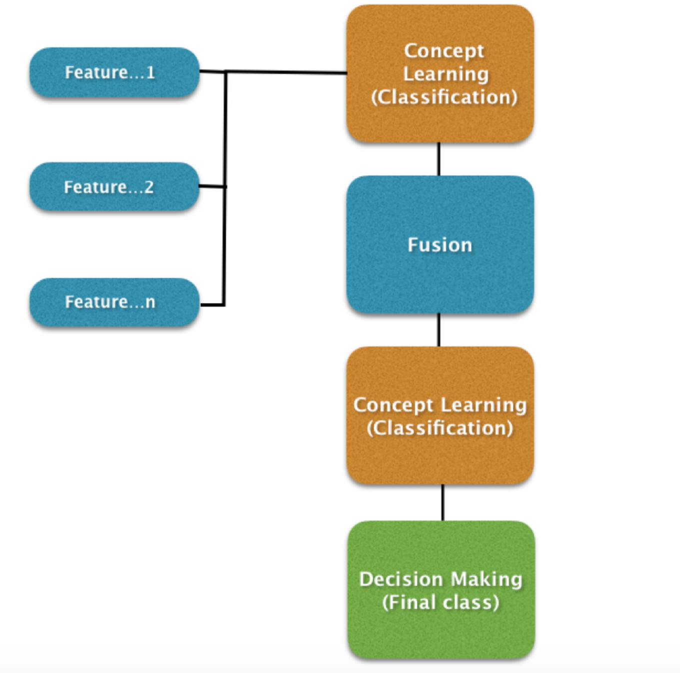

Feature...1
Feature...2
Feature...n
Concept Learning (Classification)
Fusion
Concept Learning (Classification)
Decision Making (Final class)

Figure 3.2: Pipeline of late fusion

and this can be performed in different ways. The best method depends on the following:

(i) The dataset

(ii) The features that are used, and

(iii) The metrics used to calculate the distance between the features.

Late fusion approaches have been found[31][53] to have a better performance than early fusion approaches.

3.3.3 Scikit Learn

For the machine learning algorithms and the metrics used to evaluate them, we use a popular library called Scikit-learn. Scikit-learn is a machine learning library used for Python programming. It involves various regression, clustering, and classification algorithms. In regression, it is used for predicting the continuous-valued attribute of an object. In clustering, it automatically groups similar objects into sets and is applied in the grouping of experimental outcomes, and for customer segmentation. As for classification, it identifies what category an object belongs to, and it is implemented in spam detection and image recognition. Scikit Learn is a straightforward and efficient tool for data mining and analysis which is reusable in various contexts and it is built to operate with Python and scientific libraries NumPy and SciPy.

3.4 Data mining

In the starting phase of this project a software named Weka was utilized. Weka is a set of machine learning visualization tools and algorithms used for predictive modeling and data analysis. Weka consists of tools which support graphical user data mining interphase i.e. it is an Explorer for exploratory data analysis that support classification, regression, learning, attribute selection, data pre-processing, association rules, clustering, feature selection, visualization, and Knowledge Flow for new process. The algorithms are either applied directly to a dataset or called from a java code. Weka software was used at the beginning of this project to gain an understanding of how different machine learning algorithms work. All Weka techniques are applied on the assumption that data is present as a flat file or relation.

Advantages of Weka

- It is suitable for developing new machine learning operations.
- Weka is best fit for mining association rules
- It loads data file in formats of CSV, ARFF, C4.5, and binary. Although it is open source, free, and extensible, it can be integrated into other Java packages.

3.4.1 Jupyter Notebook

The implementation of the python code is done in Jupyter notebook. A lot of scientific institutions are using these notebooks in order to clearly explain how they got the results, and not only can these notebooks show us how they got the results, but we can reproduce the results within the notebooks them self's. Jupyter is a web-based notebook or an open source web application tool with which you can share and create documents containing equations, live codes, mathematical equations, visualizations, and explanatory text. Jupyter is used for numerical simulation, data cleaning, transformation, statistical modelling, and machine learning among others. These documents contain a complete self-contained record of computation that is convertible to various formats.

The Jupyter notebook has an interactive notebook web application which is used for writing and running codes interactively as well as authoring notebook documents. It also consists of notebook documents which are self-contained documents that represent all the contents visible in the notebook web application- this includes narrative text, images, rich media representation of objects, equations, and inputs and outputs of the computations (each notebook document has a separate kernel). Lastly, Jupyter has Kernels which are separated process Initiated by the notebook web application. Kernels run user-codes in a given language and return output to the notebook web application; kernel also handles computations for interactive widgets, introspection, and tab completion.

3.4.2 Pandas

Pandas is a popular data analysis and manipulation python library. We used pandas to load, store and combine the different feature sets, especially during early fusion.

3.5 Summary

This chapter described the tools used throughout this study. We elaborate on the concepts of early and late fusion, and provide a concise description of the features obtained from LIRE that formed part of our study. We explain our usage of Weka as a data mining component of our pipeline, and our usage of Jupyter Notebooks to provide an simple method for visualizing results.

Chapter 4

Experiments

In this work, we are interested in assessing the quality of features extracted by LIRE. In particular, we would like to contrast the quality of predictions obtained by pre-processing input data using early fusion and late fusion. Early fusion refers to the concatenation of vectors obtained by different feature extraction methods, into a larger single feature vector that describes the data. Learning of a classifier or a regression model is done using this larger input, which is assumed to contain more descriptive information than using a single feature extraction method.

In contrast, late fusion learns a single feature for each vector obtained from different feature extraction methods. One could think of it as learning predictive models for each vector and using their outputs to construct another vector which is used as the input to the final predictive model.

To compare the quality between early and late fusion, we created a pipeline to automate experimentation with a predetermined set of classifiers. We evaluate early and late fusion on combinations of features extracted by LIRE on image datasets, and report their respective performance.

It is worth noting that the classifiers used for experimentation contain hyperparameters that should be optimized to obtain peak performance. In this project, however, due to time constraints we opt for settings hyperparameters to a default value, and leave their optimization as future work.

4.1 Datasets description

For our experiments we used two different image datasets: ASU-Mayo Clinic Colonoscopy dataset[27] and MMSys dataset[54].

ASU-Mayo contains around 36,483 images of GI tract taken during colonoscopy exams. The task is a binary classification of trying to identify images with showing the presence of polyps (positives) and absence of polyps (negatives).

MMSys is a dataset part of the ACM Multimedia Systems Conference Dataset Archive, consisting of a total of 4018 images split evenly into 8 classes.

Table 4.1: ASU dataset in divided in training and test set, and it is again divided into negative pictures and positive pictures(polyp).

ASU Mayo Clinic polyp			
Train		Test	
Neg	Pos	Neg	Pos

Table 4.2: MMSys dataset is divided into train and validation set, both are divided into 8 classes

MMSys	Train	esophagitis
		inked-lifted-polyps
		inked-resection-margins
		normal-cecum
		normal-pylorus
		normal-z-line
		polyps
		ulcerative-colitis
	8Validation	esophagitis
		inked-lifted-polyps
		inked-resection-margins
		normal-cecum
		normal-pylorus
		normal-z-line
		polyps
		ulcerative-colitis

We used Lucene Image Retrieval (LIRE)[8] library to extract several different sets of features for each dataset and trained a classifier for each independent set of features to obtain baseline metrics for later comparison.

4.2 Method and metrics

To evaluate the relative performance between early fusion and late fusion, we selected a set of standard performance metric in the machine learning community. These distinct performance evaluations are meant to indicate the strength and weaknesses of each approach:

4.2.1 Accuracy

This metric simply refers to the ration of correctly classified images over the total number of images. Let Nk be the number of elements for class K. If for a given input xi with true label yi, our model f(X) predicts one of the K possible classes, then accuracy is determined by:

$$Accuracy = \frac{\sum_{i=1}^{N} I(f(x^i) = y^i)}{\sum_{k}^{K} N_k}$$

Table 4.3: Name and description

Name	Description
True Positive(TP)	Refers to the actual hit. When the output is 1 and prediction is 1 too
True Negative(TN)	Refers to actual rejection. When the output is 0 and the prediction is 0 too
False Positive (FP)	Equivalent to false alarm. When the output is 0 but the predicted value is 1
False Negative(FN)	Refers to actual miss. When the output is 1 and the predicted value is 0.

Correct prediction includes the case of 00 and 11 i.e true positive and True negative. Therefor the accuracy can be in our case defined as following:

$$Accuracy = \frac{TR + TN}{TP + TN + FP + FN}$$

4.2.2 Precision

This evaluation metric is mostly used for binary classification. It determines the ratio of true positives (TP) to the sum of TP and false positives (FP). This metric is determined by the equation.

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

4.2.3 Recall

Much like precision, recall is primary used for binary classification. It determines the ratio of TP to the sum of TP and false negatives (FN).

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

4.2.4 F1 Score

This measure attempts to capture a representative number of the models performance with respect to precision and recall. It is given by the harmonic mean of precision and recall.

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

4.2.5 F-beta

The F1 measure has encountered criticism because it introduces bias when dealing with unbalanced data classes. The Fbeta measure is a generalization that allows to weight precision and recall differently.

$$F1 - beta = (1 + \beta^2) \frac{Precision * Recall}{\beta^2 * Precision + Recall}$$

This is a metric that can be useful for our project. With the weighted F1 score, we can give more importance to the precision or the recall with the beta parameter.

The question about if we want to detect most or all instances of a class(higher recall) at the risk of having a higher percentage of false positives (lower precision) is extremely important. This F-beta score could capture this information for us if we adequately define the trade-off. Basically what we want is not to miss any true positives, but at the same time keep the false alarms low.

We added the F-beta score(with beta=2) to the metrics.

4.2.6 Specificity

Also called the true negative rate, measure the proportion of true negatives (TN) that are correctly identified as such. That is the ratio of TN to the sum of TN and false positives (FP).

$$Specificity = \frac{TN}{TN + FP}$$

4.2.7 Metrics for multiclass problems

Accuracy can be used as metric for multiclass classification as it is, but the other metrics (precision, recall and the f scores) need some modifications because they are defined for binary classification only. There are two main ideas: Micro averaging means that is necessary to define true positives, true negatives, false positives and false negatives in a class by class basis, and the metrics are calculated by counting the totals. Macro averaging means that the metric is obtained by successively defining one class as positive and the rest as negatives, obtaining the metric using that definition, and finally averaging all the metrics obtained (one by class).

We used macro averaging to obtain precision, recall and the f scores for the MMSys dataset[54] (which contains 8 classes) and the standard definition for the ASU-Mayo Clinic Colonoscopy dataset[27].

4.3 Classification Algorithms

The following is a presentation of the classification algorithms that was used to obtain the results.

4.3.1 Random forest classifier

It is an ensemble learning for regression and classification purpose[55]. It is a bagging based algorithm where a random set of features along with random sample subset of the training set are selected and used to train a number of (usually) shallow decision trees. The nodes position (or split) in the decision tree is decided by the gini gain or information gain. After the training is complete we have a 'forest' of the trained decision trees. The test data is applied on the decision trees and then voting is performed to obtain the final class from all the decision trees [A]. The important parameters for implementing Random Forest from sklearn are:

1. n-estimators : The number of decision trees to be trained (default: 10)

2. criterion : i.e. whether we want to use gini gain (default) or information gain.

3. max-features : It is the number of features to see for the best fit (default: sqrt(n-features))

4. max-depth : The max depth of a decision tree.

4.3.2 Gaussian Naive Bayes Classifier

It is simple probabilistic supervised classifier[56] based on the famous Bayes theorem. It assumes that the variables are mutually independent (that's why they are "naive"). It is also highly scalable. It assumes gaussian distribution of data in each class and the probability of an instance v in class c is given by the normal distribution as:

$$p(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}}$$

Although this model appears very simple it works exceptionally well for a lot of difficult problems. It is used in fields such as text retrieval where frequency of a word is used to determine the probability of word in a document, and Medical Diagnosis (We are using it here for polyp detection).

It can be used with GaussianNB from sklearn with no important parameters.

4.3.3 Logistic Regression based classification

It is a regression model where output is in the form of classes. It is a simple and popular technique for cases where we have a binary classification[57]. We have two classes for our experiment thus we have a binomial logistic regression. A 0 represents absence of the class while a 1 represents presence of the class, depending on the output obtained from the logistic function used. It uses a logistic function for determining the output class.

$$y = \frac{1}{1 + y^{-z}}$$

Where y is the output variable and z is linear combination of features including the bias term. The algorithm can also be extended to multi-class classification using a 1-vs-rest approach. It can be used from sklearn with following important parameters:

1. penalty : 'l1' or 'l2' which indicates which norm do we have to use for penalization.
2. C : It is the inverse of regularization strength.
3. class weight : Default it assumes 1 for all.

4.3.4 KNN based Classification

This algorithm is best suited for supervised classification based tasks[58]. The data is distributed in the vector space and then the distance of the local neighbours are calculated. This distance can be l1 or l2 norm based, or even custom distance functions. Thus it is a kind of lazy learning too. In this technique the nearest k neighbours are checked and then voting is done on the k neighbours. The class corresponding to the majority of the neighbours is assigned as the class of the new instance. It can be used from the sklearn with following parameters :

1. nNeighbours : The number of neighbours to look for while making the decision (default: 5).
2. weight : Weight of the points in the feature space (default: uniform weights).
3. metric : Used for determining which distance to use. The default is equivalent to the Euclidean metric.

4.3.5 Support Vector Machine Classifier

SVM generally depends on finding a hyperplane that optimally divides the plane into subplanes with each subplane representing a class[59]. This hyperplane has the maximum distance from the nearest set of datapoints for the classes present. This distance is called as margin. The larger the margin the lower is the error. For more complex data we apply kernel trick. Here complex data refers to the data that is not linearly separable. With proper kernel trick, the data is transformed to other dimensions and then optimal plane is obtained. Then classification is performed. It can be used from sklearn with following important parameters.

1. C : penalty parameter of the error (default=1.0).
2. kernel : type of kernel required, i.e.: rbf, linear, sigmoid, poly, etc. (default='rbf', radial basis function).
3. gamma : kernel coefficient (default=3).

4.3.6 Multilayer Perceptron Classifier

MLP is a feed forward neural network with input in the first layer and output i.e class at the last layer[60]. There are multiple layer of nodes which are fully connected. There is an activation function (sigmoid, tanh etc) associated to each node in the neural network which determines the firing of that node. It utilizes back-propagation for learning. The error is back-propagated to the nodes as per required output. They can learn and classify data that are non linearly separable. Parameters for MLP classifier in sklearn :

1. hidden-layer-size : The number of neurons in hidden layers (default: 1 layer with 100 neurons: (100)).
2. activation : Type of activation function required (default: 'relu', rectified linear unit function).
3. solver : Optimization function that will be used like lbfgs, sgd, etc. (default: 'adam', Adaptive Moment Estimation, a modification of Stochastic Gradient Descent.

4.4 Feature Description

As we have described previously, LIRE[8] can be used to extract features. To speed up the process of extracting we utilized a small LIRE program to obtain features from all the images from a directory and save them into a CSV file. The java script we applied is called prallelExtraction.java. This is a modification of a program included in the Lire Sample Application. The script can easily be used to extract all the features mentioned in Table 3.2.

After studying different types of global features, we decided to start extracting 3 sets of some of the most notable global features, which are Joint Composite Descriptor (JCD)[49], Tamura[40] and Texture Histogram (FCTH)[37]. These single global features have been reported to work outstanding, especially for ASU dataset[1].This therefor makes it interesting for us to fuse these 3 sets of features.

4.4.1 Baseline

To obtain a baseline metric, we trained a random forest classifier with 100 trees and looked at the performance of each of the three feature extraction methods independently. To asses the quality of the prediction we measured precision, recall, f1-score, and accuracy:

Below are the results obtained for JCD, Tamura, and FCTH:

Table 4.4: Results obtained for FCTH

FCTH: Accuracy: 82.5%

Values	Precision	Recall	F1 Score	Support
0	0.85	0.92	0.88	13261
1	0.67	0.48	0.56	4313
Avg/total	0.80	0.82	0.80	17574

Table 4.5: Results obtained for JCD

JCD: Accuracy: 82.6%

Values	Precision	Recall	F1 Score	Support
0	0.84	0.95	0.89	13261
1	0.75	0.44	0.55	4313
Avg/total	0.82	0.83	0.81	17574

Table 4.6: Results obtained for Tamura

Tamura: Accuracy: 81.1%

Values	Precision	Recall	F1 Score	Support
0	0.84	0.93	0.88	13261
1	0.68	0.43	0.53	4313
Avg/total	0.80	0.81	0.80	17574

4.4.2 Dimensionality reduction

Dimensionality reduction reduces the dimension of data by projecting the data from the higher to a lower dimension form. This can be of two types - ie linear dimensionality reduction and non linear dimensionality reduction. In linear dimensionality reduction the new dimensions are linear combination of existing dimensions. For eg - PCA (Principal Component Analysis), SVD (Singular Value Decomposition), LDA (Linear Discriminant Analysis), Random Projection etc. While in non linear Dimensionality reduction the relation between the existing and new dimension is non linear. For eg - Kernel PCA, t-SNE etc. Kernel PCA uses kernel trick which is non linear transformation of data with the help of a kernel function.

4.4.3 Cross-validation

It is used for validation of a model. It is generally used to avoid overfitting of a model and to generalize how the model will perform on an independent dataset. It is generally based for models that performs the prediction task ie how accurately the model can predict the outcome. But still it works for classification tasks as well. We generally perform multiple rounds of validation steps for determining the accuracy of a model. for eg - K fold cross validation. In this validation technique the sample is divided into K subsets. One of these subsets is reserved for validation while remaining are used for training. This step is performed k times with each time having different subset for validation. Thus we have k different results obtained after validation. The accuracy is determined by averaging all the k accuracies obtained.

4.5 Early fusion

To define features based on the early fusion methodology, we constructed a single feature vector for every possible combination of the three aforementioned feature types. We performed our tests using logistic regression, Gaussian naïve Bayes, and random forest.

Due to the large number of tests performed, we will not include all performance metric, but rather discussed the best performing setup.

ASU Dataset

Surprisingly, for this dataset, the accuracy was obtained by models using only one of the three possible features. Tamura seems to capture most of the necessary information needed to create accurate models, obtaining an accuracy of 84% when trained with Naïve Bayes, and 82% with logistic regression. Logistic regression on JDC was able to achieve 82% accuracy.

It is difficult to say with absolute certainty why combining features seems to decrease predictive performance, but a possible explanation could be tied to the increase in feature dimensionality and the amount of data available. As the number of dimensions representing the data increases, the patterns that help differentiate between classes are more difficult to identify, and the learning algorithms require more data to distinguish between noise and signal.

The best performing setups on the ASU dataset are given in the table below:

MMSys Dataset

For this dataset, we decided to use all global features available from LIRE and compare the performance of several different classification methods using early fusion. The classifiers in questions were: logistic regression,

Table 4.7: Early fusion of 3 set of features(JDC, Tamura and FCTH

Features	Model	Accuracy	F1	Precision	Recall
Tamura	Logistic	0.840	0.688	0.661	0.717
Tamura	Naïve Bayes	0.829	0.659	0.647	0.672
JDC	Logistic	0.829	0.630	0.671	0.594

multi-layer perceptron (MLP), support vector machines (SVM), random forest, k-nearest neighbors (KNN), Gaussian Naïve Bayes.

Table 4.8: Early fusion of all global features

Model	Accuracy	F1	Precision	Recall	F beta
LogRegression	0.87	0.869	0.870	0.87	0.869
MLP	0.863	0.862	0.864	0.863	0.862
SVM	0.777	0.775	0.779	0.777	0.775
RandForest	0.749	0.747	0.751	0.749	0.747
KNN	0.683	0.674	0.699	0.683	0.676
Gaussian NB	0.564	0.544	0.565	0.564	0.551

Logistic regression and multi-layer perceptrons clearly outperform the rest of the models. A possible explanation for this difference in performance is that out of all the tested classifiers, logistic regression and MLP are better suited for distinguishing between classes that are non-linearly separable. Introducing polynomial features of higher order would likely improve the performance on the rest of the classifiers, but that line of work is beyond the scope of this project.

Using PCA

We decided to experiments with PCA to reduce the dimensionality of the dataset, given that combining all the global features resulted in a dataset with 3778 variables.

We used the sklearn implementation of the PCA algorithm, were the most important parameter is the number of components to keep. This parameter can be defined in two ways: as an integer, predefining the exact number of components to keep, or as a float between 0 and 1, defining the percentage of variance that needs to be explained (the algorithm will keep the minimum number of components needed to explain that amount of variance).

We defined n-components = 0.99 (99% of variance explained), and the resulting reduced dataset had 1040 variables, about 72% less than the original.

The resulting metrics were slightly worse than without using PCA, with three exceptions: the case of SVM and Gaussian NB which had an improvement, and Random Forest with was significantly worse:

Table 4.9: Early Fusion of all global features(using PCA)

model	accuracy	F1	precision	recall	Fbeta
MLP	0.865	0.865	0.865	0.866	0.865
LogisticRegression	0.865	0.864	0.865	0.865	0.864
SVM	0.823	0.822	0.823	0.823	0.822
KNN	0.680	0.670	0.695	0.679	0.672
Gaussian NB	0.676	0.672	0.702	0.676	0.670
RandomForest	0.409	0.406	0.418	0.408	0.406

SVM and Naive Bayes are probably benefiting of the reduced number of variables, especially SVM which is sensitive to the number of dimensions.

4.6 Late fusion

A standard technique used for late fusion is based on ensemble methods. Several distinct classifiers are fit to the data, and their outputs is combined and used as a feature to fit another classifier. This has the benefit of leveraging the strength of each individual model in the final prediction.

In this work, we obtained late fusion features from LIRE and used them to test several different classification models on each dataset.

Late fusion is performed after classification tasks are performed. In this we can't learn about the correlation between the attributes or the features. It doesn't have the problem of high dimensions as each set of features is used by a different classifier. The Late fusion comprises of stacking of classification models. In the base model (layer 1) we have performed classification with existing features and the output of base model are added as features in the training data.

We apply following methods to the result obtained-

4.6.1 Stacking

Basically, in this method we have two layers of classifiers: after we have performed the classification tasks with the 1st layer of classifiers we use the resulting predicted probabilities as training data for a 2nd level classifier. This classifier learns to combine the results of the first layer classifiers to make predictions on the data, and the output of the 2nd level classifier is the final prediction.

In order avoid over-fitting (memorization of the training set), it is important to follow a set of guidelines for doing stacking. The main idea is to randomly divide the training set into a number of folds, 5 for example. Then, when training the 1st layer of classifiers, we separate one of the folds, train the algorithm on the rest, do prediction on the separated fold, and store the predictions corresponding to those examples. This procedure is

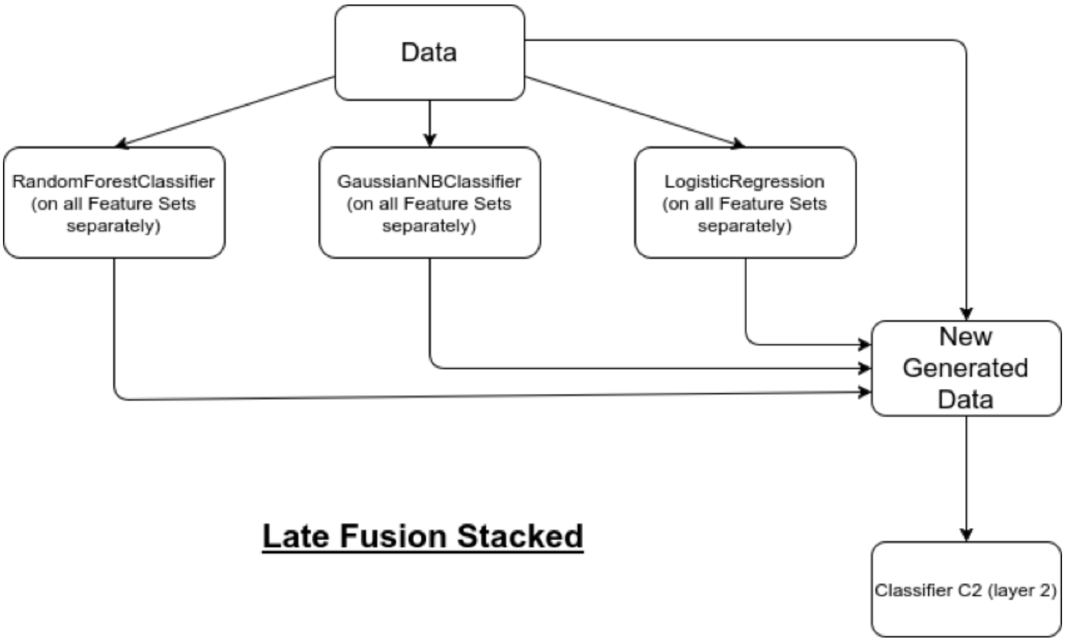

Data
RandomForestClassifier (on all Feature Sets separately)
GaussianNBClassifier (on all Feature Sets separately)
LogisticRegression (on all Feature Sets separately)
New Generated Data
Late Fusion Stacked
Classifier C2 (layer 2)

Figure 4.1: Pipeline of Late Fusion Stacked

repeated for all of the folds, at the end obtaining predictions for all the examples in the training set. That way, the predictions that are going to be used as training data for the 2nd level classifier are obtained without 'looking' at the corresponding examples.

4.6.2 Average Late Fusion

The probability output of each classifier in the first layer is averaged. The averaged output is the final output, after being converted into a class prediction. Let c1,c2,c3 be the results of the classification obtained on the second layer. Therefore the new result :

$$y = \frac{c_1 + c_2 + c_3}{3}$$

(where y is the result averaged output), and finally: y = (y > 0.5).

4.6.3 Weighted Late Fusion

When obtaining the results from 1st layer classifier we also obtain cross-validation metrics (accuracy or f1-score) on the performance of each classifier. Latter, to combine the results of the first layer, we do weighted averaging of the probability outputs obtained, where each assigned weight is the result of the cross-validation metric so each output is weighted with the performance of each classifier: that way better classifiers will have higher weights.

$$y = \frac{(\beta_1 c_1 + \beta_2 c_2 + \beta_3 c_3)}{\beta_1 + \beta_2 + \beta_3}$$

Where y is the resultant output and beta1,,beta2 and beta3 are the weights obtained.

For our purpose we have used both Average and Weighted Late Fusion. This improves the result a lot than compared to the baseline model.

4.6.4 ASU Dataset

Not surprisingly, using late fusion clearly outperformed the best performing methods in the ASU dataset. Having an ensemble of methods allow each classifier to include their own benefits, and handle corner cases that other classifiers are not able to do. In this experiment, the final classification was done by logistic regression, random forest, and Gaussian naïve bayes.

Table 4.10: Results of late fusion of all global features(using ASU dataset)

Model	**Accuracy**	**F1**	**Precision**	**Recall**	**F beta**
LogRegression	0.883	0.732	0.842	0.648	0.679
RandForest	0.869	0.658	0.923	0.511	0.562
Gaussian NB	0.856	0.630	0.851	0.500	0.545

Even though the best performing classifier was still logistic regression, in this case the accuracy increased from 84% to over 88%. Likewise, precision and F1 score saw a significant improvement.

4.6.5 MMSys Dataset

We perform the same set of experiments as we did with the ASU dataset. In this case, however the tested classification models were logistic regression, multi layer perceptron (MLP), random forest, support vector machines (SVM), k-nearest neighbors (KNN), and Gaussian naïve bayes.

Table 4.11: Results of late fusion of all global features(using MMSys dataset)

Model	**Accuracy**	**F1**	**Precision**	**Recall**	**F beta**
MLP	0.896	0.895	0.898	0.896	0.895
LogRegression	0.88	0.883	0.886	0.884	0.883
RandForest	0.857	0.856	0.857	0.857	0.856
SVM	0.849	0.847	0.851	0.849	0.847
KNN	0.835	0.832	0.840	0.835	0.833
Gaussian NB	0.63	0.581	0.726	0.63	0.595

Once more, we were able to obtain a significant accuracy improvement in our best performing classification model, going from 87% accuracy to almost 90%. As it happened on our previous experiment, the classifiers that are able to create non-linear separating regions were the best performing ones, namely: multi-layer perceptron and logistic regression.

Using PCA

As we did with Early Fusion, we added PCA to the experiments with this dataset. In this case the results were very similar to those obtained without PCA, probably because the original datasets didn't have a problem of too many variables. Again, only SVM and Naive Bayes benefited with PCA.

Table 4.12: Results of late fusion of all global features(using MMSys dataset and by using PCA)

Model	Accuracy	F1	Precision	Recall	F beta
MLP	0.893	0.893	0.895	0.893	0.893
LogRegression	0.88	0.881	0.885	0.882	0.881
RandForest	0.83	0.83	0.83	0.83	0.828
SVM	0.859	0.858	0.862	0.859	0.858
KNN	0.835	0.832	0.840	0.835	0.833
Gaussian NB	0.825	0.823	0.823	0.825	0.923

4.7 Results Comparison

In this section, we do a side by side comparison of the results obtained using early fusion, averaged late fusion, and stacked late fusion. For the sake of conciseness, we discuss the three more relevant metrics: accuracy, precision, and recall.

4.7.1 ASU Dataset

The graphs below show a side by side comparison of using early fusion and late fusion on the ASU dataset. We looked at three different classification methods, Gaussian Naïve Bayes, logistic regression, and random forest.

Accuracy

All models were able to achieve at least an 80% accuracy on this first dataset, which is a fairly reasonable result. However, there is a clear benefit in using late fusion as opposed to early fusion, in particular, stacked late fusion. When using early fusion, model selection did not seem to have a significant impact on accuracy, but late fusion is able to capture the most relevant information through the use of multiple learners. Logistic regression was able to outperform the other models, achieve approximately 90% accuracy with late fusion.

Precision

Just as with accuracy, late fusion is able to significantly improve precision. Early fusion was able to achieve a precision score of .76, while late fusion was able to obtain a precision of .92. Random forest was able to obtain

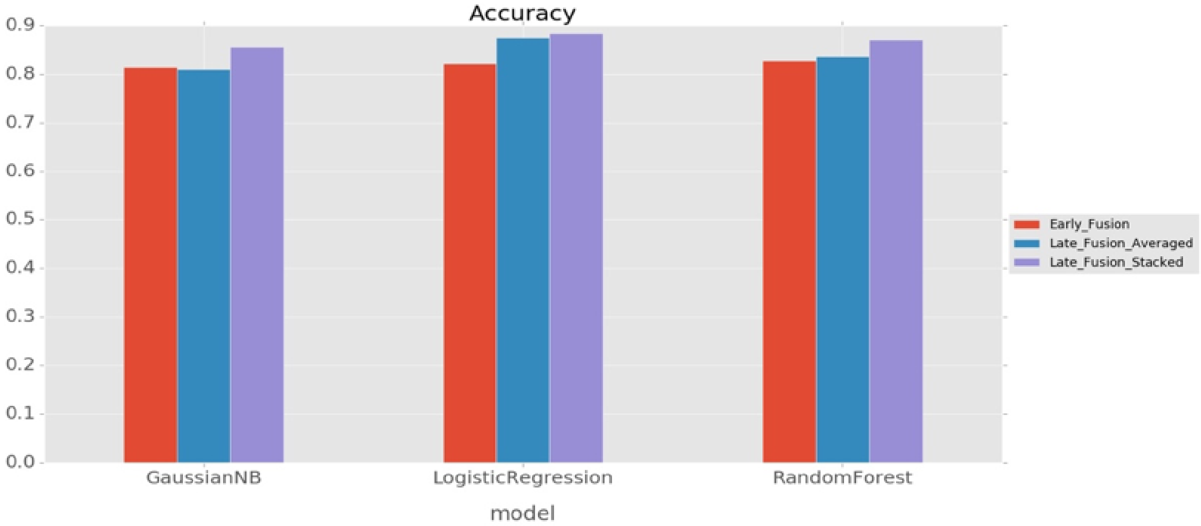


Figure 4.2: Accuracy comparison of different fusing methods and various classifiers(ASU dataset)

the highest precision score of all learners, a possible reason being that the nonlinear partitions it creates on the space of the data is well tailored to the data presented. It generalizations abilities are limited, as evidenced by the accuracy results, but it is less prone to false positives than other classifiers.

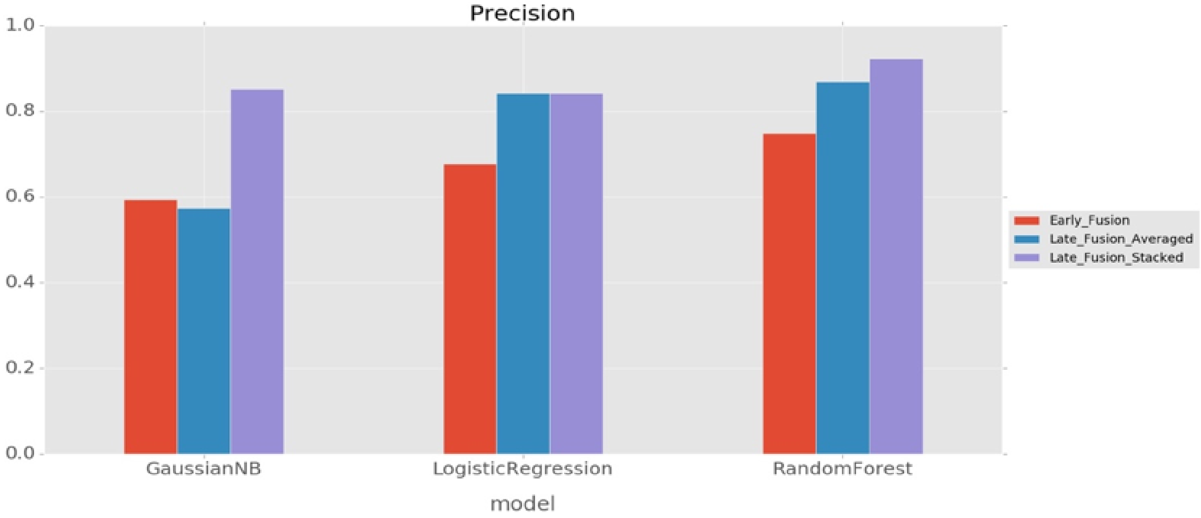


Figure 4.3: Precision comparison of different fusing methods and various classifiers(ASU dataset)

Recall

Surprisingly, the effect of feature extraction method on recall does not seem to be clear. The combined selection of early or late fusion, and the learner used for classification seem to give highly varied results. Averaged late fusion with naïve Bayes gave the highest recall score, reaching 0.87. However, the same learner with stacked late fusion had a recall of 0.5. On the other hand, logistic regression and random forest seem to have benefited from stacked late fusion, increasing their performance by over 0.1 from early fusion.

A possible explanation of why naïve Bayes could be able to outperform the other learners by such a large margin is that while logistic regression and random forest seek to partition the space into positive and negative regions, naïve bayes is simply looking at the probability of an example being positive or negative given the features that describes it. In other words, it takes a completely different approach to prediction that seems to performed particularly well when measuring recall.

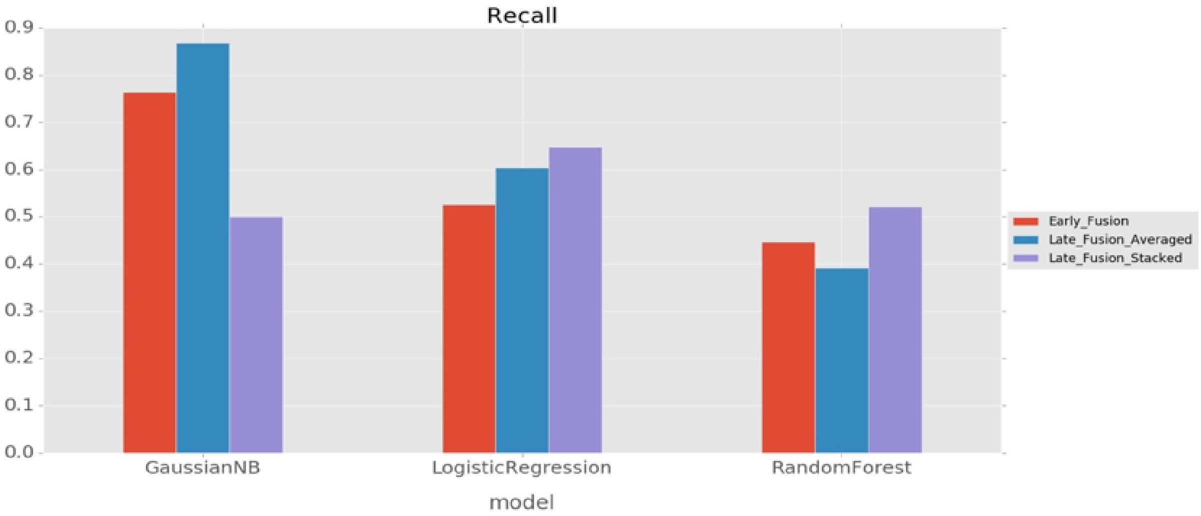


Figure 4.4: Recall comparison of different fusing methods and various classifiers(ASU dataset)

4.7.2 MMSys Dataset

The graphs below show a side by side comparison of using early fusion and late fusion on the ASU dataset. We looked at several different classification methods: naïve Bayes, logistic regression, neural networks, support vector machines, k-nearest neighbors, and random forest.

Accuray

Logistic regression and neural networks were able to produce the most accurate models. Although not very significant, late fusion did improve prediction accuracy over early fusion; taking logistic regression from 83% to 88% accuracy, and neural networks from 83% to 89%. Even though the other models were not as accurate as these two, they all saw an accuracy improvement by using late fusion.

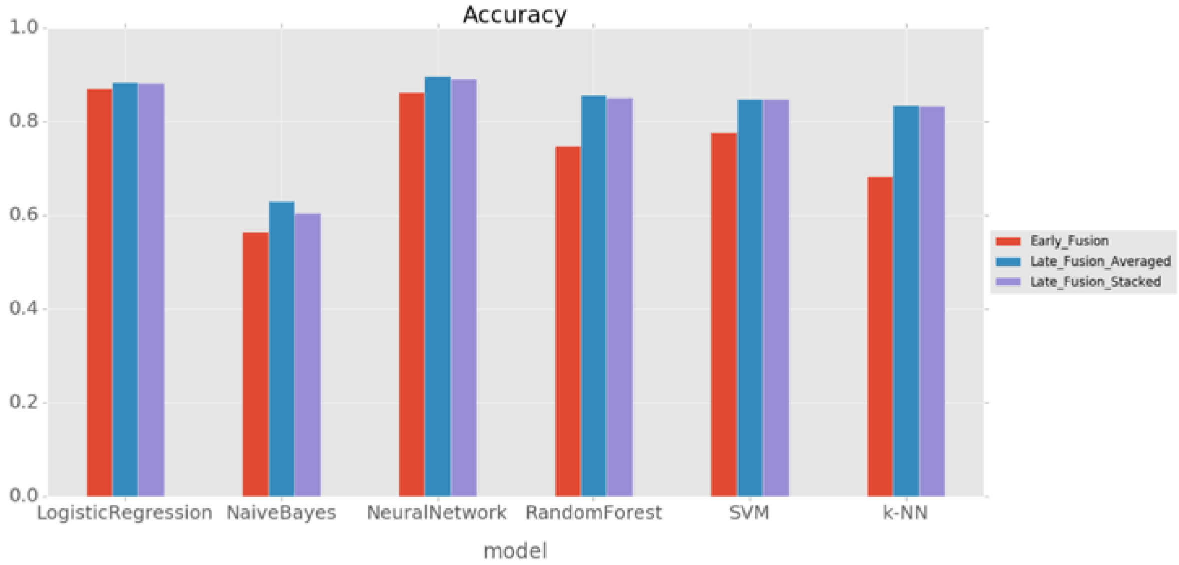


Figure 4.5: Accuracy comparison of different fusing methods and 6 different types of classifiers(MMSys dataset)

Precision

Just as it happened with the previous dataset, all models experience an improvement in precision when using late fusion over early fusion. Logistic regression and neural networks seems to be the ones able to achieve the highest score, with a precision of .88 and .89 respectively.

Recall

As opposed to the previous dataset, in this dataset late fusion seems to improve recall regardless of the learning model. Again, logistic regression and neural networks are able to achieve the highest recall.

These results further support the idea that in this dataset there is a nonlinear relationship between the features that the other models are not able to capture.

Confusion matrices

The confusion matrices are useful tools to check the errors that the classifiers are making, especially to examine which classes the classifiers are confounding with which others, something than can be useful in a diagnostic context. We will examine the matrices for the best (MLP) and

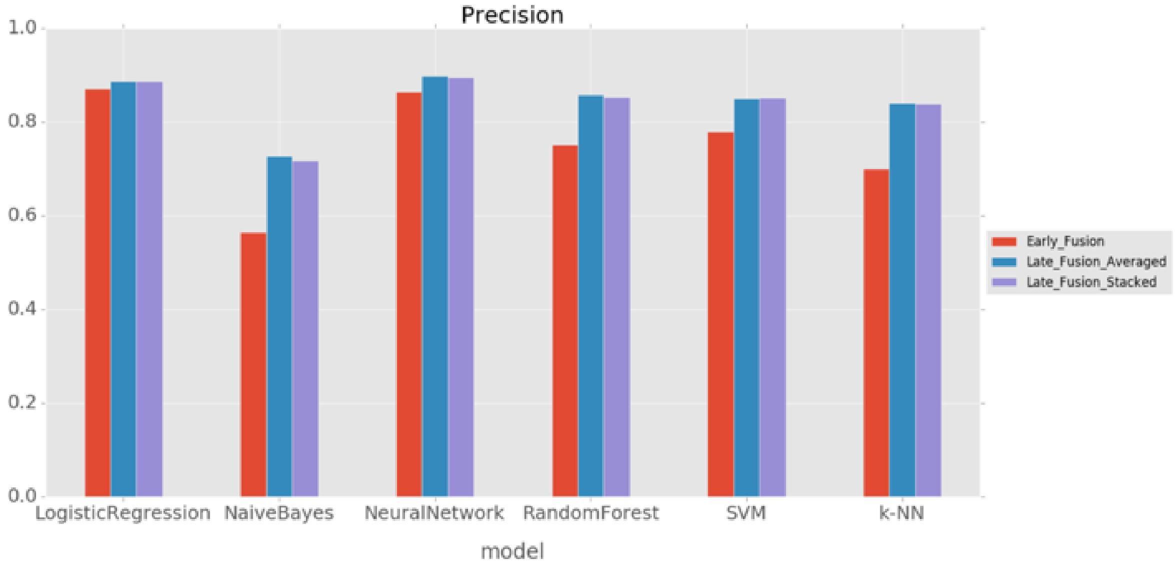


Figure 4.6: Precision comparison of different fusing methods and 6 different types of classifiers(MMSys dataset)

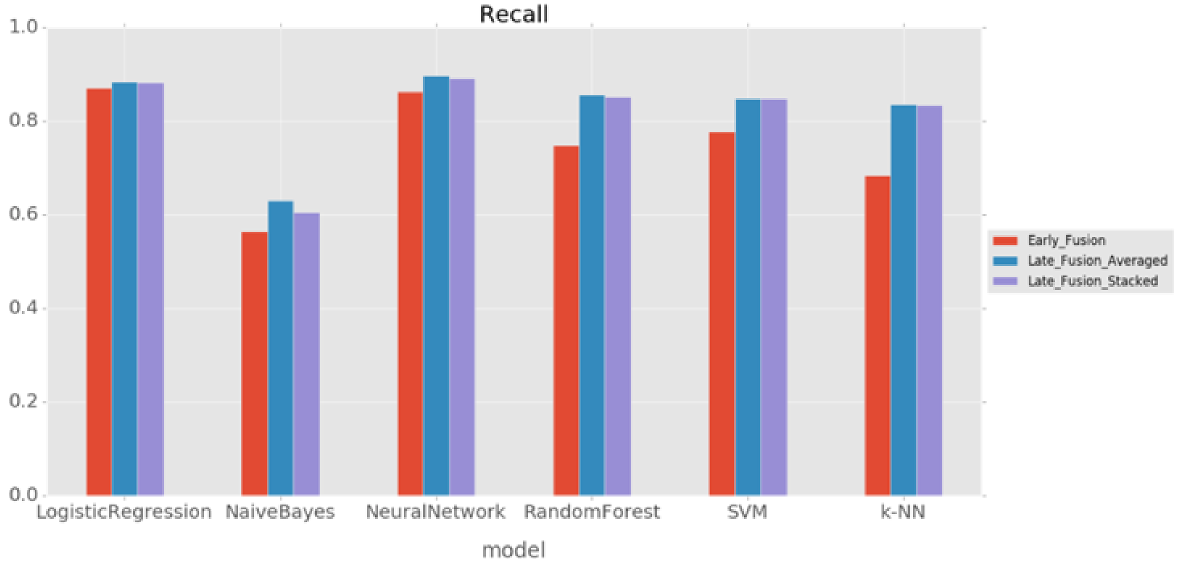


Figure 4.7: Recall comparison of different fusing methods and 6 different types of classifiers(MMSys dataset)

worst (GaussianNB) classifiers, according to accuracy, both in early and late fusion. The numbers in the matrices are normalized to reflect rates and not total numbers.

In this matrix(Figure 4.8) we can see the types of errors that the MLP classifier is making. In particular, the classifier is confounding inked-lifted-polyps with inked-resection-margins about 30% of the time (and viceversa at a lower rate). Also, MLP is having problems distinguishing normal-z-

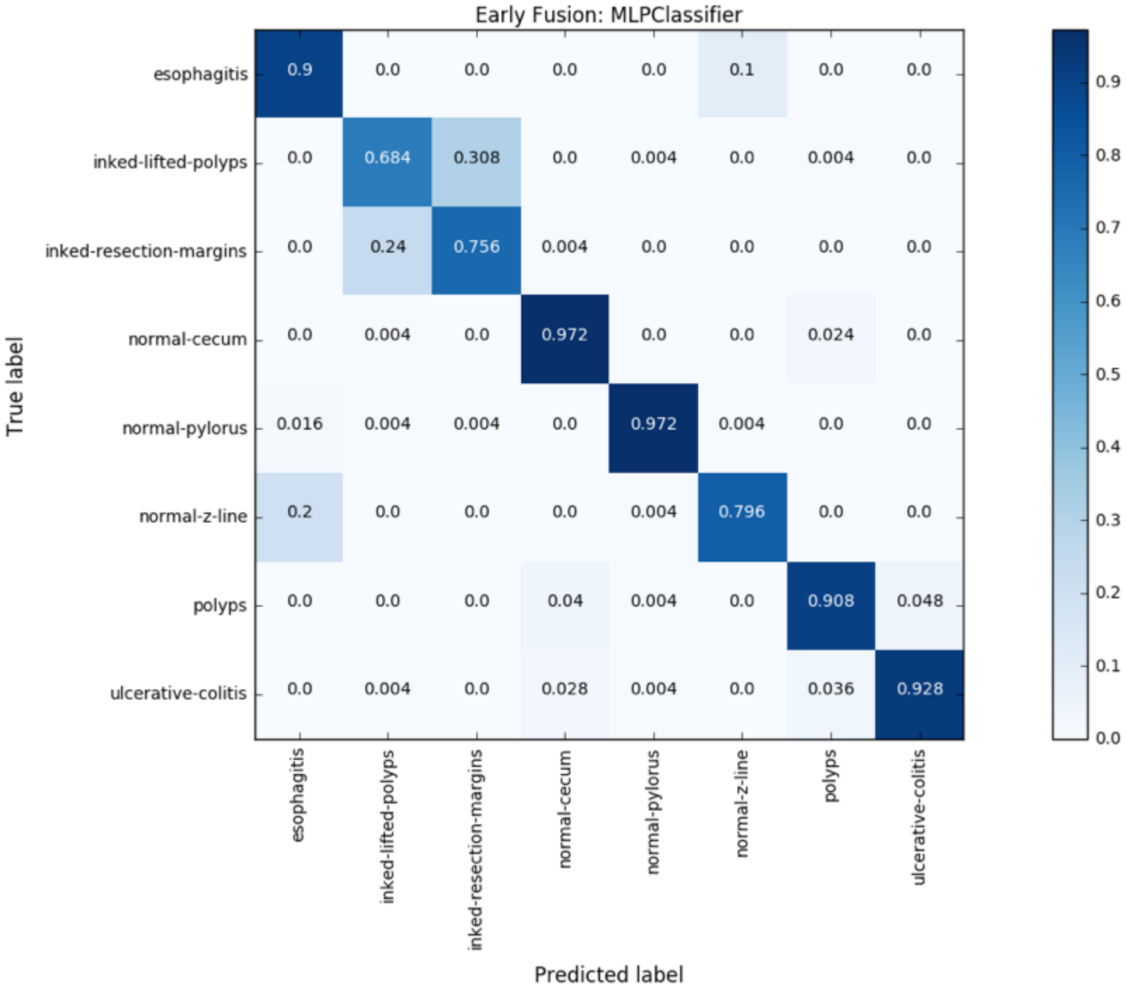


Figure 4.8: Early Fusion: MLP classifier(MMSys dataset)

line and esophagitis about 20% of the time (and viceversa at a lower rate). All other errors are much less prevalent.

This information could inform the doctors while reading the diagnostic so they could pay especial attention, and maybe ordering additional tests, if the classification falls into these areas.

Naive Bayes(Figure 4.9) is making the same mistakes of MLP but at a much higher rate, and also have other highly prevalent errors, like confounding normal-pylorus with esophagitis about 36% of the time, polyps with inked-resection-margins 20% and with ulcerative-colitis 30% of the time, and normal-pylorus with normal-z-line 23% of the time. Given the number of prevalent errors, this classifier is much less useful for diagnostic.

In the case of late fusion(Figure 4.10), we can see that the types of errors are similar than with early fusion, but in general the classifier is being confounded less (observe the higher prevalence of 0.0 outside the main diagonal). This is a sign of a better classifier, in line of what we have found

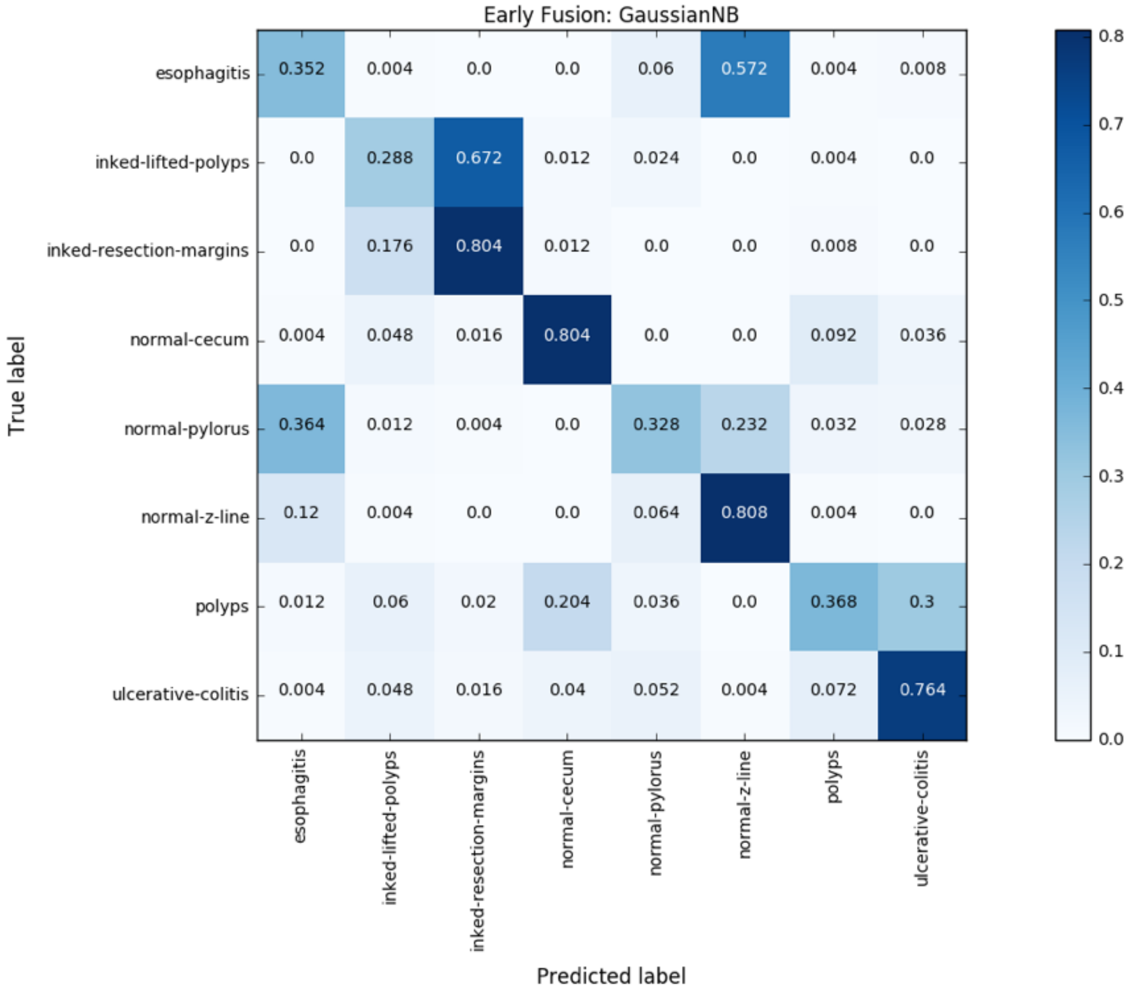


Figure 4.9: Early Fusion: GaussianNB classifier(MMSys dataset)

earlier.

As in the case of early fusion, GaussianNB(Figure 4.11) is much worse than MLP, for example, is almost completely confounding inked-lifted-polyps with inked-resection-margins, but also is not making the corresponding error of confounding inked-resection-margins with inked-lifted-polyps, probably because it's giving much more weight to the inked-resection-margins class. Other systemic errors are also present, but overall is much better than the same classifier with early fusion.

4.8 Summary

In this chapter, we analysed the performance of early fusion and late fusion as means of feature extraction for classification tasks. We ran experiments on two challenging datasets, and compared the accuracy of several different classification methods using early and late fusion. From these results, the benefits of late fusion over early fusion seem evident.

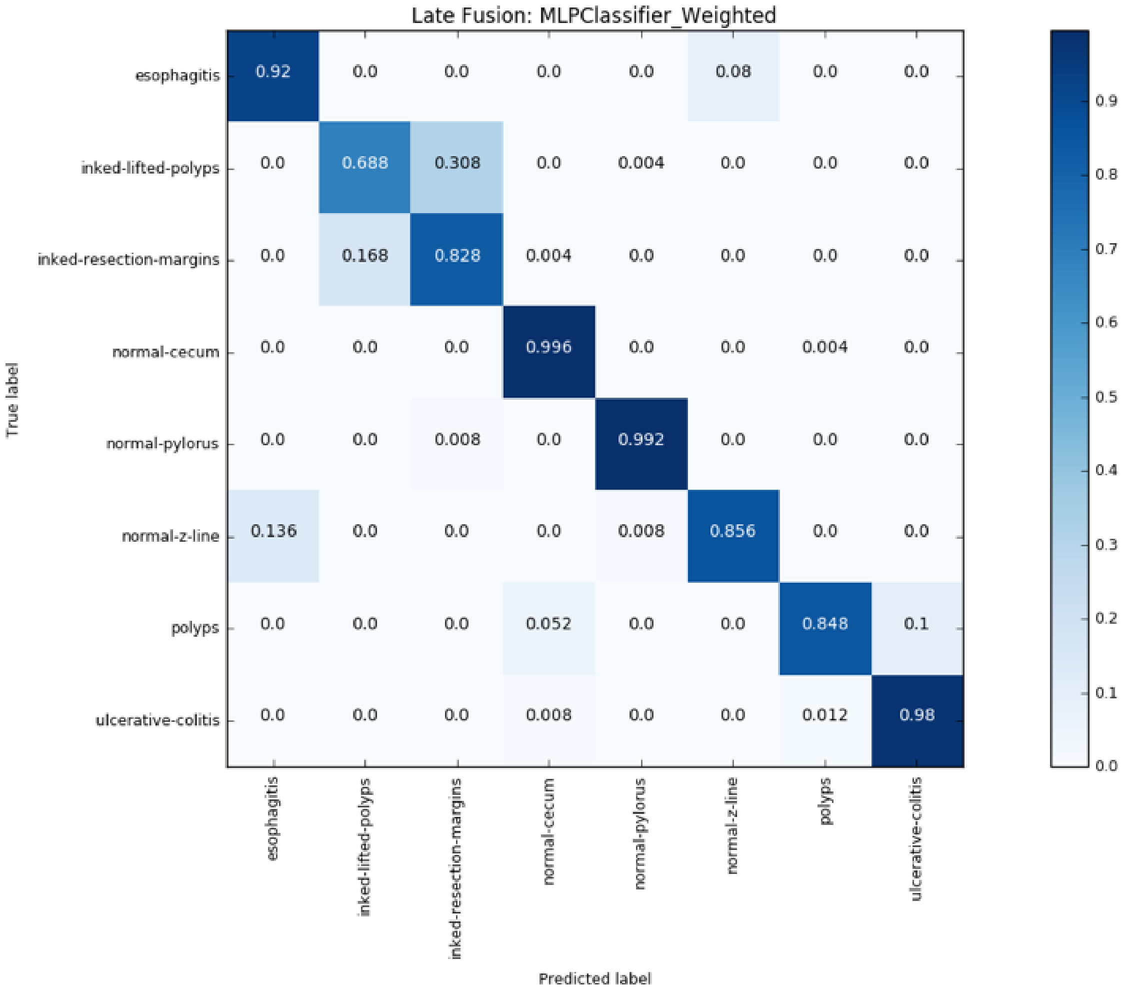
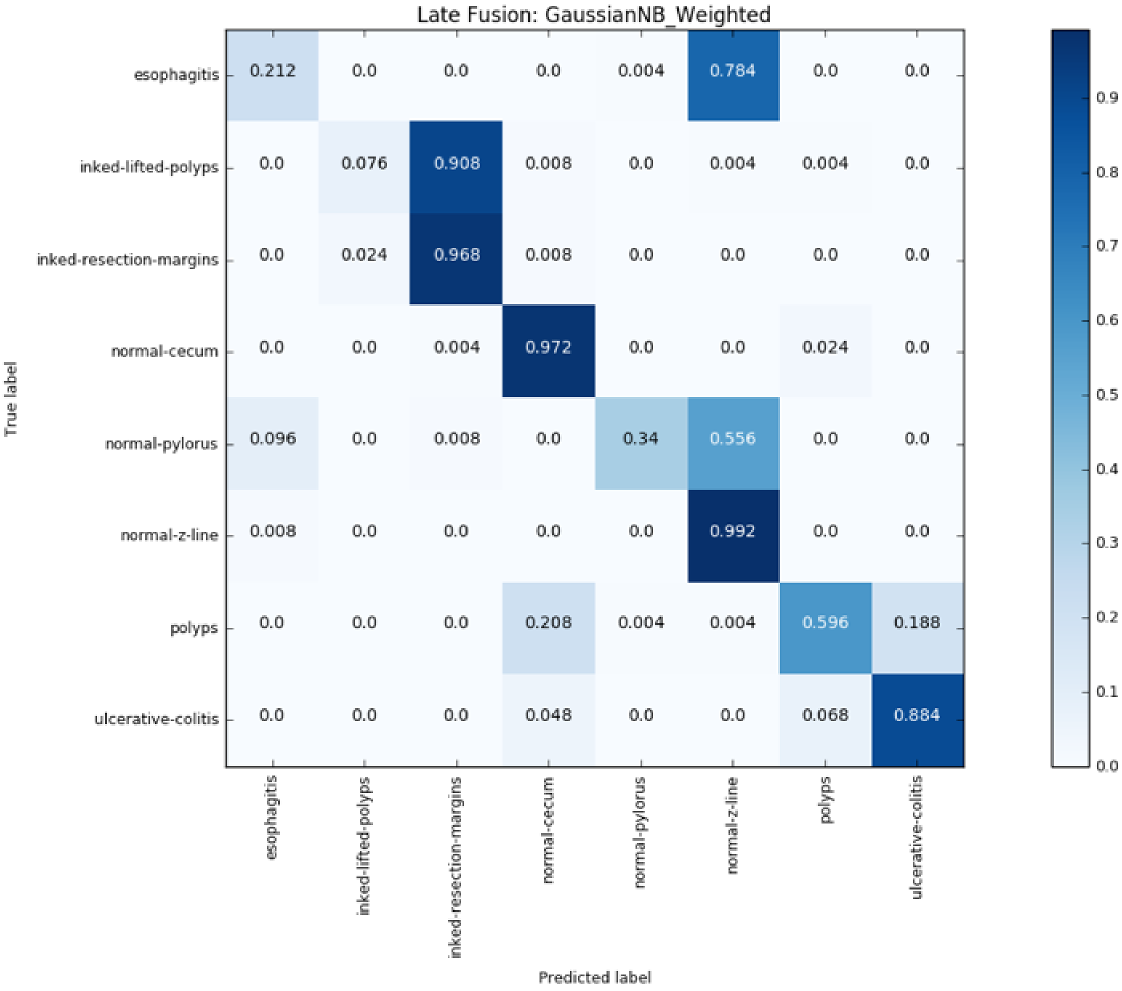


Figure 4.10: Late Fusion: MLP classifier Weighted classifier(MMSys dataset)

Not only was late fusion able to improve the best obtained accuracy across different models for both datasets, but we observed an improvement of accuracy for each model individually when comparing to early fusion. However, these improvements come with a cost. Obtaining features for late fusion, require fitting several different models to the data, which can be quite time consuming.

We hypothesize that the reason for these accuracy improvements is that, by using late fusion, we are incorporating information extracted from several different models that we would not have been to capture had we only used a single model. In a way, late fusion takes the best of each classifier, and produces much more informative features on which one can learn a model.

The experiments suggest that, whenever possible, late fusion or some other ensemble method should be considered instead of simply attempting to learn a model through early fusion.



	esophagitis	inked-lifted-polyps	inked-resection-margins	normal-cecum	normal-pylorus	normal-z-line	polyps	ulcerative-colitis
esophagitis	0.212	0.0	0.0	0.0	0.004	0.784	0.0	0.0
inked-lifted-polyps	0.0	0.076	0.908	0.008	0.0	0.004	0.004	0.0
inked-resection-margins	0.0	0.024	0.968	0.008	0.0	0.0	0.0	0.0
normal-cecum	0.0	0.0	0.004	0.972	0.0	0.0	0.024	0.0
normal-pylorus	0.096	0.0	0.008	0.0	0.34	0.556	0.0	0.0
normal-z-line	0.008	0.0	0.0	0.0	0.0	0.992	0.0	0.0
polyps	0.0	0.0	0.0	0.208	0.004	0.004	0.596	0.188
ulcerative-colitis	0.0	0.0	0.0	0.048	0.0	0.0	0.068	0.884

Figure 4.11: Late Fusion: GaussianNB classifier Weighted classifier(MMSys dataset)

Part III

Conclusion

Chapter 5

Conclusion

5.1 Main Contributions

In this work, we conduct an in depth experimental study of different feature fusion methodologies. Early and late fusion approaches have been used extensively, and are a prevalent way of constructive informative features for predictive models. Up to this point, we had an intuitive understanding of the pros and cons of both early and late fusion, but we have not run them through extensive experimental testing.

Through the use of two large publicly available datasets, we studied the performance of each feature extraction method, and found evidence that further supports our initial intuition; that is, early fusion is cheaper to run for feature construction, whereas late fusion is able to more informative features which tends to obtain higher accuracy in predictive models. We have found that the choice of predictive model can have a substantial impact in overall accuracy, but the use of late fusion tend to improve accuracy for each model independently. We also considered other standard measures with which we could asses the performance of these features: precision, and recall. For virtually every model we tested, we found that accuracy improvement given by feature fusion seems to be correlated with an improvement in precision. This is also true for recall, however to a lesser extent, where we found that naïve Bayes is a bit less predictable in terms of how different features influences it.

5.2 Future work

Even though we aimed to be as thorough as possible, there are still questions that should be explored. In our experiments, we focused on discrimination by considering feature vectors living in the Euclidean space. However, since we are working with images, there might be a more robust explanation of the features if we consider a lower dimensional manifold representation. We discussed the results using PCA for one of our dataset, but that is only dealing with a linear subspace projection. A place for further exploration is to consider non-linear embeddings, such as locally linear embedding (LLE), isomap, multi-dimensional scaling (MDS), etc.

Another area for further study would be looking at a different style of datasets. We explored different feature fusion method on image datasets, which could really bias the results. It is possible that these early and late fusion are particularly well (or poorly) suited for image data, so further exploration could be done in the areas of natural language processing (NLP), transfer learning (TL) or reinforcement learning (RL), for example. One might also argue that, while these feature extraction methods seem intuitive and lead to better performance, most state-of-the-art results in recent years have been achieved through implicit feature learning through deep learning techniques. A more direct comparison between features extracted through deep learning, and early and late fusion techniques would be a reasonable step to further study how these different approaches compare to each other.

We also would like to extend our current experimental setup to other related dataset, in particular Kvasir dataset[61] and Nerthus dataset[62]. Kvasir is a dataset containing images from inside the gastrointestinal (GI) tract. The collection of images are classified into three anatomical landmarks and three clinically significant finding. It also contains two categories of images related to endoscopic polyp removal. Nerthus is a dataset containing videos from inside the gastrointestinal (GI) tract, showing different degrees of bowel cleansing.

Since these datasets come from a similar set of tasks as the ones used in our experiments, extending our study to these datasets would provide further evidence that learning models on these type of images could greatly benefit from feature fusion, specially from late fusion.

5.3 Final remarks/thoughts

Our findings throughout these experiments further support what we intuitively thought all along: late fusion tends to improve model accuracy, at the expense of added computational cost.

At this point, we can safely suggest that machine learning practitioners deciding between early and late feature extraction methods should follow the following rule of thumb. If the intent is to create a system that should be updated or retrained fairly often, it might be worth sacrificing a bit of accuracy for meeting certain time constraints. In those situations, early fusion might be a more pragmatic choice. On the other hand, if the system does not experience regular updates and the model is frozen once it was trained, then the improved accuracy of using late fusion techniques might be worth the extra cost.

Given the results we obtained, it seems like logistic regression and multi-layer perceptrons are able to model relationships which other learning methods cannot find. This finding is consistent with current approaches using deep learning, which can be thought of as taking MLP nonlinear capacities to a much higher degree.

We suggest that, if thorough experimentation with different models is not possible and prior knowledge if available for modelling a problem, machine learning practitioners should opt neural network based approaches since they have consistently show to be a robust learning model.

Appendix A
Source Code

The source code is available at the following link:

https://github.com/SalmanMasterThesis/MasterThesis

All of the visualizations and graphs are also added there.

Bibliography

[1] Centers for Disease Control, Prevention (CDC, et al. Vital signs: colorectal cancer screening test use–united states, 2012. *MMWR. Morbidity and mortality weekly report*, 62(44):881, 2013.

[2] Digestive Diseases Statistics. National institute of diabetes and digestive and kidney diseases. 2009.

[3] NHS.Gastroscopy. Retrieve from http://www.nhs.uk/conditions/gastroscopy. 2015.

[4] Michael Riegler, Mathias Lux, Vincent Charvillat, Axel Carlier, Raynor Vliegendhart, and Martha Larson. Videojot: A multifunctional video annotation tool. In *Proc. of ACM ICMR*, pages 534–537, 2014.

[5] A. Gross. Gastrointestinal diseases rise in asia. medtech intellegence. retrieved from https://www.medtechintelligence.com/feature$_{a}rticle/gastrointestinal - diseases - rise - in - asia/.$2016.

[6] 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.

[7] Michael Riegler, Konstantin Pogorelov, Jonas Markussen, Mathias Lux, Håkon Kvale Stensland, Thomas de Lange, Carsten Griwodz, Pål Halvorsen, Dag Johansen, Peter T Schmidt, et al. Computer aided disease detection system for gastrointestinal examinations. In *Proceedings of the 7th International Conference on Multimedia Systems*, page 29. ACM, 2016.

[8] Mathias Lux. Lire: open source image retrieval in java. In *Proceedings of the 21st ACM MM*, pages 843–846. ACM, 2013.

[9] World Cancer Research Fund International. Retrieve from http://www.wcrf.org/int/cancer-facts-figures/worldwide-data. 2015.

[10] Hermann Brenner, Matthias Kloor, and Christian Peter Pox. Colorectal cancer. *The Lancet*, 383(9927):1490–1502, Feb. 2016.

[11] O. Holme, M. Bretthauer, A. Fretheim, J. Odgaard-Jensen, and G. Hoff. Flexible sigmoidoscopy versus faecal occult blood testing for colorectal cancer screening in asymptomatic individuals. *Cochrane Database of SR*, 2013.

[12] L. von Karsa, J. Patnick, and N. Segnan. European guidelines for quality assurance in colorectal cancer screening and diagnosis. first edition–executive summary. *Endoscopy*, 44 Suppl 3:SE1–8, 2012.

[13] S Kudo, S Hirota, T Nakajima, S Hosobe, H Kusaka, T Kobayashi, M Himori, and A Yagyuu. Colorectal tumours and pit pattern. *Journal of Clinical Pathology*, 47(10):880–885, 1994.

[14] S Oba, S Tanaka, Y Sano, S Oka, and K Chayama. Current status of narrow-band imaging magnifying colonoscopy for colorectal neoplasia in japan. *Digestion*, 83(3):167–172, 2011.

[15] H Inoue, H Kashida, S Kudo, M Sasako, T Shimoda, H Watanabe, S Yoshida, M Guelrud, CJ Lightdale, K Wang, et al. The paris endoscopic classification of superficial neoplastic lesions: esophagus, stomach, and colon: November 30 to december 1, 2002. *Gastrointest Endosc*, 58(6 Suppl):S3–43, 2003.

[16] Ayso H de Vries, Shandra Bipat, Evelien Dekker, Marjolein H Liedenbaum, Jasper Florie, Paul Fockens, Roel van der Kraan, Elizabeth M Mathus-Vliegen, Johannes B Reitsma, Roel Truyen, et al. Polyp measurement based on ct colonography and colonoscopy: variability and systematic differences. *European radiology*, 20(6):1404–1413, 2010.

[17] Sean R Stanek, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, Ruwan D Nawarathna, Jayantha Muthukudage, and Piet C De Groen. Sapphire middleware and software development kit for medical video analysis. In *Computer-Based Medical Systems (CBMS), 2011 24th International Symposium on*, pages 1–6. IEEE, 2011.

[18] Sean R Stanek, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, Ruwan D Nawarathna, Jayantha Muthukudage, and Piet C De Groen. Sapphire: A toolkit for building efficient stream programs for medical video analysis. *Computer methods and programs in biomedicine*, 112(3):407–421, 2013.

[19] Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Marius Tennøe, Espen Helgedagsrud, Mikkel Næss, Henrik Kjus Alstad, Asgeir Mortensen, Ragnar Langseth, Sigurd Ljødal, Østein Landsverk, et al. Bagadus: An integrated real-time system for soccer analytics. *ACM TOMM*, 10(1s), 2014.

[20] Ragnar Langseth, Vamsidhar Reddy Gaddam, Håkon Kvale Stensland, Carsten Griwodz, and Pål Halvorsen. An evaluation of debayering algorithms on gpu for real-time panoramic video recording. In *Multimedia (ISM), 2014 IEEE International Symposium on*, pages 110–115. IEEE, 2014.

[21] Danyu Liu, Yu Cao, Ki-Hwan Kim, Sean Stanek, Bancha Doungratanaex-Chai, Kungen Lin, Wallapak Tavanapong, Johnny Wong, JungHwan Oh, and Piet C De Groen. Arthemis: Annotation software in an integrated capturing and analysis system for colonoscopy. *Computer methods and programs in biomedicine*, 88(2):152–163, 2007.

[22] Michael Riegler, Mathias Lux, Vincent Charvillat, Axel Carlier, Raynor Vliegendhart, and Martha Larson. Videojot: A multifunctional video annotation tool. In *Proc. of ACM ICMR'14*, page 534. ACM, 2014.

[23] Yi Wang, Wallapak Tavanapong, Johnny S Wong, JungHwan Oh, and Piet C de Groen. Detection of quality visualization of appendiceal orifices using local edge cross-section profile features and near pause detection. *BME*, 2010.

[24] A.V. Mamonov, I.N. Figueiredo, P.N. Figueiredo, and Y.-H.R. Tsai. Automated polyp detection in colon capsule endoscopy. *Transs on MI*, 2014.

[25] 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.

[26] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[27] Nima Tajbakhsh, Suryakanth Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging*, 35(2):630 – 644, February 2016.

[28] Francesco Ciompi, Kaman Chung, Sarah J van Riel, Arnaud Arindra Adiyoso Setio, Paul K Gerke, Colin Jacobs, Ernst Th Scholten, Cornelia Schaefer-Prokop, Mathilde MW Wille, Alfonso Marchiano, et al. Towards automatic pulmonary nodule management in lung cancer screening with deep learning. *arXiv preprint arXiv:1610.09157*, 2016.

[29] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *arXiv preprint arXiv:1412.1897*, 2014.

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

[31] Xirong Li, Cees GM Snoek, and Marcel Worring. Unsupervised multi-feature tag relevance learning for social image retrieval. In *Proc. of ACM ICMR*, pages 10–17, 2010.

[32] Cees G. M. Snoek, Marcel Worring, and Arnold W. M. Smeulders. Early versus late fusion in semantic video analysis. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, MULTIMEDIA '05, pages 399–402, New York, NY, USA, 2005. ACM.

[33] 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 *Proc. of CBMI*, 2016.

[34] Christopher M Bishop. *Information science and statistics*. Springer, New York, 2006.

[35] Savvas A Chatzichristofis and Yiannis S Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer Vision Systems*, pages 312–322. Springer, 2008.

[36] Jing Huang, S. Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Image indexing using color correlograms. In *Proceedings of the CVPR '97*, volume 00, pages 762–768, San Juan, Puerto Rico, June 1997. IEEE.

[37] Savvas A Chatzichristofis and Yiannis S Boutalis. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In *WIAMIS'08*, pages 191–196. IEEE, 2008.

[38] Markus A Stricker and Markus Orengo. Similarity of color images. In *IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology*, pages 381–392. International Society for Optics and Photonics, 1995.

[39] Hang Cheng, Xinpeng Zhang, and Jiang Yu. Ac-coefficient histogram-based retrieval for encrypted jpeg images. *Multimedia Tools and Applications*, 75(21):13791–13803, 2016.

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

[41] Koen EA van de Sande, Theo Gevers, and Cees GM Snoek. Evaluating color descriptors for object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1582–1596, 2010.

[42] Rosebrock A. A guide to utilizing color histograms for computer vision and image search engines. 2014.

[43] Ju Han and Kai-Kuang Ma. Fuzzy color histogram and its use in color image retrieval. *IEEE Transactions on image Processing*, 11(8):944–952, 2002.

[44] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002.

[45] Vincent Barra. Expanding the local binary pattern to multispectral images using total orderings. In *International Conference on Computer Vision, Imaging and Computer Graphics*, pages 67–80. Springer, 2010.

[46] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th CIVR '07*, pages 401–408, New York, NY, USA, 2007.

[47] Shih-Fu Chang, Thomas Sikora, and Atul Puri. Overview of the MPEG-7 standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):688–695, June 2001.

[48] Thomas P Weldon, William E Higgins, and Dennis F Dunn. Efficient gabor filter design for texture segmentation. *Pattern recognition*, 29(12):2005–2015, 1996.

[49] Savvas A. Chatzichristofis, Yiannis S. Boutalis, and Mathias Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. In *SPPRA 09'*, 2009.

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

[51] Ajith H. Gunatilaka and Brian A. Baertlein. Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):577–589, 2001.

[52] 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.

[53] Cees GM Snoek, Marcel Worring, and Arnold WM Smeulders. Early versus late fusion in semantic video analysis. In *Proc. of ACM MM*, pages 399–402, 2005.

[54] Konstantin Pogorelov, Sigrun Losada, Carsten Griwodz, Thomas de Lange, Kristin Ranheim Randel, Duc Tien Dang Nguyen, Håkon Kvale Stensland, Francesco De Natale, Dag Johansen, Michael Riegler, and Pål Halvorsen. A holistic multimedia system for gastrointestinal tract disease detection. In *Proc. of MMSys*, 2017.

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

[56] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.

[57] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.

[58] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

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

[60] DRGHR Williams and Geoffrey Hinton. Learning representations by back-propagating errors. *Nature*, 323(6088):533–538, 1986.

[61] Konstantin Pogorelov, Kristin Ranheim Randel, Carsten Griwodz, Thomas de Lange, Sigrun Losada Eskeland, Dag Johansen, Concetto Spampinato, Duc-Tien Dang-Nguyen, Mathias Lux, Michael Riegler, and Pål Halvorsen. Kvasir: A multi-class image-dataset for computer aided gastrointestinal disease detection. In *Proc. of MMSYS*, 2017.

[62] Konstantin Pogorelov, Kristin Ranheim Randel, Thomas de Lange, Sigrun Losada Eskeland, Dag Johansen, Carsten Griwodz, Concetto Spampinato, Mario Taschwer, Mathias Lux, Michael Riegler, and Pål Halvorsen. Nerthus: A bowel preparation quality video dataset. In *Proc. of MMSYS*, 2017.