Jimma University

School of Graduate Studies

Jimma Institute of Technology

School of Biomedical Engineering

MSc. In Biomedical Engineering (Biomedical Imaging Stream)

A Master's Thesis Report On

A Comparative Study of Texture Descriptors for Polyp Detection in Colonoscopy Images

A thesis report submitted to the School of Graduate Studies of Jimma University in partial fulfillment of the requirements for the Degree of Master of Science in Biomedical Engineering (Biomedical Imaging)

Author:
Yordanos Kahsay

December, 2018 G.C
Jimma, Ethiopia

Jimma University

School of Graduate Studies

Jimma Institute of Technology

School of Biomedical Engineering

Biomedical Imaging Stream

A comparative study of texture descriptors for polyp detection in colonoscopy images

A final report submitted to the School of Graduate Studies of Jimma University in partial fulfillment of the requirements for the Degree of Master of Science in Biomedical Engineering (Biomedical Imaging)

Author:
Yordanos Kahsay

Advisor:
Kinde Anlay (Ph.D.)
Co-advisor:
Mohammed Aliy (M.Sc.)

December, 2018 G.C
Jimma, Ethiopia

Declaration

I, the undersigned, declare that this thesis entitled: **"A comparative study of texture descriptors for polyp detection in colonoscopy images."** is my original work, and has not been presented by any other person for an award of a degree in this or any other university, and all sources of material used for this research have been duly acknowledged.

Candidate:

Yordanos Kahsay Signature: _______________ Date_______________

On behalf of the School of Biomedical Engineering at Jimma Institute of Technology, we the advisors of this research with the title of **"A comparative study of texture descriptors for polyp detection in colonoscopy images."** and I, the evaluator, confirm that this research is approved as MSc. Thesis for the student.

Advisors

Name: Dr. Kinde Anlay Signature: ______________ Date_______________

Name: Mr. Mohammed Aliy Signature: _______________ Date________________

External Examiner

Name: ______________________ Signature: ______________ Date_______________

Internal Examiner

Name: ______________________ Signature: __________________ Date_______________

Abstract

Cancer has become the second leading cause of death in the adult population of Ethiopia. Colon cancer is cancer of the large intestine (colon), most cases of colon cancer start as small, noncancerous clumps of cells called adenomatous polyps. Over time some of these polyps can become colon cancers. Polyps may be very small and show few symptoms. Colon is part of the large intestine, and it belongs to our body's digestive system. It reabsorbs large quantities of water and nutrients from undigested food products as they pass through it. Therefore, physicians recommend regular screening test in order to prevent colon cancer by identifying and removing polyps before they become cancer.

Colonoscopy is the first method to detect and remove polyps. The accuracy of polyp detection depends on the attentiveness and experience of the endoscopist during the procedure. But computer aided algorithms helps to increase the accuracy of polyp detection .Texture descriptors are one of the methods used to detect colon polyp. Texture is a property that represents the surface and structure of an image. The motivation behind using texture information for polyp detection is that the polyp has different color, shape, size, and appearance. Various texture descriptors are used for polyp detection. However, no work has been reported on the performance comparison of texture descriptors on publicly available polyp dataset and which combines several texture descriptors to improve the accuracy of automatic polyp detection systems even though improvement was reported for natural image datasets. In this research, the performance of the texture descriptors is studied in isolation and by combining multiple of them on recently available mayo clinic large polyp dataset using MATLAB 2018a. The dataset contains 18,500 polyp images with their ground truth image masks. The optimal single descriptor and combination of the texture descriptors which can achieve high classification rate are determined. In this research, the combination of the Wavelet transform, Local binary pattern and grey level co-occurrence matrix gives the highest classification accuracy of 93.74%. The sensitivity and specificity results are 0.934 and 0.9425 while using Support vector machine as a classifier.

Key words: Colon cancer, Polyp, Texture, classification

Acknowledgement

Praise and glory be to God for his grace and granted guidance throughout my life. I would like to express my sincere gratitude to my main advisor Dr. Kinde Anlay for his great help, limitless efforts and for providing me basic materials through my work. And my gratitude also goes to my co-advisor Mr. Mohammed Aliy for his support, feedback and guidance. I would also like to acknowledge Dr. Timothy Kwa and Mr. Hakkins Raj for their comment and support.

I wish to express my respect to my family and who made everything possible to succeed in my life. I want to express my gratitude to Jimma University for the academic sponsorship of my study. I would like to thank School of Biomedical Engineering, Biomedical Imaging stream, and my friends who have helped me.

Table of Contents

List of figures

List of tables

List of abbreviations

CRC	Colorectal Cancer
GLCM	Gray Level Co-occurrence Matrix
HSV	Hue Saturation Value
ISBI	International Symposium on Biomedical Imaging
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
RGB	Red Green Blue
SIFT	Scale Invariant Feature Transform
SVM	Support Vector Machine
TSH	Texture Spectrum Histogram
WCE	Wireless Capsule Endoscopy
WT	Wavelet Transform

CHAPTER ONE

Introduction

1.1 Background of the study

Medical diagnosis is performed by obtaining information from different sources such as the result of clinical examinations, histological findings, and patient history. There are also different data physicians consider in order to reach a final diagnostic decision. Imaging techniques have been used in the last decades, and they help experts in decision making [1].

Colon cancer is cancer of colon cells that arise from mucosal colonic polyps. A polyp is the overgrowth of cells that line the inner colon wall. It arises from mucosal colonic polyps. Polyps grow into surrounding tissues if they are not treated. The two common histologic types of polyps are hyperplastic and adenomatous. They can be removed and tested for cancer [2].

Colonoscopy is a medical device used to examine the condition of the colon. The development of inventive methods for the identification of colon status is explored as a computer-aided tool for the early detection of colon cancer. Computer-assisted image analysis extracts the representative features of images together with quantitative features of images together with quantitative measurements. Extracting quantitative parameters to represent the characteristics properties of the colon from colonoscopy images is essential. The quantitative features are used for detecting the normal or abnormal conditions of a colon. This measurement contributes to the interpretation by colonoscopy expert [3].

Colonoscopy images are used to define features of the normal and abnormal colon. It is a method used to detect and remove polyps which are precursors to colon cancer [3]. Colonoscopy image contains rich information of texture and color. This information provides good results for the image analysis which are better than intensity information. Texture analysis is an important feature used in image processing and pattern recognition. It gives information about the spatial property and arrangement of essential image elements. They are specific methods like Texture Spectrum Histogram (TSH), Gray Level Co-occurrence Matrix (GLCM), Wavelet Transform (WT) and Local Binary Pattern (LBP) these techniques can be used for natural images and medical images.

1.2 Statement of the problem

Colon cancer is cancer of the large intestine (colon), which is the final part of our digestive tract. It happens when tumorous growths develop in the large intestine. Most cases of colon cancer begin as noncancerous (benign) clumps of cells called adenomatous polyps. Over time some of these polyps can become colon cancers. Polyps may be small in size and produce few symptoms.

Physicians advice regular screening test to prevent colon cancer by identifying and removing polyps before they become cancer. Colonoscopy is the primary method for detecting and removing polyp indicators to colon cancer. It is an effective procedure in the process to decrease the incidence and mortality of colon cancer. The polyp has different color, shape, size, and appearance. This variation makes the usage of geometry based feature descriptors inefficient for polyp detection. Texture has been widely used as the essential cue for the identification of polyps in colonoscopy images. Texture analysis is one of the most important features used in image processing. It can give information about the arrangement and spatial properties of fundamental image elements. No work has been reported on the performance comparison of texture descriptors on publicly available large polyp dataset and on the combination of several texture descriptors to improve the accuracy of automatic polyp detection systems even though certain improvement was reported for natural image datasets.

.

1.3 Objectives

1.3.1 General objective

The main objective of this research is to evaluate texture extraction methods and to combine multiple texture descriptors to increase the performance of colonic polyp detection.

1.3.2 Specific objectives

The specific objectives of the research are

- To compare the effectiveness of many texture descriptors.
- To evaluate the efficiency of texture extraction methods for classification of the polyp.
- To evaluate the multiple combinations of texture descriptors.
- To improve colonic polyp detection.
- To select the best classifier and best method.

1.4 Significance of the thesis

This research was mainly focused on a comparative study of texture descriptors for detection of frames with a polyp in a colonoscopy images. It compared the effectiveness of many texture features on publicly available labeled colonoscopy database. It evaluated and compared the efficiency of texture extraction methods for classification of the polyp. It compared each method in isolation by using Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) classifiers. Moreover, also the multiple combinations of these methods were evaluated. Finally, best performing texture descriptor and best performing classifier were proposed for colonic polyp detection which will make significant improvement in accuracy and efficiency of diagnosing and treating colon polyps before they turn in to colon cancer.

1.5 Scope of the thesis

The scope of this thesis was up to exploring colonic polyp detection methods and developing an algorithm by evaluating the efficiency of different texture extraction methods using image acquired from Mayo-clinic image dataset and comparing the effectiveness of many texture features. In this research, multiple combinations of texture descriptors were evaluated to select the best classification and best feature extraction method.

1.6 Organization of the thesis

To understand the work easily this thesis in segregated into five chapters**.** From **chapter 1,**

The reader will get the overall introduction, the base problem, the purpose and significance of this research thesis. The next chapter, **Chapter 2**, discusses the physiological background of the digestive system, explains medical image analysis, texture analysis, machine learning algorithms and classification methods. Literature survey that discusses the works performed in the area of feature extraction and classification are also covered in this chapter. **Chapter 3** explains the materials and methods used in the research. The tests performed in this thesis are discussed and analyzed in detail in **Chapter 4.** The last chapter, **chapter 5,** summarizes all the chapters, discusses the main achievements of the research and leaves a clue to be addressed in the future. To make the reader fully satisfied codes are included in the form of Appendix at the end of the document.

CHAPTER TWO

Digestive system and Image Analysis

2.1 Digestion

Digestion is the process to break down food into pure chemical substances that can be used as nutrients by the body. Most of the contents in our diet must be broken into smaller particles, and then they can be absorbed into the blood and distributed to different parts of the body for usage. The digestive process is performed by mechanical and enzymatic breakdown of food into simpler chemical compounds. An ordinary young healthy adult devours about 1 kg of solid diet and about 1up to 2 liter of liquid diet per day. All these food materials went through the digestive process, before being absorbed into the blood and distributed to the tissues of the human body: such as carbohydrates into mono saccharides, proteins into amino acids, and triglycerides into fatty acids and glycerol. Digestive system plays a significant role in the digestion and absorption of food substances. As illustrated in figure 1 and 2 Digestive system consists of digestive tract and associated accessory organs. The functions of the digestive system are ingestion, mastication, propulsion, mixing, secretion, digestion, absorption and elimination [4, 5].

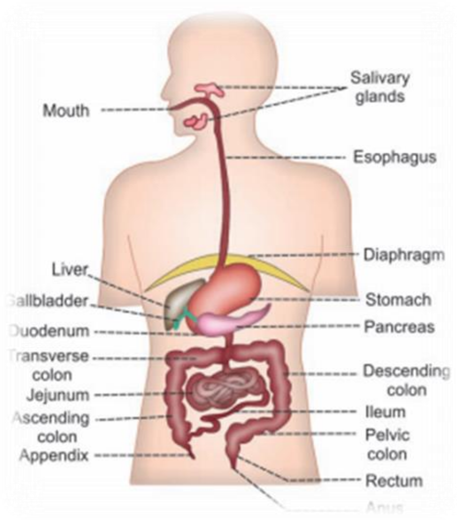


Figure 1 Gastrointestinal Tract *[4]*

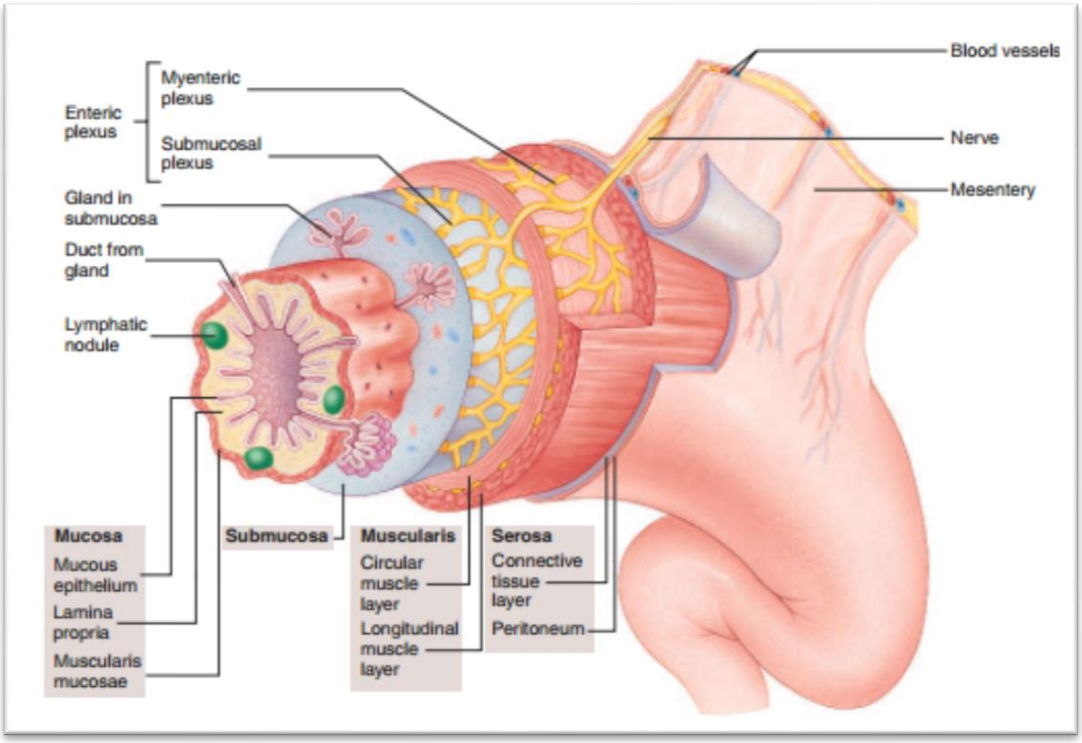


Figure 2 Digestive Tract Histology *[5]*

2.2 Large Intestine

The large intestine is part of the digestive tract extending from the ileocecal junction to the anus. As illustrated in figure 3 it consists of the cecum, colon, rectum, and anal canal. Usually, 18-24 hours are required for a material to pass through the large intestine, but in different 3up to 5 hours are needed for the movement of chyme through the small intestine. Thus, the actions of the colon are slower than those of the small intestine. Although in the colon, chyme is converted to feces. Absorption of water and salts, the secretion of mucus, and extensive action of microorganisms and they are involved in the formation of wastes, which the colon stores until the wastes are eliminated by the process of defecation [5].

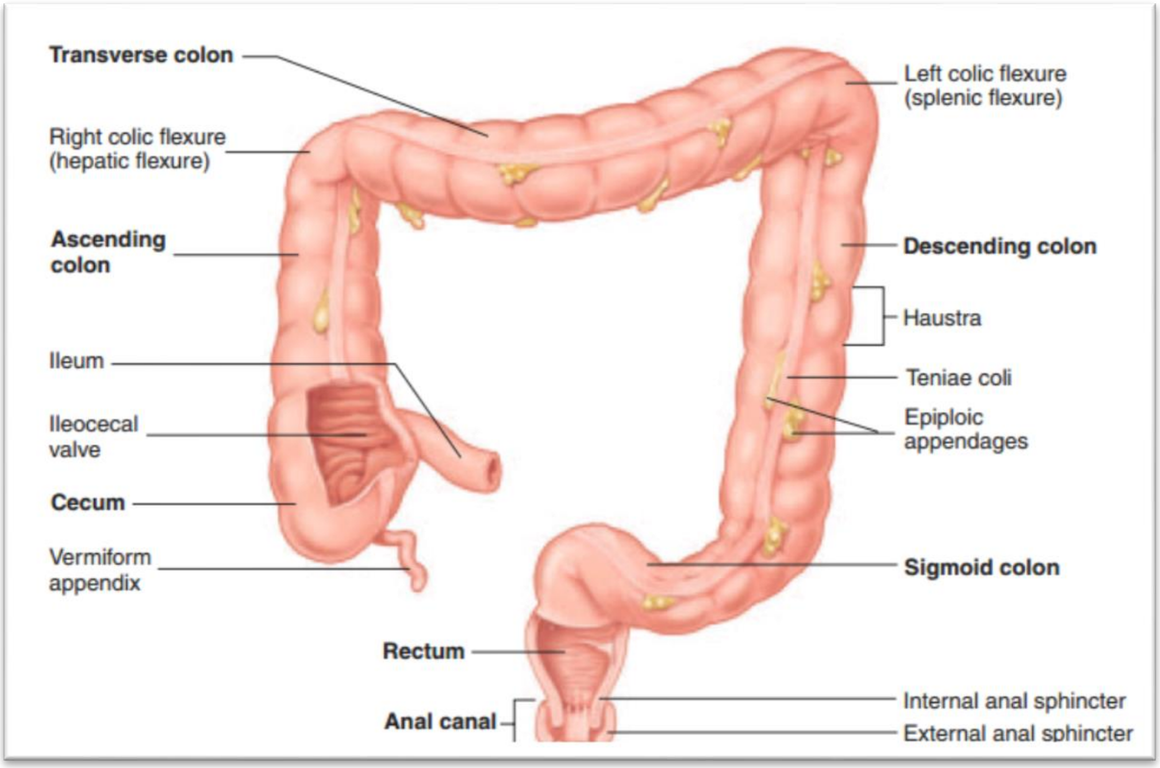


Figure 3 Large Intestine *[5]*

2.3 Anatomy of the large intestine

Anatomy of large intestine consists of cecum, colon, rectum and anal canal. From these parts colon is mainly discussed in this work

2.3.1 Colon

The colon is about 1.5–1.8 m long, and as illustrated in figure 3 it consists of four parts: the sigmoid, descending, transverse, and ascending colon. The ascending colon is located in the above it extends from the cecum and ends at the right hepatic flexure, which is closer to the right corner of the liver. The transverse colon continues from the right colic flexure to the left colic flexure, and the descending colon extends from the left colic flexure to the upper opening of the right pelvis, where it starts the sigmoid colon. The sigmoid colon makes an S-shaped tube that extends into the pelvis and ends at the rectum. The circular muscle layer of the colon is complete, but the longitudinal muscle layer is incomplete. The longitudinal layer does not entirely enclose the intestinal wall, but it forms three bands, called the teniae coli, which is a band along the colon. The mucosal lining of the large intestine consists of simple columnar epithelium. This epithelium is not formed into bends or villi like structures like that of the small intestine, but it has a lot of straight tubular glands called crypts. These small cavities are similar to the intestinal glands of the small intestine; they consist of three types of cells goblet, absorptive and granular cells. The significant difference is the two types of cells are significantly reduced in number, and large intestine goblet cells are more abundant in quantity. Figure 4 illustrates the physiology of normal colon [5].

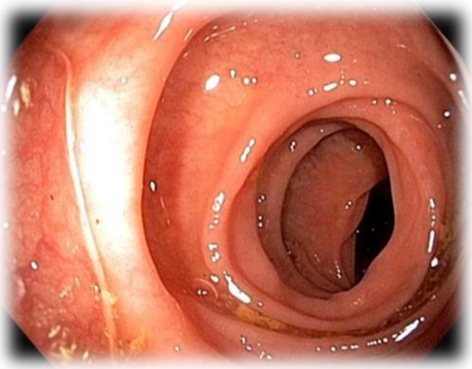


Figure 4 Normal Colon [31]

2.4 Colon polyp

Mariam Webster dictionary defines polyp as "a growth projecting from a mucous membrane (as of the colon or vocal cords)." A colon polyp is the main focus in this work. A colon polyp is extending structure that appears from the colon wall. An example of an image containing colonic polyp, which is taken using optical colonoscopy, is shown in figure 5. Polyps are characterized according to their bleeding tendency, their color, the presence of ulcers, the appearance of their mucosal surface and presence of pedunculus (non-pedunculated or pedunculated). All polyps are not cancerous. The most common type cancer related to polyp is colorectal cancer (CRC). The polyp has a high possibility to develop into colorectal cancer unless it is detected early and removed. Hence early detection and removal of colon polyp reduce the possibility of its advancement into cancerous stages [15].

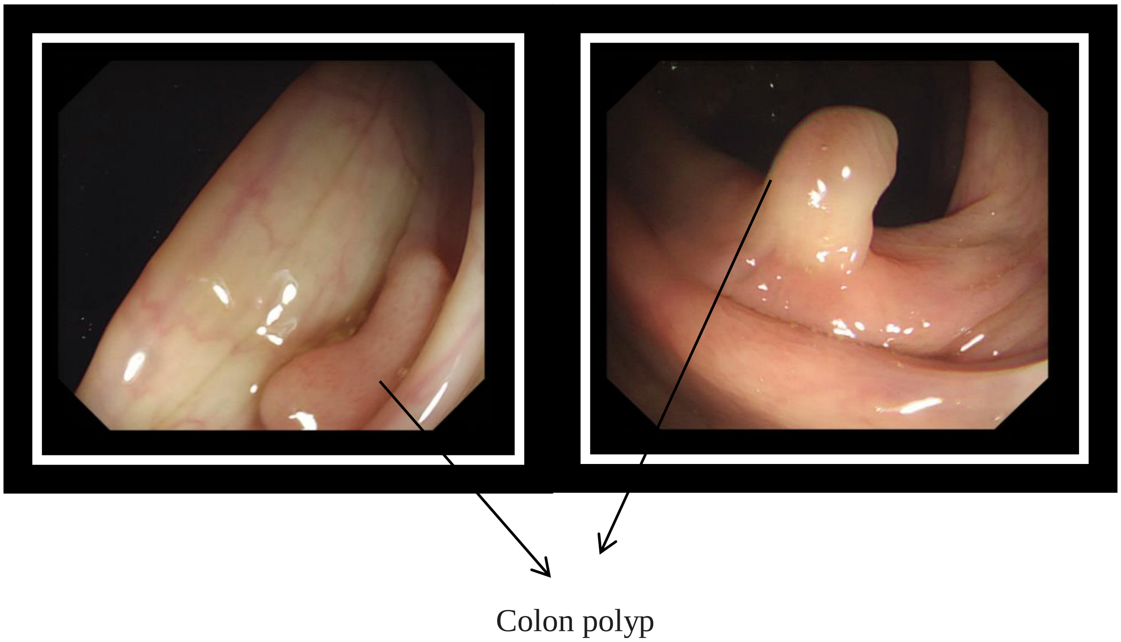


Figure 5 colon polyp [31]

2.5 Colonoscopy

Screening the colon using colonoscopy procedure is the most effective preventing method of colorectal cancer. It has contributed to a 30% reduction of the incidence of CRC. As illustrated in figure 6 the colonoscopy procedure has two phases. The first one is the insertion phase. In this phase, the flexible endoscope is inserted via anus and pushed until it reaches the end of the colon.

When it reaches the end of the colon, the withdraw phase begins. In the second phase, the flexible endoscope is slowly removed while the endoscopist carefully examines the presence of abnormalities such as polyp on the inner wall of the colon in real time of screen connected to the other end of the flexible tube. A polyp found during the procedure is removed regardless of its status. The process takes on the average about 25 minutes [6].

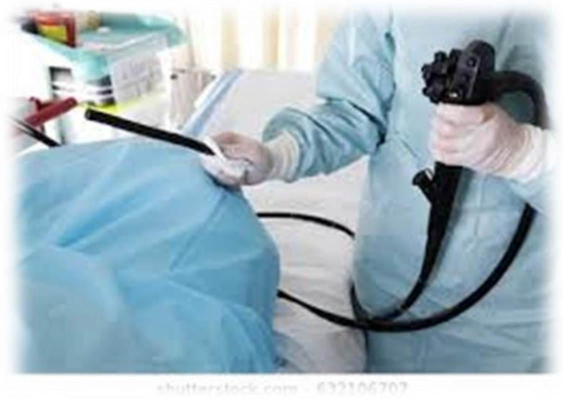


Figure 6 colonoscopy *[32]*

The accuracy of polyp detection depends on experience, fatigue, and attentiveness of the colonoscopist during the procedure. The polyp miss rate by human subject ranges from 4% to12%. A polyp missed at the early stage has high possibility to advance into the cancerous stage, and it can significantly reduce the survival rate of the patient.

Computer-aided polyp detection algorithms can assist the endoscopist during the procedure. It can urge the colonoscopist to give more attention to the regions labeled as polyp by the algorithm as shown in figure 7. Polyps in colon appear in different color, size, shape, and texture which make it quite challenging to detect using classical feature detection methods.

The significant problems of polyp detection in optical colonoscopy video are that there is a high similarity between the polyp region and non-polyp region. There may not exist a sharp boundary between the polyp region and its surrounding. A robust polyp detection algorithm must be able to be invariant to texture, shape, size, and color of the polyp region or capture these different features effectively [6].

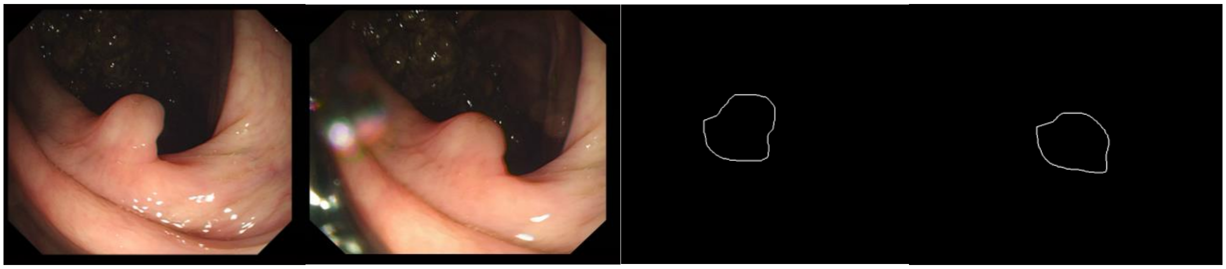


Figure 7 Colonoscopy Images with their corresponding mask labeling abnormal regions [31]

2.6 Medical image analysis

Image analysis methods have an important role in different medical applications. These applications include the automatic extraction of features from an image. It is used for classification tasks, such as distinguishing normal tissue from abnormal tissue. Based on the particular classification task, the extracted features capture morphological properties, color properties, or specific textural properties of the image.

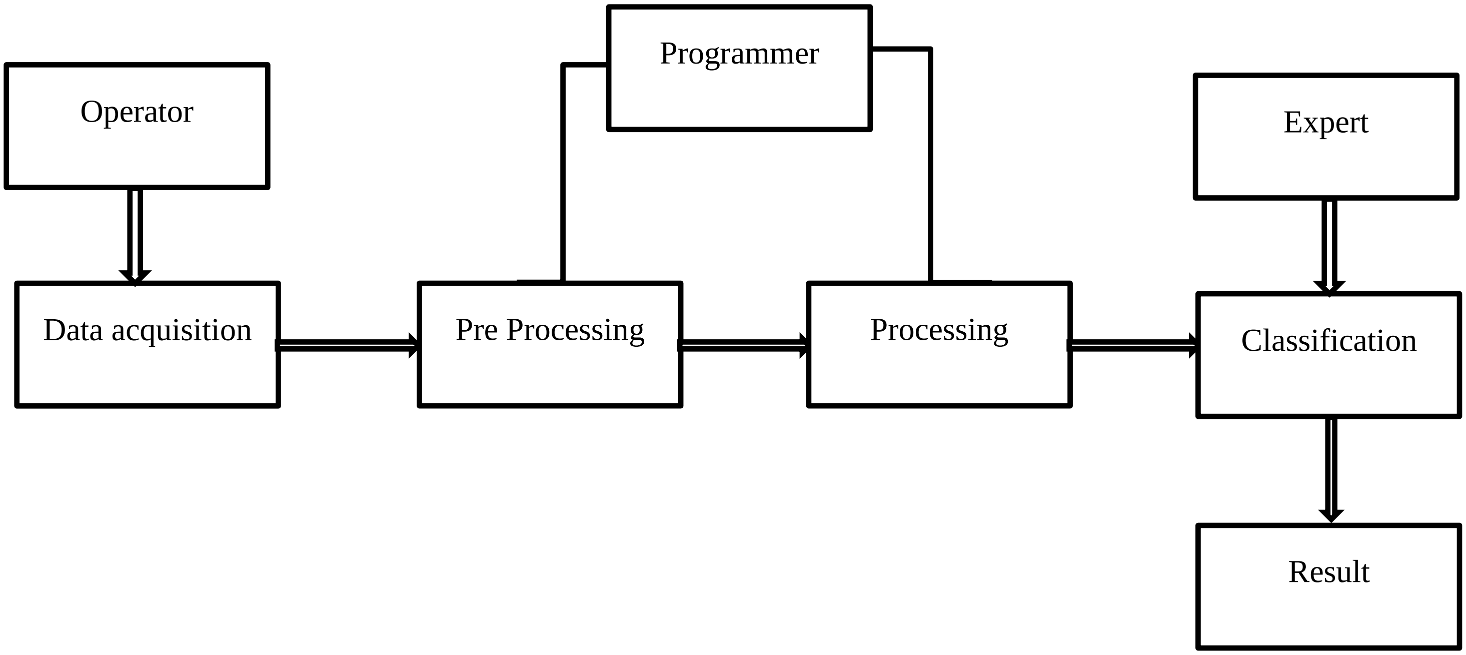


Figure 8 Block diagram of the typical stages of image processing and analysis [7]

The general block diagram for image analysis and processing is illustrated in figure 8 it is the image acquisition, pre-processing, proper analysis, classification, and result. According to the block diagram, an operator is a person operating a medical imaging device, in this case, colonoscopist or endoscopists is an operator, who is also responsible for positioning the person during tests. An expert or gastroenterologist is the person who indicates specific areas in the image and classifies patients (for example, into two classes: "polyp," "non-polyp") based on his/her expert knowledge. A programmer is a person who proposes an appropriate algorithm which enables automatic measurement or classification after training with the expert system [7].

2.7 Texture Analysis

Texture analysis is essential for image processing, computer graphics, and vision. It is quantifying properties like regularity, smoothness, coarseness and measuring the variation in the intensity of a surface. Since it is repeating pattern of this intensity variation, it cannot be defined for a point. It is usually used as a region descriptor in image analysis.

The texture is a property of areas, and in its definition, it must include the gray values in a spatial neighborhood. The size of this neighborhood depends on the type of texture or the size of the members defining the surface. It involves the spatial distribution of gray levels.

Image texture has different qualities which play a significant role in describing the surface. Studies identified that the following properties are essential in describing texture: uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency, and phase. Some of these identified qualities are not independent. For example, density is not independent of frequency, and the property of direction only applies to directional textures. Texture has so many different dimensions. To be adequate for a variety of surfaces, there is no single method of texture representation.

The three methods used to describe texture are statistical, structural and spectral. The composition can be characterized by the statistical properties of the grey levels of the points covering a surface in the case of analytical techniques. These properties are calculated from the grey level histogram or grey level co-occurrence matrix of the surface. The texture is characterized as the combination of elements called "texels" (texture elements) while using fundamental techniques. Texture elements are regularly arranged on a surface according to some rules. Spectral methods describe global periodicity of the grey levels of a surface by identifying high energy peaks in the spectrum since they depend on the properties of the Fourier spectrum.

Various assumptions are made in machine vision and image processing algorithms about the uniformity of intensities in local image regions. Real objects image usually do not show areas of uniform intensities. For example, a wooden surface image contains variations of intensities which form various repeated patterns called visual texture, but the wood surface is not uniform. The models can be the result of reflectance differences such as the color on a surface, or they might be the result of physical surface properties such as roughness or oriented strands which usually have a tactile quality [18].

2.8 Machine learning

Machine learning studies computer algorithm for learning to do activities where humans do naturally. We can learn to accomplish a task, to make correct predictions or to act intelligently. As illustrated in figure 9 it is about learning to do better in the future depending on what was experienced in the past. Learning is based on some observation, instruction, direct experience or data.

The Emphasis of machine Learning is on automatic methods, and the goal is to build learning algorithms that do the learning automatically without human interference or help. It uses computational methods to learn information directly from data without depending on a predetermined equation as a model. The algorithms find natural patterns in data that generate insight and help to make better decision and prediction [26].

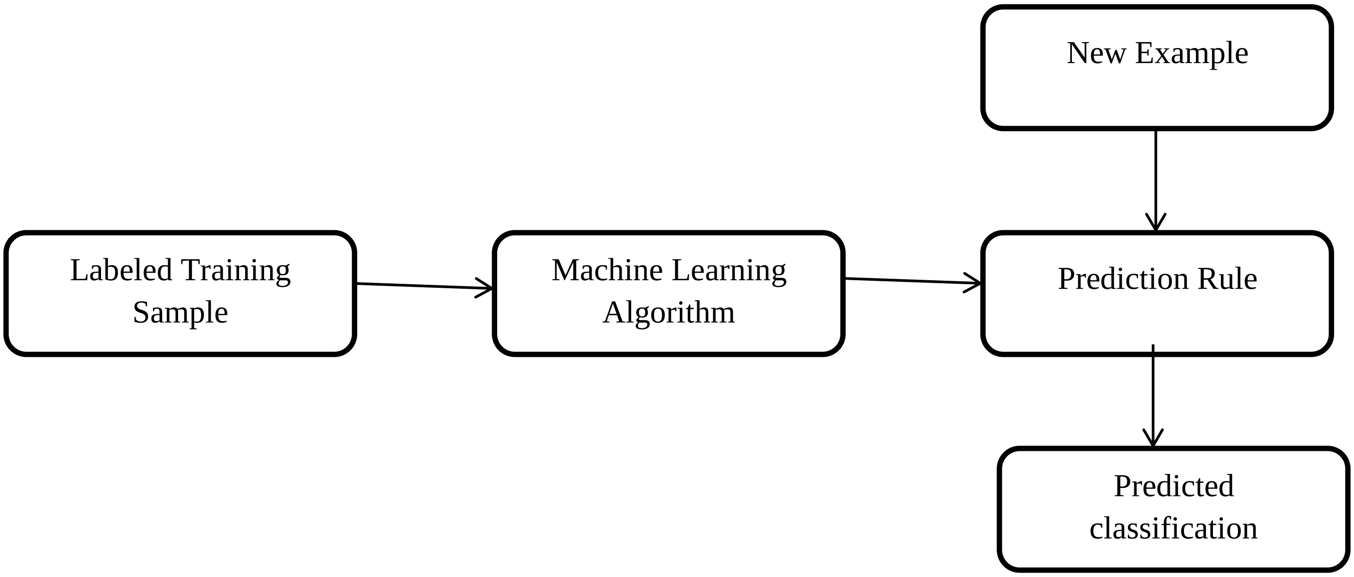


Figure 9 Typical learning problem [2]

As shown in figure 10 Machine learning uses two types of techniques

- Supervised learning which trains a model on known input and output data .so it can predict the future output.
- Unsupervised learning which finds hidden patterns or natural structures in input data.

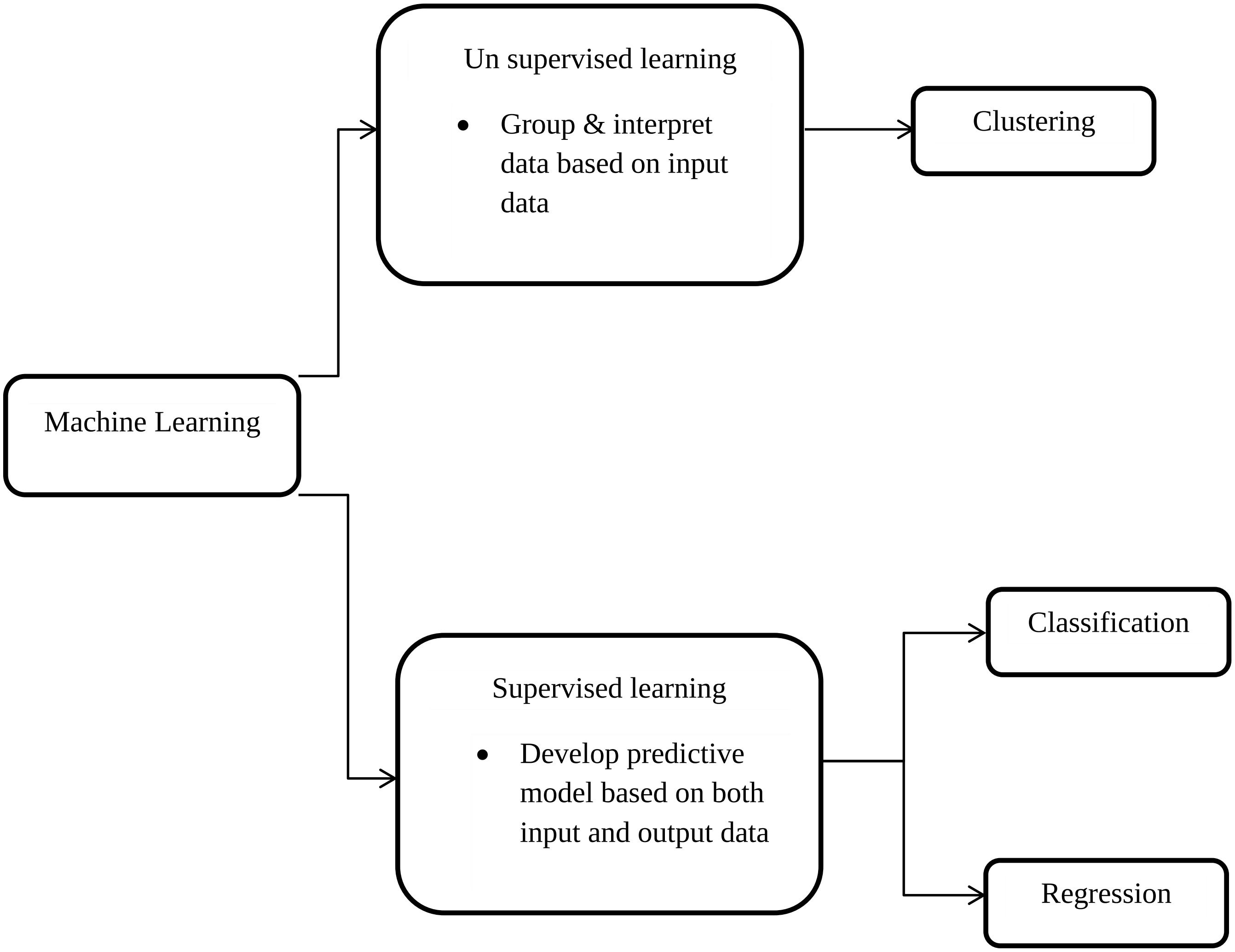

Un supervised learning
Group & interpret data based on input data
Clustering
Machine Learning
Supervised learning
Develop predictive model based on both input and output data
Classification
Regression

Figure 10 Machine learning technique [26]

2.8.1 Unsupervised learning

Unsupervised learning finds hidden patterns or vital structures in data. It is used to draw inferences from datasets that consist of input data without labeled responses.

2.8.1.1 Clustering

Clustering is a widely used unsupervised learning technique. It is used to find hidden patterns or for exploratory data analysis to group data. Gene sequence analysis, market research, and object recognition can be mentioned from the application areas of clustering.

2.8.2 Supervised learning

The objective of supervised machine learning is to construct a model that predicts based on Facts when there is uncertainty. A supervised learning algorithm uses a known set of input data and known output or response to the input data then it trains a model to generate reasonable predictions as a response to new data. Supervised learning uses classification and regression techniques.

2.8.2.1 Classification

Classification models are used to classify into data into categories based on their training. These models can be used in different application areas like medical imaging, credit scoring and speech recognition.

2.8.2.2 Regression

This type of technique is usually used to predict continuous responses like the change in temperature or variation in power demand. Its typical application area includes electricity load forecasting [26].

2.9 Related works in the area of texture analysis

Different texture descriptors capture a different facet of the image texture[29]. M. Sharma et al. evaluated the performance of autocorrelation, edge frequency, primitive-length, Law's method and co-occurrence matrices on Meastex database. It is a database of natural images. The co-occurrence matrices based texture descriptors are reported to give the highest classification performance on this database. The authors have shown that combining different texture descriptors improves the performance of the classification.

A.Barley et al. [8] compared the performance of the co-occurrence matrix, Gabor wavelets steerable pyramids and SIFT on Brodatz, UIUCTex, and KTH-TIPS publically available texture image datasets. They have evaluated these texture descriptors both in isolation and by combining several of them. They have reported that Gabor wavelets based texture descriptors give the highest recognition performance on Brodatz and KTH-TIPS datasets. It was also said SIFT gives the highest classification accuracy for UIUC dataset. Different classification performance was reported when various descriptors are combined. The combination of steerable pyramids and Gabor wavelets gives the highest classification accuracy on Brodatz and KTH-TIPS datasets. Moreover, the combination of steerable pyramids and SIFT gives the highest recognition for UIUCTex dataset. By combining several texture descriptors up to 24%, performance improvement has been reported as compared to classification using a single texture descriptor [9]. Generally, it is challenging to choose the best combination of texture descriptors for a range of datasets. This is because of the different texture aspects which are possessed by different datasets. Hence, the combination of texture descriptors which are suitable for a particular dataset has to be experimentally determined. As far as classification of polyp is concerned, polyp has different texture facet as compared to other texture datasets, such as asphalt, concrete, and grass, which are used in the above works. The literature review of the texture descriptors employed for gastrointestinal polyp detection is given as follows. Wavelet-based uniform local binary pattern (LBP) is used in [9]. The wavelet transform decomposes an image into high frequency and low-frequency components. The high-frequency components contain mainly the edges in the images. These patterns are captured using LBP as feature descriptors.

Iakovidis et al. [10] presented the comparative study of the texture spectrum histogram, texture spectrum, and color histogram statistics, Local binary pattern histogram and color wavelet covariance for discrimination of gastric polyps. The highest classification performance was reported by using color wavelet covariance. Their work combines color information with texture information. J. Silva et al. [11]have used co-occurrence matrices statistics for classification of localized polyp regions. First, the polyp candidates are extracted using circle detection on edge image. Then the co-occurrence matrices based second order statistics of the area of interest is computed. Finally, the regions are classified using a machine learning algorithm, adaboost in this case. Kodogiannis and Boulougoura [7] used a local texture extractor on 3 x 3 neighborhood of the pixel. Their method has quite a similarity to LBP except for the shape of the neighborhood which is rectangular in this case. The texture spectrum histogram (TSH) of the neighborhood relationship is applied to HSV and RGB color spaces. Another polyp detection method based texture was proposed in [12]. The work uses the color combines the color and texture information using color wavelet covariance. The color image is decomposed into approximate and detail components using a discrete wavelet transform. Each color channel is treated separately. The detail components of the second level discrete wavelet decomposition are taken for computing the second-order statistics measure. The three-level wavelet transform is not needed as claimed in the paper as they did not use it. The covariance these statistical measures are computed between the color channels to produce the final feature descriptor. Linear Discriminant Analysis (LDA) was used for classification. The authors in and the texture information was used to detect small bowel tumors in WCE images. [13] used second-order statistical measures of the wavelet decomposed images components. Specificity of 96.6% and sensitivity of 98.7% were ``reported. The same authors used curvelet instead of wavelet with the algorithm given [16]. A sensitivity of about 97.2 % and specificity of about 97.4% were reported. The methods used multilayer neural network for classification.

These algorithms were mainly applied to a small number of datasets. Performance of the methods needs to be evaluated on a large dataset. Recently a large polyp database was introduced by Mayo clinic as part of ISBI Automatic Polyp Detection Challenge 2015. The dataset contains ground truth image masks for polyp regions. The dataset is freely available. This dataset can be used to compare methods without difficulty.

From the literature survey, it has observed that texture has been used for polyp detection as both global [14] [10] [7] and local feature descriptor [11]. Color and texture information were combined to improve the classification performance [10][7]. No work has been reported which combines several texture descriptors to improve the accuracy of automatic polyp detection systems even though up to 24% improvement was reported for other datasets [13]. No study has been published on the performance comparison of state-of-the-art texture descriptors on publicly available polyp dataset. The comparative study of the texture descriptors might give a clue about the most critical texture features (edges around polyp boundary or the structure of the interior polyp region) that distinguishes the polyp region from the non-polyp region. In this work, the performance comparison of the texture descriptors for polyp classification on publicly available polyp image database was studied[15]. The performance of the texture descriptors was thoroughly studied in isolation and by combining multiple of them. The single optimal descriptor and optimal combination of the texture descriptors which can achieve high classification rate were determined.

CHAPTER THREE

Texture descriptors and classification

3.1 Overview

This section briefly describes the materials and methodology used to achieve the objectives of the research. It gives detail explanation of feature extraction methods or texture descriptors used in this research. The methods are Texture Spectrum Histogram (TSH), Local Binary Pattern (LBP), Gray level co-occurrence Matrix (GLCM) and Wavelet Transform (WT). The working principle of these descriptors is discussed.

Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) are classifiers used in this research. These classifiers are also discussed in detail. The block diagrams in figure 11 and figure 12 illustrates the methodology of the research. In figure 11 the workflow of extracting features from single texture descriptors and classifiers used are shown. The second block diagram figure 12 indicates the workflow to extract features using combined texture descriptors and to classify using two different classifiers.

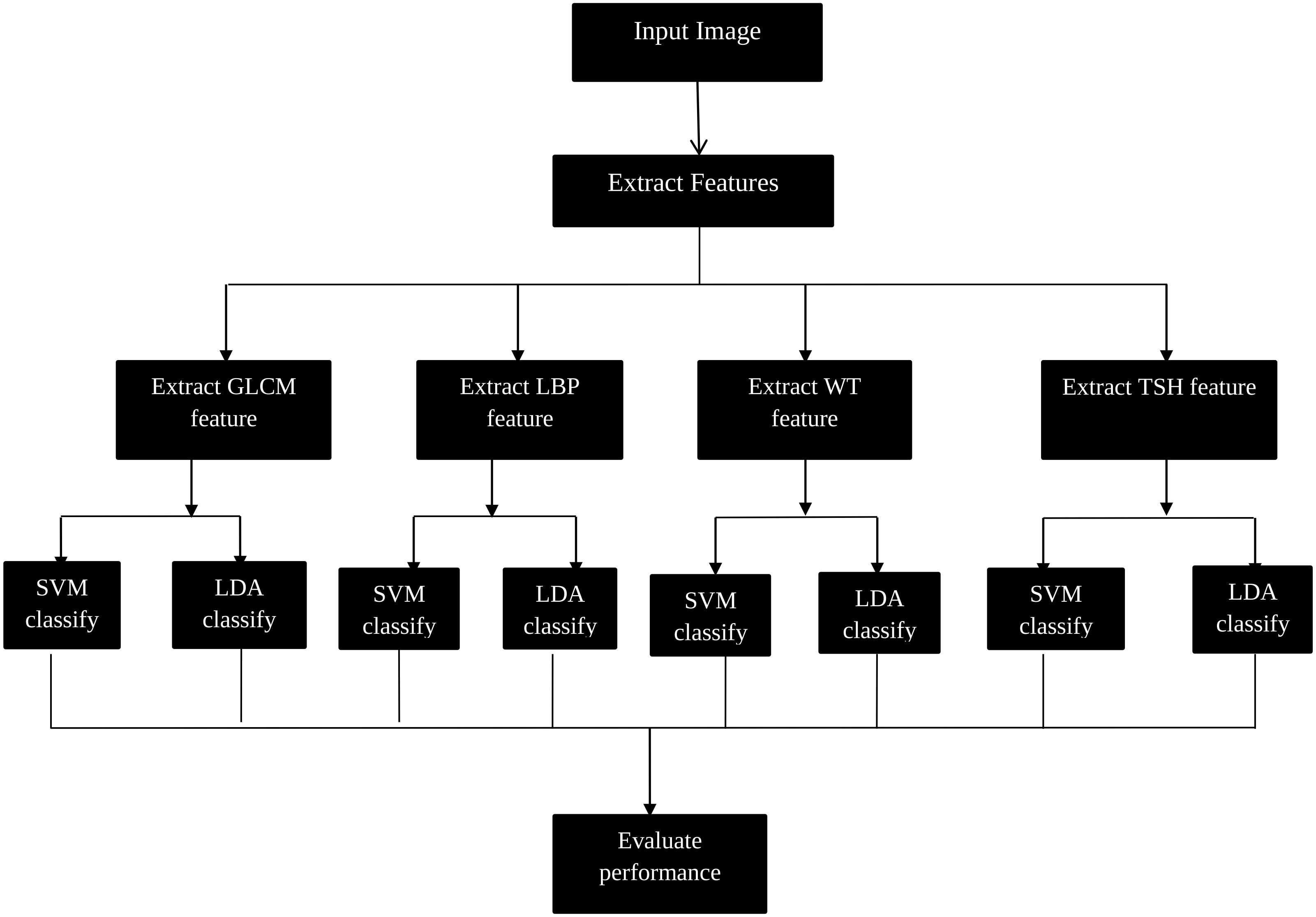


Figure 11 Block diagram of the methodology for single texture descriptors

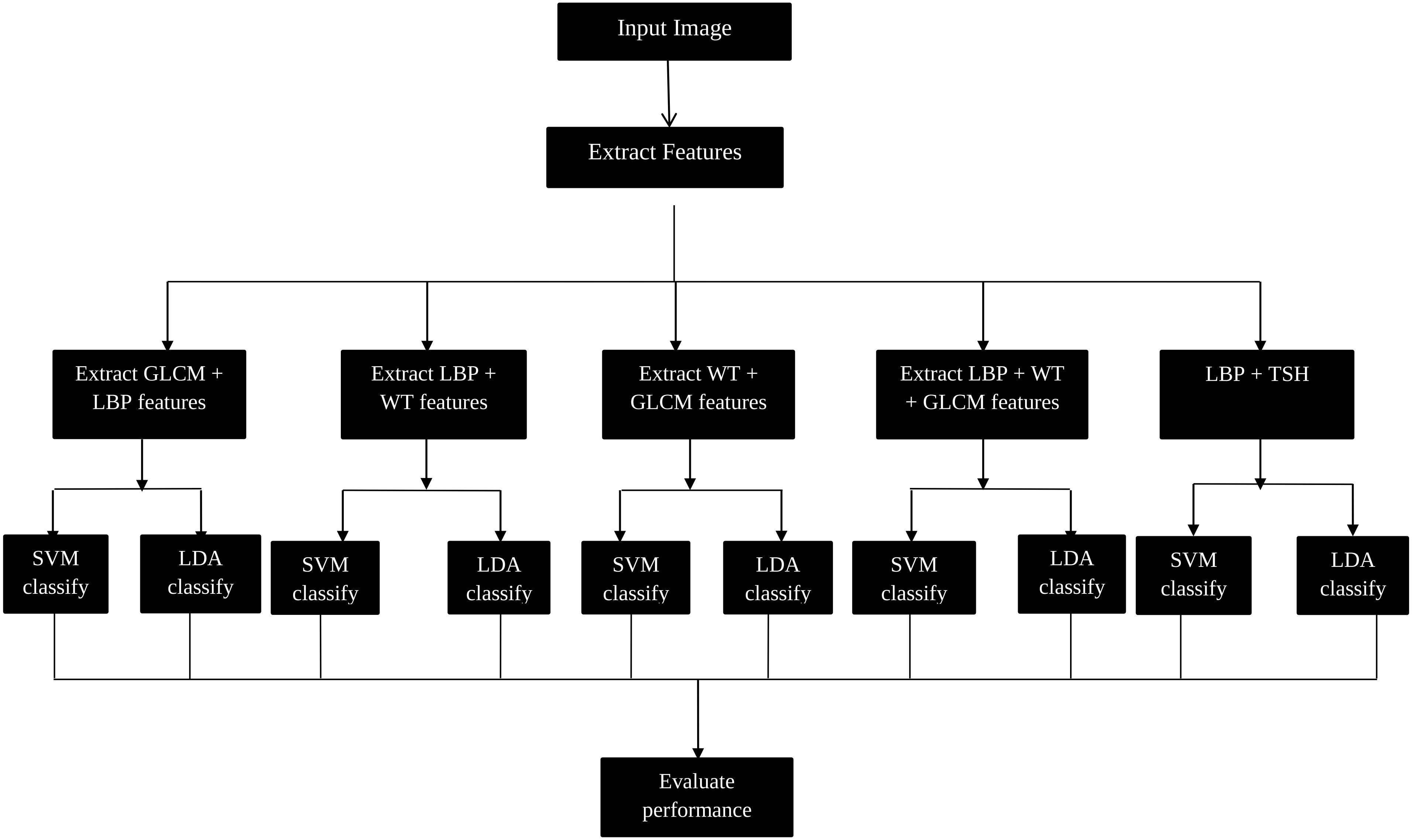


Figure 12 Block diagram of the methodology for combination of different descriptors

3.2 Image Acquisition

In this work, the performance comparison of the texture descriptors for polyp classification on publicly available polyp image dataset is studied. Recently a large polyp dataset was introduced by Mayo clinic as part of ISBI automatic polyp detection challenge 2015. The dataset contains 18,500 images with ground truth masks for polyp and non-polyp regions [31]. Around 10,000 images were used for training and test. The images of normal colon and colon image containing polyp and their labels are illustrated in figure 13 and figure 14 respectively.

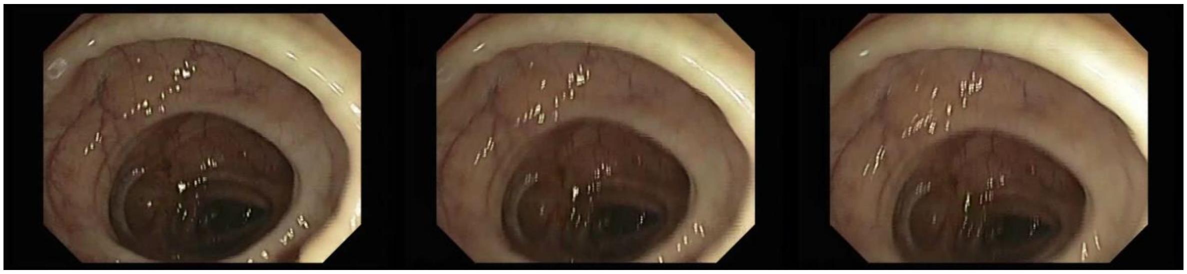


Figure 13 normal colon images

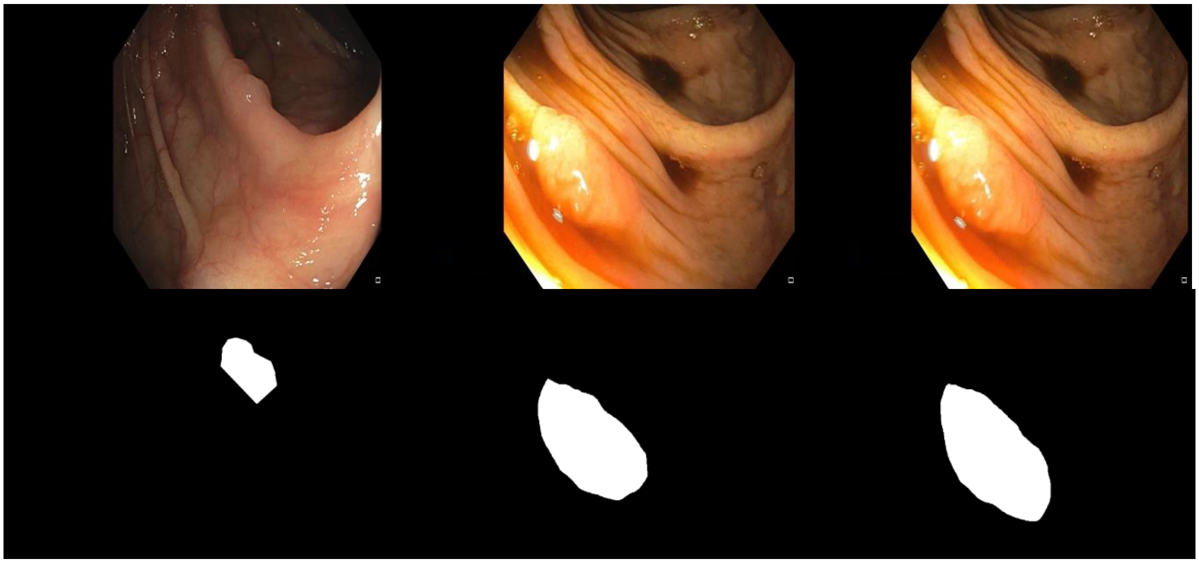


Figure 14 colon image with polyp

3.3 Image Processing and Analysis

After image acquisition image processing and analysis step are continued using different texture descriptors and classification algorithms. The machine and software used in the process are TOSHIBA and MATLAB R2018a version. The working principle of each texture descriptor is discussed.

3.4 Texture descriptors

There are different types of texture descriptors which are used for the identification of polyps in the colonoscopy images. The reason for using texture information is since polyps have different color, shape, size, and appearance. In this paper among the texture descriptors Texture spectrum histogram, Local binary pattern, gray level co-occurrence matrix and wavelet transform were discussed.

3.4.1 Texture Spectrum Histogram

Digital filtering techniques are usually used in digital image processing and pattern recognition. They have an essential role in the set of image transformation. From the application edge noise suppression, detection, recognition, smoothing, and enhancement of images can be mentioned. These techniques are divided into linear and non-linear filters. Based on conventional digital filtering the primary linear filters are low pass, high pass, and band pass. They can be combined to design a wide variety of filters. To filter a signal means to modify its Fourier spectrum or to eliminate or attenuate some unwanted frequency components and transmit others without alteration. Undesired results may be obtained when applying these filters to texture analysis of an image because we need some specific spatial filters which can transform an image in the sense of texture rather than spectral properties and texture analysis depends on the relative intensity relations between the pixels in a small neighborhood, not in their absolute intensity values and the spatial relationship of pixels in a large scale. Since it is difficult to represent or describe efficiently spatial relationship of texture only using a few spatial frequency components of the Fourier spectrum. Therefore some specific spatial filtering techniques named as textural filtering is required. In recent studies texture spectrum methods for texture analysis are introduced in this statistical approach, those studies indicate that its texture spectrum can characterize an image [19, 20, and 21].

3.4.1.1 Texture Units

In a digital image, local texture information can be extracted using texture unit since each pixel is surrounded by eight neighboring pixels the texture information of a pixel can be extracted from a neighborhood of 3x3 pixels. This neighborhood contains a set of nine elements**:**
$V = \{VO, V1 \dots., V8\}$

VO Is the intensity value of the central pixel and $V_i (i = 1,2, \dots, 8)$, is the intensity value of pixel i, which is the neighboring pixel. Texture Unit is defined by a set containing eight elements, $TU = \{E1, E2, \dots E8\}$.

Where, the following formula determines E_i

$$E_i = \begin{cases} 0 \ if \ Vi < (V_0 - \Delta) \\ 1 \ if \ (V_0 - \Delta) < V_i \ \leq (V_0 + \Delta) \\ 2 \ if \ V_i > (V_0 + \Delta) \end{cases}$$

Δ - represents a small positive value

The element E_i holds the same position as the pixel i. Each element of TU has three possible values, the combination of all the eight elements result is $3^8 = 6561$, which is possible Texture units.

3.4.1.2 Labeling Texture Units

The 6561 texture units (NTU) are labeled by using the following formula

$$NTU = \sum_{i=1}^{8} \text{Ei} \ \times 3^{i-1}$$

E_i - is the ith element of the texture unit set

$$TU = \{E1, E2, \dots, E8\}$$

The eight elements can be ordered in a different position. As shown in table 1 the eight elements are ordered in a clockwise direction. When these elements are ordered in a clockwise direction the first element may take eight possible positions from top-left (A) to middle-left (H), then the 6561 Texture Units can be labeled by the above labeling formula under eight different ordering ways (from A to H) [19] [20] [21].

A	B	C
H		D
G	F	E

Table 1 Order of elements

The first element E1 may take eight possible positions from A to H. Transforming an image neighborhood to a texture unit under the ordering method A and with Δ = 0 is shown in table 2.

Neighborhood

63	28	45
88		35
67	40	21

→

Texture units

2	0	2
2		0
2	1	0

↓

$$V = \{40,63,28,45,35,21,40,67,88\} \rightarrow TU = \{2,0,2,0,0,1,2,2\} \rightarrow N_{TU} - 6065$$

Table 2 Transforming to texture unit

3.4.1.3 Texture Spectrum

The 6561 set of Texture Units describes the local texture aspect of a given pixel; it is the relative grey level relationships between the central pixel and the eight neighbors. The statistics on the frequency of occurrence of all the Texture Units over a whole image should show texture information. Texture Spectrum is the frequency function of all the Texture Units, with the x-axis indicating Texture Unit number NTU and the y-axis representing its occurrence frequency.

This texture unit based texture spectrum was proposed in1990, and it uses texture analysis, including texture edge detection, texture characterization, texture classification, and textural filtering. However, recent studies show that a large number of texture units and redundancy is the disadvantage of using this method. Also, reducing the number of texture units without significant loss of discriminating power is proposed.

In the early studies, texture units described as a central pixel with its neighboring pixels in all eight directions to construct 3×3 grids. The grey-level differences of the central pixel and its eight neighbors are simplified to three situations. Moreover, the histogram of these texture units with a moving window gives us the texture spectrum. This texture spectrum with a dimension of $3^8 = 6561$ is the unique feature to characterize the image's texture information. However, recent studies indicate that the calculations in eight directions are probably redundant and the texture spectrum is defined in four directions, 0°, 45°, 90°, and 135° can give similar discriminating power, but with a reduced dimension of $3^4 = 81$.

In this case we consider neighborhood in four direction 0°, 45°, 90°, and 135°, which can be taken us a set containing five elements : $V = \{VO, V1, V2, V3, V4\}$ where VO is the grey- level value of the central pixel and $V1, V2, V3, V4$ are the gray-level value neighbors in the four direction. Since the concept of texture spectrum method is to use the relative intensity relations between pixels, instead of their absolute intensity values the corresponding texture unit is defined us asset containing four elements.

$$TU = \{E1, E2, E3, E4\}$$

Where $E1, E2, E3 \; and \; E4$ are determined by comparing the gray level difference between the central pixel. Table 3 shows transforming the four neighborhoods in to texture unit.

V4	V3	V2
	V0	V1

→

E4	E3	E2
		E1

Table 3 Transforming to texture unit

The different relationships can be simplified into 3 situations (=, < and >) which will noted as 1,2 and 6 respectively.

$$EI = \begin{cases} 1 \; if \; |V_0 - V_i| \le \Delta \\ 2 \; if \; (V_i - V_0) > \Delta \\ 6 \; if \; (V_0 - V_i) > \Delta \end{cases}$$

Where Δ represent small positive value, and it is influenced by the image noise and the image grey-level distribution. In this case, it is simplified first by calculating a histogram of all the difference values $(V_0 - V_i)$ for the entire image to be analyzed and then assign the Δ values so that the histogram will be divided into three equal proportions centered around the $(V_0 - V_i)$ = 0 axes. Generally, these recent studies focus on reducing the number of texture units [19] [20] [21].

3.1.1 Local binary pattern

Local binary pattern is one from the methods of analyzing textures. When it is used for an essential measure of image textures, it shows excellent results regarding accuracy and computational complexity. This operator has a unifying approach to the traditionally divergent statistical and structural models of texture analysis since it combines both texture analyses.

The local binary pattern was introduced as a complementary measure for local image contrast .it is invariant to monotonic changes in gray scale and it was supplemented by an independent measure of local contrast [22].

Local binary pattern Algorithm

- As shown in table 4 images are first labeled by thresholding the difference between a pixel and its neighbors using a step function
- For the basic version, Local binary pattern neighbors mean eight direct neighbors of a pixel
- Then the values of pixels in the thresholded neighborhoods are multiplied by binomial weights given to the corresponding pixels as illustrated in table 5
- Finally, values of the products are summed up to obtain LBP number of this neighborhood.

Step 1 Thresholding

6	1	2
7		4
8	9	7

→

1	0	0
1		0
1	1	1

Table 4 Theresholding

Step 2 multiplying thresholded neighborhoods pixel value with the binomial weights given

1	0	0
1		0
1	1	1

→

1	2	4
128		8
64	32	16

→

1	0	0
128		0
64	32	16

Table 5 Multiplying thresholded neighborhoods with corresponding weight

Step 3 summation of the product value gives us LBP

$$LBP = 1 + 0 + 0 + 0 + 16 + 32 + 64 + 128$$

$$LBP = 241$$

3.1.2 Gray level co-occurrence matrix

Gray level co-occurrence matrix, a method introduced by Haralick, was extracted. It is a statistical method used to measure the textural information of images, and it is created from the gray scale image. It considers the spatial relationship of pixels of an image by calculating how often pairs of the pixel with specific values and in a specified spatial relationship occur in an image. Statistical measures are extracted after creating a gray level co-occurrence matrix.

Steps of GLCM calculation

- The image I to be analyzed is a rectangular image with M_x rows and M_y columns.
- Assume that the gray levels appearing at each pixel is quantized to Mg levels

  $L_x = \{1,2,\dots,M_x\}$ Is horizontal spatial domain
  $L_Y = \{1,2,\dots,M_y\}$ Is vertical spatial domain
  $G = \{0,1,2,\dots,M_{g-1}\}$ Is the set of $N_g$quantized gray levels

- The set L_x L_Y is the set of pixels of the image arranged by their row-column design
- Then the image can be represented as a function of co-occurrence matrix that assigns some gray levels.

- The gray level transitions are computed based on the parameter
  - Displacement
  - Angular orientation
- As shown in table 6 By using a distance of one pixel and angles quantized to 45-degree intervals, four matrices of horizontal, diagonal, vertical and second diagonal are used.

3	3	3
1	3	3
1	3	2

(a)

	1	2	3
1	0	0	2
2	0	0	0
3	0	1	3

(b)

	1	2	3
1	0	0	2
2	0	0	0
3	0	1	2

(c)

	1	2	3
1	0	0	2
2	0	0	0
3	0	1	2

(d)

	1	2	3
1	0	0	2
2	0	0	0
3	0	1	2

(e)

Table 6 (a) original matrix (b)0 degree, (c) 45 degree ,(d) 90 degree and (e) 135 degree orientations of the original matrix

From gray level co-occurrence matrix parameters the most common are Energy, Entropy, Contrast, Local Homogeneity, and Correlation.

- Entropy measures the randomness of the intensity distribution, low values for smooth images than for a coarse image
- Energy measures the number of repeated pairs and also measures the uniformity of the normalized matrix
- The contrast feature is a different moment of a matrix and is a standard measurement of the number of local variations present in an image. The higher the value of contrast is, the sharper the structural variations in the image
- Local homogeneity measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal [23] [24].

3.1.3 Wavelet transform

Wavelet theory is used to analyze texture since it provides a capable tool for multi resolution analysis. In the case of wireless capsule endoscopy (WCE) images, the wavelet transform can provide zooming ability and local characterization to better analyze the mucosa of the inner GI tract, by using information at different scales image analysis can be performed. Wavelet transform of an image can be obtained with discrete wavelet transform (DWT).

The DWT is the same with a hierarchical sub-band system where sub-bands are spaced in the frequency domain. For a 2D image, DWT is implemented with a separable filter-bank and an image is convoluted with a low pass filter L and a high pass filter H recursively. Owing to the decomposition of an image using DWT, the image is transformed into four sub-images which are generally denoted as LL, LH, HL, and HH. The LL sub-image is obtained from low pass filtering in both directions, and it looks like the original picture, so it is called the approximation component. The remaining sub-images are called detailed components. The HL is derived from low pass filtering along the vertical direction, and high pass filtering along the horizontal direction, and so has the label HL. The other two sub-images LH and HH have similar explanations. In this study, three levels DWT to each color channel of a colonoscopy image is applied, and table 7 illustrates such a representation of one color channel for this transformation Textural features are better encoded in detailed sub-images. Three levels of DWT were applied to each color channel, i.e., {HL_i, HH_i, LH_i,(i=1,2,3)} as the basis for textural feature analysis [25].

<table>
<tr><td>LL3</td><td>HL3</td><td rowspan="2">HL2</td><td rowspan="3">HL1</td></tr>
<tr><td>LH3</td><td>HH3</td></tr>
<tr><td colspan="2">LH2</td><td>HH2</td></tr>
<tr><td colspan="3">LH1</td><td>HH1</td></tr>
</table>

Table 7 Wavelet decomposition

3.2 Classification

After extracting the texture descriptors such as LBP, GLCM, WT, and TSH descriptors, the data was fed to SVM and LDA classifier both for training and testing. The working principle of those classifiers is discussed below.

3.2.1 Support vector machine

Support vector machine is classified under a supervised machine learning algorithm. It can be used for classification and regression purposes. In this method each data item is plotted as a point in n-dimensional space, n is the number of features that we have. The value of each feature is the value of each coordinate.

The classification is performed by finding the hyper plane that differentiates the two classes very well. There may be many possible linear classifiers that can separate two classes, but the preferred one is that maximizes the distance between it and the nearest data point of each class. This linear classifier is called the best separating hyper plane [27] [28].

In figure 15 there is a small margin which minimizes the distance between it and the nearest data point of each class. So in figure 16, there is a large margin which maximizes the distance between it and the nearest data point of each class.

.

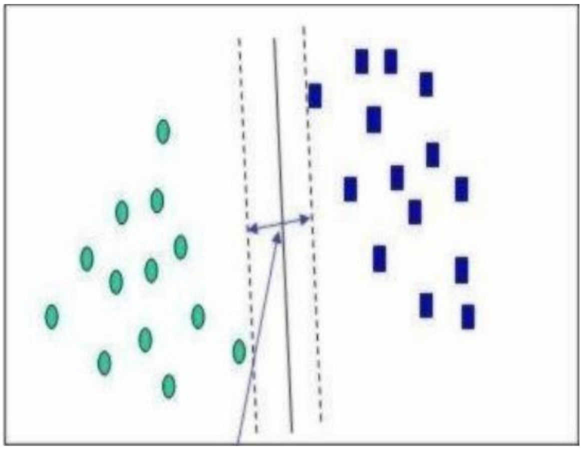


Figure 15 Small Margin

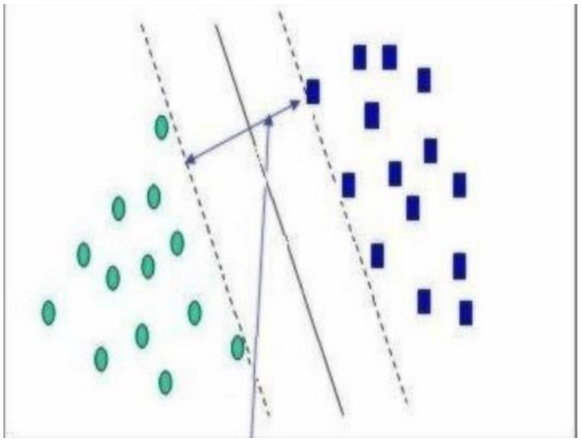


Figure 16 Large Margin

3.2.2 Linear discriminant analysis

Linear discriminant analysis is a commonly used technique for data classification and dimensionality reduction. It keeps classing discriminatory information. This technique efficiently manages the case where the within-class frequencies are unequal, and their performance has been checked on randomly generated test data. This method increases the ratio of between-class variance to the within-class variance in any particular dataset thereby it ensures maximum separability.

Using LDA provides better classification compared to principal component analysis (PCA). The difference between LDA and PCA is that PCA is usually used for feature classification and LDA for data classification. In PCA, the shape and location of the original data set changes when transformed to another space whereas LDA does not change the location, but it gives detail information on class separability and draws a decision region between the given classes. This method also helps to understand the distribution of the feature data. Figure 17 illustrates the theory of LDA showing datasets and test vectors [30].

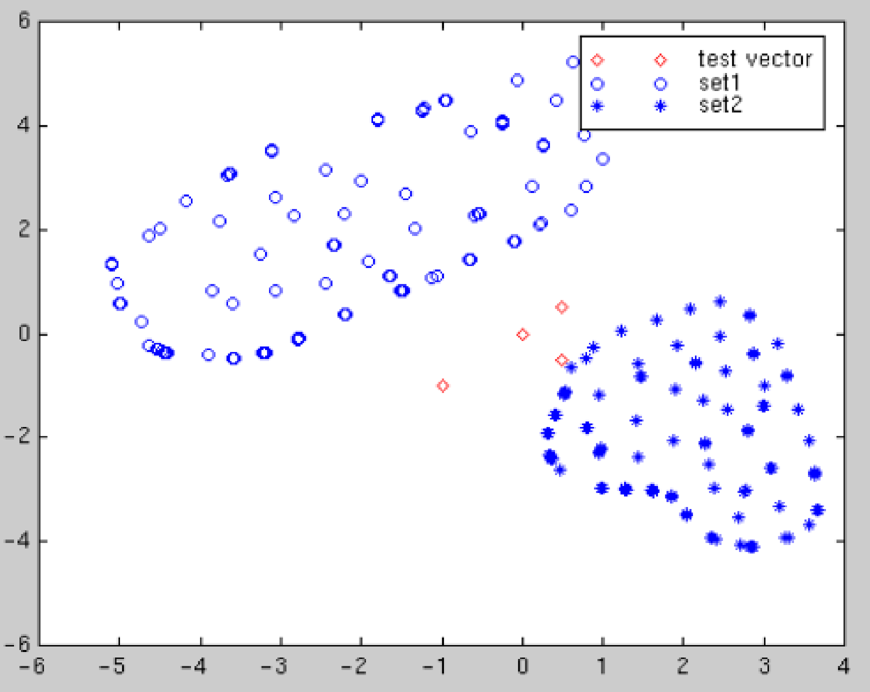


Figure 17 Data Sets and Test Vectors [30]

3.3 Performance metrics

In order to evaluate the performance of automatic polyp detection algorithms, several statistical measures of performance can be used. The objective evaluation of the efficiency of Texture descriptors classification rate is analyzed using the ground truth. The metrics used were classification accuracy, sensitivity, and specificity. Before describing these metrics, here are useful terms to describe them and there result is presented in the form of confusion matrix in chapter 4.

- True Positive (TP) Detection is when the algorithm correctly detects a frame with polyp
- True Negative (TN) Detection is when the algorithm correctly detects a frame without a polyp.
- False positive (FP) detection is when a frame without polyp is labeled as a frame with polyp by the detection algorithm
- False Negative (FN) detection is when a frame with polyp is labeled as a frame without polyp by the detection algorithm

I. Classification accuracy

Classification accuracy is the percentage of correctly classified samples to total sample.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}}$$

Where P is the total number of frames with polyp and N denotes the total number of frames without polyp in the test dataset.

II. Sensitivity

Sensitivity is one parameter to evaluate the rate of true positive predictions. The detection algorithm is

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

III. Specificity

Specificity is the rate of true negative predictions .the detection algorithm is

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

Chapter Four

Results and Discussions

The proposed method has been tested on the publically available dataset. For this thesis Toshiba Laptop (Intel Core i5 with a speed of 1.60 GHz 2.30 GHz, 6GB RAM, and Windows 10 operating system), MATLAB R2018a were testing hardware and software platforms. The chapter presents the result of the algorithm, an objective evaluation of the simulated dataset and qualitative comparison of the proposed algorithms and their results with each other.

4.1 Test results from the dataset

In this work, the performance comparison of the texture descriptors for polyp classification on publicly available polyp image dataset is studied. Recently a large polyp dataset was introduced by Mayo clinic as part of ISBI automatic polyp detection challenge 2015. The dataset contains 18,500 images with ground truth masks for polyp and non-polyp regions. 5000 images were used for training, and 5000 images were used for the test.

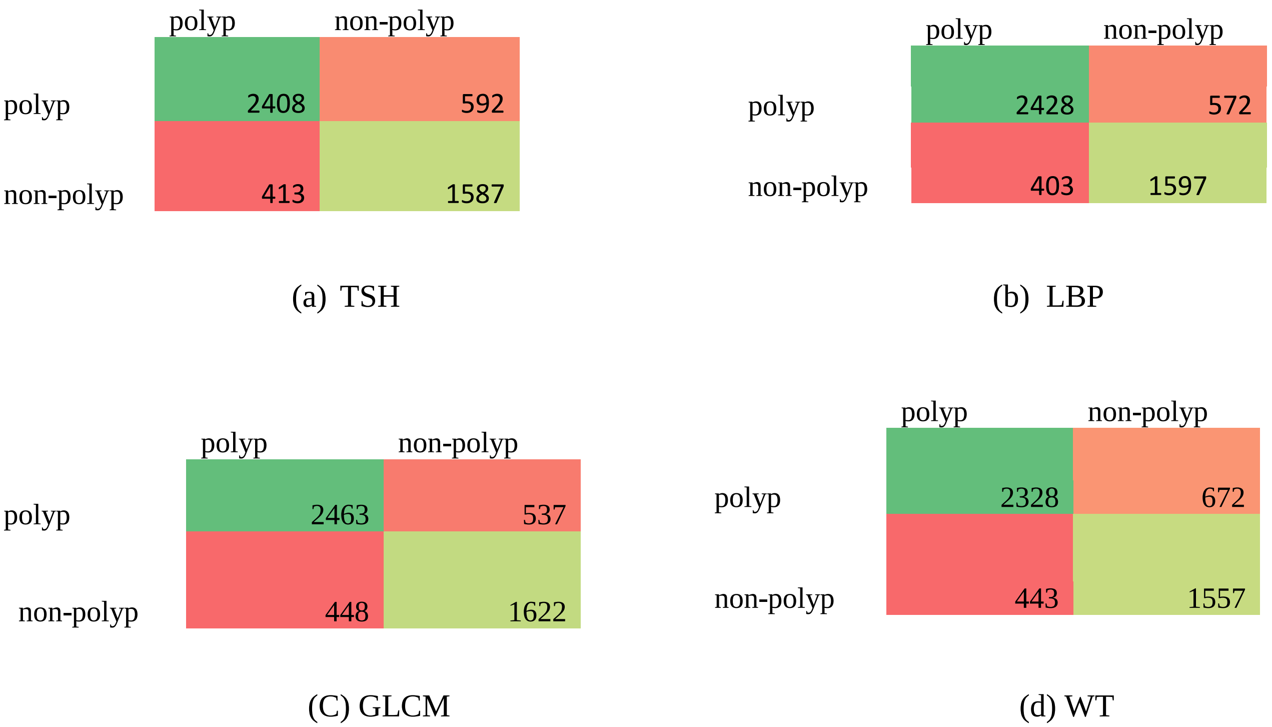
The performance of the texture descriptor is thoroughly studied in isolation and by combining multiple of them. The code used to perform this feature classification is attached in the appendix (from A.1 up to A.22).

4.1.1 Test 1

- Texture descriptors – Texture spectrum histogram (TSH) , Local binary pattern (LBP), Gray level co-occurrence matrix (GLCM) and wavelet transform (WT)
- Classifier – Support vector machine

The classification performance of the texture descriptors is studied using 5000 polyp and 5000 non-polyp images for training and test purpose. In Test one, the classification performance of single texture descriptors using a support vector machine is discussed. Table 8 below illustrates the confusion matrix for each algorithm. The TP, TN, FP, FN, N and P values can be obtained from the table to calculate the accuracy, sensitivity, and specificity of each algorithm as shown in table 9.

.



	polyp	non-polyp
polyp	2408	592
non-polyp	413	1587

(a) TSH

	polyp	non-polyp
polyp	2428	572
non-polyp	403	1597

(b) LBP

	polyp	non-polyp
polyp	2463	537
non-polyp	448	1622

(C) GLCM

	polyp	non-polyp
polyp	2328	672
non-polyp	443	1557

(d) WT

Table 8 the confusion matrix for each single method using SVM

Texture descriptor	Classifier	Classification rate

Texture Spectrum Histogram	SVM	Sensitivity= 80.26%
		Specificity= 79.35%
		Accuracy= 79.9%

Local Binary Pattern	SVM	Sensitivity= 80.9%
		Specificity= 79.85%
		Accuracy= 80.5%

Gray Level Co-occurrence Matrix	SVM	**Sensitivity= 82.1%** **Specificity= 78.35%** **Accuracy= 81.7%**

Wavelet Transform	SVM	Sensitivity= 77.6%
		Specificity= 77.85%
		Accuracy= 77.7%

Table 9 Single Texture Descriptors Output using SVM as a classifier

4.1.2 Test 2

- Texture descriptors – Texture spectrum histogram , Local binary pattern, Gray level co-occurrence matrix and wavelet transform
- Classifier – Linear Discriminant Analysis

The classification performance of the texture descriptors is studied using 5000 polyp and 5000 non-polyp images for training and test purpose. In Test two the classification performance of single texture descriptors using linear discriminant analysis is discussed. Table 10 below illustrates the confusion matrix for each algorithm. The TP, TN, FP, FN, N and P values can be obtained from the table to calculate the accuracy, sensitivity, and specificity of each algorithm as shown in table 11.

	polyp	non-polyp
polyp	2348	652
non-polyp	423	1577

(a) TSH

	polyp	non-polyp
polyp	2398	602
non-polyp	433	1602

(b) LBP

	polyp	non-polyp
polyp	2498	502
non-polyp	433	1567

(c) GLCM

	polyp	non-polyp
polyp	2393	607
non-polyp	518	1482

(d) WT

Table 10 the confusion matrix for each method using LDA

Texture descriptor	Classifier	Classification rate

Texture Spectrum Histogram	LDA	Sensitivity= 78.2%
		Specificity= 78.85%
		Accuracy= 78.5%

Local Binary Pattern	LDA	Sensitivity= 79.93%
		Specificity= 78.72%
		Accuracy= 80.0%

Gray Level Co-occurrence Matrix	LDA	**Sensitivity= 83.26%** **Specificity= 78.35%** **Accuracy= 81.3%**

Wavelet Transform	LDA	Sensitivity= 79.7%
		Specificity= 74.1%
		Accuracy= 77.5%

Table 11 Single Texture Descriptors output using LDA as a classifier

4.1.3 Test 3

- Texture descriptors – Texture spectrum histogram , Local binary pattern, Gray level co-occurrence matrix and wavelet transform
- Classifier – support vector machine

The classification performance of the texture descriptors is studied using 5000 polyp and 5000 non-polyp images for training and test purpose. In Test 3 the classification performance of combined texture descriptors using Support vector machine is discussed. Table 12 below illustrates the confusion matrix for each algorithm.

The TP, TN, FP, FN, N and P values can be obtained from the table to calculate the accuracy, sensitivity, and specificity of each algorithm as shown in table 13.

	polyp	non-polyp
polyp	2518	482
non-polyp	373	1627

(a) LBP + WT

	polyp	non-polyp
polyp	2598	402
non-polyp	313	1687

(b) LBP + GLCM

	polyp	non-polyp
polyp	2478	522
non-polyp	373	1747

(c) LBP + TSH

	polyp	non-polyp
polyp	2538	462
non-polyp	353	1647

(d) WT +GLCM

	polyp	non-polyp
polyp	2802	198
non-polyp	115	1885

(e) LBP + GLCM + WT

Table 12 the confusion matrix for each combined method using SVM

Texture descriptor	Classifier	Classification rate

LBP + WT	SVM	Sensitivity=83.9%
		Specificity=81.35%
		Accuracy=82.9%

LBP + GLCM	SVM	Sensitivity=86.6%
		Specificity=84.35%
		Accuracy=85.7%

LBP + TSH	SVM	Sensitivity=82.6%
		Specificity=82.4%
		Accuracy=84.5%

WT + GLCM	SVM	Sensitivity=84.6%
		Specificity=82.35%
		Accuracy=83.7%

LBP + GLCM + WT	SVM	**Sensitivity=93.4%** **Specificity=94.25%** **Accuracy=93.74%**

Table 13 combined Texture Descriptors output using SVM as a classifier

4.1.4 Test 4

- Texture descriptors – Texture spectrum histogram , Local binary pattern, Gray level co-occurrence matrix and wavelet transform
- Classifier – Linear discriminant analysis

The classification performance of the texture descriptors is studied using 5000 polyp and 5000 non-polyp images for training and test purpose.

In Test four the classification performance of combined texture descriptors using linear discriminant analysis is discussed. Table 14 below illustrates the confusion matrix for each algorithm using. The TP, TN, FP, FN, N and P values can be obtained from the table to calculate the accuracy, sensitivity, and specificity of each algorithm as shown in table 15.

	polyp	non-polyp
polyp	2483	517
non-polyp	358	1642

(a) LBP + WT

	polyp	non-polyp
polyp	2558	442
non-polyp	333	1702

(b) LBP + GLCM

	polyp	non-polyp
polyp	2448	552
non-polyp	393	1767

(c) LBP + TSH

	polyp	non-polyp
polyp	2513	487
non-polyp	333	1667

(d) WT + GLCM

	polyp	non-polyp
polyp	2752	248
non-polyp	165	1835

(e) LBP + GLCM + WT

Table 14 the confusion matrix for each combined method using LDA

Texture descriptor	Classifier	Classification

LBP + WT	LDA	Sensitivity=82.76%
		Specificity=82.1%
		Accuracy=82.5%

LBP + GLCM	LDA	Sensitivity=85.26%
		Specificity=83.6%
		Accuracy=85.2%

LBP + TSH	LDA	Sensitivity=81.6%
		Specificity=81.95%
		Accuracy=84.3%

WT + GLCM	LDA	Sensitivity=83.76%
		Specificity=83.35%
		Accuracy=83.6%

LBP + GLCM + WT	LDA	**Sensitivity=91.73%** **Specificity=91.75%** **Accuracy=91.74%**

Table 15 combined Texture descriptors output using LDA as a classifier

4.2 Objective Comparison

The above four tests show as the classification rate of texture descriptors separately and in combination.

- In Test 1 the output of four texture descriptors using a support vector machine as a classifier is discussed, in this case, the gray level co=occurrence matrix gives good classification rate. As illustrated in fig 18 which is sensitivity 0.821, specificity 0.7835 and accuracy of 81.3% this method classifies the polyp and non-polyp images. The local binary pattern also gives good classification rate which is a good result when compared to Texture spectrum Histogram and Wavelet transform.

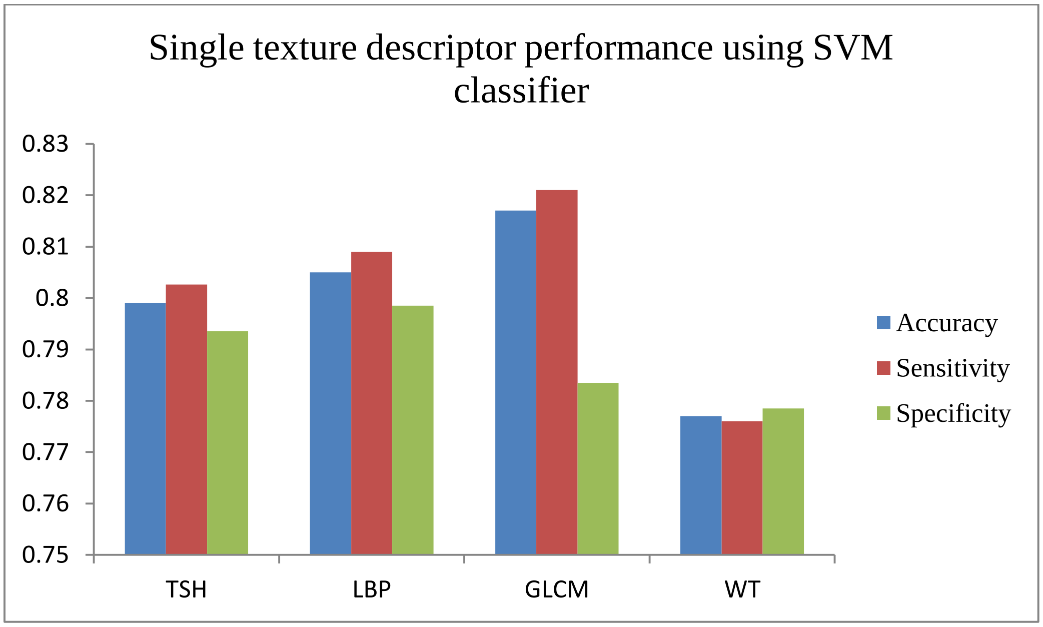


Figure 18 Performance comparison chart for single descriptors using SVM

- In Test 2 the output of four texture descriptors using Linear discriminant analysis as a classifier is discussed, in this case, the gray level co=occurrence matrix also gives good classification rate. As illustrated in fig 19 which is sensitivity 0.832, specificity 0.7835 and accuracy of 81.3% this method classifies the polyp and non-polyp images. The local binary pattern also gives good classification rate which is a good result when compared to Texture spectrum Histogram and Wavelet transform.

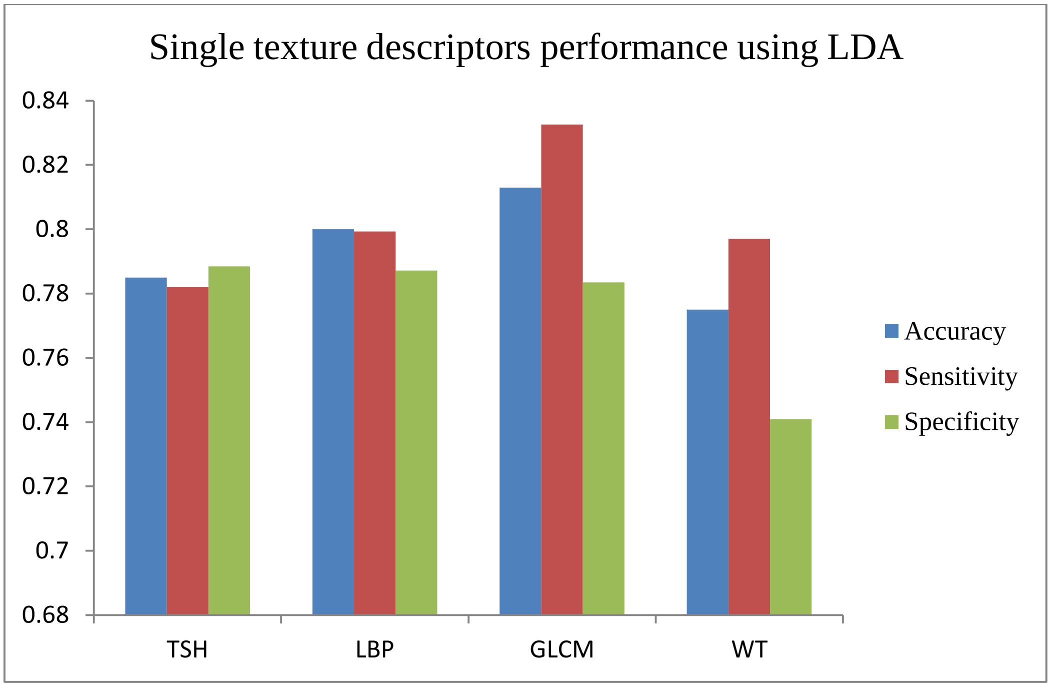


Figure 19 Performance comparison chart for single descriptors using LDA

- In Test 3 the classification rate of five combinations of texture descriptors using a support vector machine as a classifier is discussed. In this case, the combination of texture descriptors shows classification performance improvement. For instance, as illustrated in fig 20 the combination of Local binary pattern, Wavelet transform, and Gray level co-occurrence matrix gives sensitivity to 0.934, specificity 0.9425 and accuracy 93.74% which is an outstanding result. The combination of Local binary pattern and Gray level co-occurrence matrix also gives a good result.

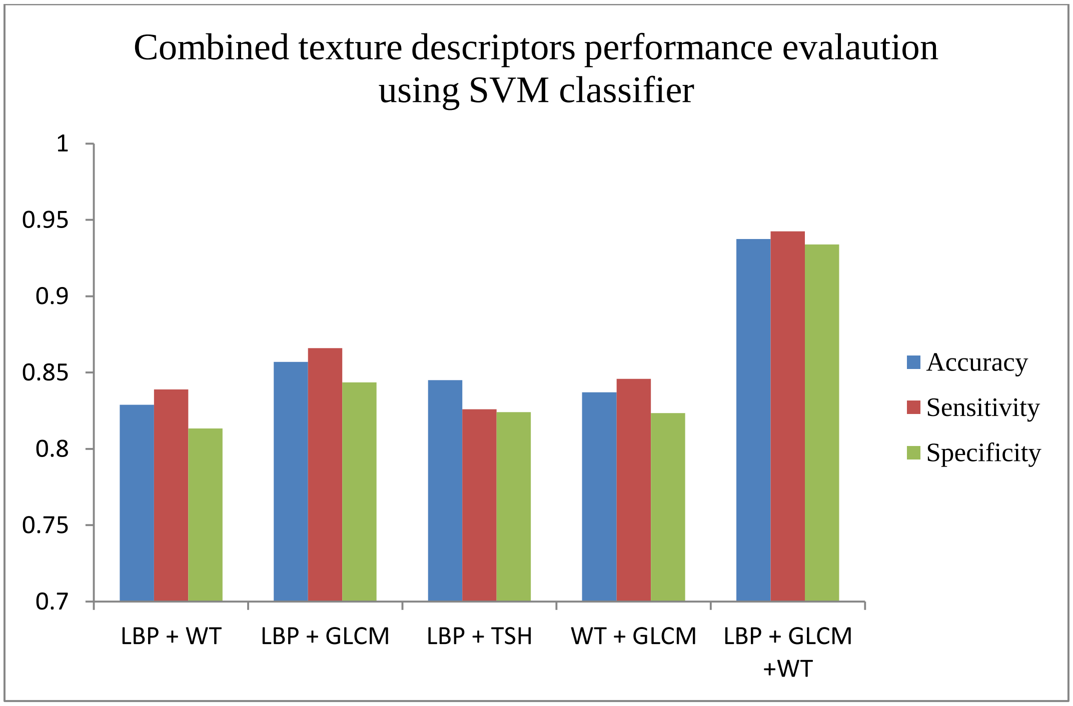


Figure 20 Performance comparison chart for combined   descriptors using SVM

- In Test 4 the classification rate of five combinations of texture descriptors using linear discriminant analysis as a classifier is discussed. In this case, also the combination texture descriptors show classification performance improvement. For instance, as illustrated in fig 20 the combination of Local binary pattern, Wavelet transform, and Gray level co-occurrence matrix gives sensitivity 0.9173, specificity 0.9175 and accuracy 91.74% which is an outstanding result. The combination of Local binary pattern and Gray level co-occurrence matrix also gives a good result.

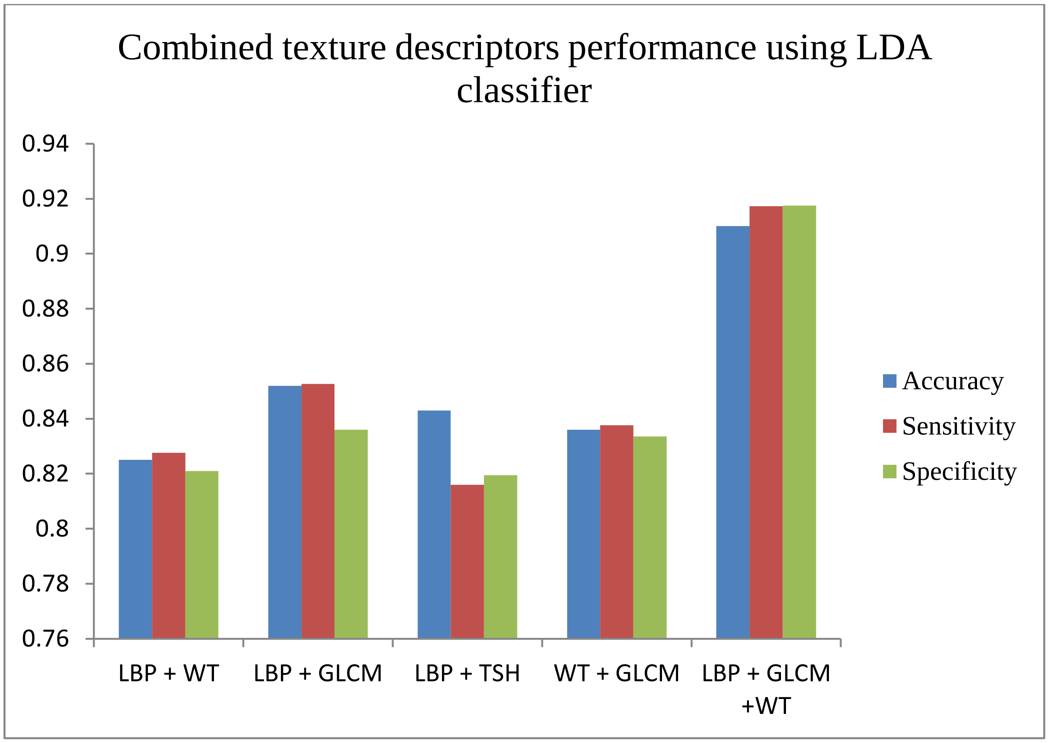


Figure 21 Performance comparison chart for single descriptors using LDA

Chapter five

Conclusion and Future Work

5.1 Conclusion

A polyp is an abnormal growth of tissue. Most of the colon cancer cases start with a polyp. Physicians remove the polyps even if it is non-cancerous to prevent its development into cancer. Colonoscopy is a device used to screen the colon. The colonoscopy is a tiny camera with a tube that can capture the color picture of the wall of the colon. The captured images are then diagnosed on the workstation. Developing automatic disease detection system has great importance because it reduces the miss by human subject due to lack of concentration and experience.

Texture has been widely used as the primary cue for the identification of polyps in colonoscopy images. The motivation behind using texture information for polyp detection is that the polyp has different color, shape, size, and appearance. This variation makes the usage of geometry based feature descriptors inefficient for polyp detection. The texture content of the frames with polyp and frames without polyp vary. The presence of polyp in a frame changes its texture content. This variation has been used to discriminate frames with polyp from normal frames in Colonoscopy images.

Different texture descriptors capture a different facet of the image texture. In this thesis Texture spectrum histogram, Gray level co-occurrence matrix, Local binary pattern and wavelet transform are used. These algorithms are applied on large polyp database introduced by Mayo clinic as part of ISBI Automatic Polyp Detection Challenge 2015. The dataset contains ground truth image masks for polyp regions. The dataset is freely available. This dataset can be used to compare methods without difficulty.

In the literature review survey, the authors have shown that combining different texture descriptors improve the performance of the classification.

A. Barley et al. [8] compared the performance of the co-occurrence matrix, Gabor wavelets steerable pyramids and SIFT on Brodatz, UIUCTex, and KTH-TIPS publically available Natural image texture datasets. They have evaluated these texture descriptors both in isolation and by combining several of them. They have reported that Gabor wavelets based texture descriptors give the highest recognition performance on Brodatz and KTH-TIPS datasets. It was also reported SIFT gives the highest classification accuracy for UIUC dataset. Different classification performance was reported when various descriptors are combined. All the methods were applied on natural image data set.

No work has been reported which combines several texture descriptors to improve the accuracy of automatic polyp detection systems even though certain improvement was reported for other datasets Natural image. No study has been reported on the performance comparison of state-of-the-art texture descriptors on publicly available polyp dataset. The comparative study of the texture descriptors can give us a clue about the essential texture features (edges around polyp boundary or the structure of the interior polyp region) that distinguishes the polyp region from the non-polyp region. In this work, we studied the performance comparison of the texture descriptors for polyp classification on publicly available polyp image dataset. The performance of the texture descriptors is thoroughly studied in isolation and by combining multiple of them. We determine the single optimal descriptor and optimal combination of the texture descriptors which can achieve high classification rate using support vector machine and linear discriminant analysis.

Therefore, Based on tests performed in the above, from the texture descriptors Gray level co-occurrence matrix gives the good classification accuracy while using SVM and LDA. From the combined texture descriptors the combination of texture descriptors shows classification accuracy improvement.

By combining several texture descriptors performance certain improvement has been reported when compared to classification using single texture descriptor. Generally, it is challenging to choose the best combination of texture descriptors for a range of datasets. This is because of the different texture aspects which are possessed by different datasets. Hence, the combination of texture descriptors which are suitable for a particular dataset has to be experimentally determined. As far as classification of polyp is concerned, polyp has different texture facet as compared to other texture datasets such as asphalt, concrete, and grass. Therefore, the texture descriptors which are combined gives improved classification accuracy of 93.74%.

5.2 Future Work

The aim of this thesis was to compare the effectiveness of many texture descriptors, to evaluate their efficiency in classification of polyp and evaluate the multiple combinations of texture descriptors in order to improve colonic polyp detection and select best classifier and methods**.**

The studies performed in natural image dataset gives highest classification accuracy by combining multiple texture descriptors in the case of polyp classification also the same result is obtained. And this thesis proposes best texture descriptor that gives highest classification accuracy. The future work includes the performance evaluation of texture based polyp image segmentation methods. In addition to their performance their computational time will be considered and studied in the future.

References

[1] Kodogiannis, V.S., Boulougoura, M. An Adaptive Neurofuzzy Approach for the Diagnosis in Wireless Capsule Endoscopy Imaging. International Journal of Information Technology .2007; 13(1): 46-56

[2] Mitchell, S., Cappell. The pathophysiology, clinical presentation, and diagnosis of colon cancer and adenomatous polyps. Medical clinics of north America.2005; 89:1-42

[3] J. G. Williams et al., “Management of the malignant colorectal polyp: ACPGBI position statement,” Color. Dis., vol. 15, pp. 1–38, Aug. 2013.

[4] Sembulingam,p.,sembuligam,k.Essentials of Medical Physiology.jaypee brothers medical publishers.2012;6:219

[5] Tate,s.s, 24 Digestive System.compressed | Human Digestive System | Gastrointestinal Tract. The Mc Graw-Hill Companies.2004;6.

[6] T.P.Matha and M.K.Shankar.feature extraction for the analysis of colon status from the endoscopic images.US National library of medicine, National institute of health.2:9.2003.

[7] I. r .koprowski. Department of Biomedical computer systems, university of silesia,faculity of computer science,UL, Bedzinka 39 ,Sosnowiez .2014.

[8] A. Barley and C. Town, “Combinations of Feature Descriptors for Texture Image Classification,” *J. Data Anal. Inf. Process.*, vol. 02, no. 03, pp. 67–76, Aug. 2014.

[9] B. Li and M. Q.-H. Meng, “Automatic polyp detection for wireless capsule endoscopy images,” Expert Syst. Appl., vol. 39, no. 12, pp. 10952–10958, Sep. 2012.

[10] S. A. Karkanis, D. K. Iakovidis, D. E. Maroulis, D. A. Karras, and M. Tzivras, “Computer-aided tumor detection in endoscopic video using color wavelet features,” IEEE Trans. Inf. Technol. Biomed., vol. 7, no. 3, pp. 141–152, Sep. 2003.

[11] J. Silva, A. Histace, O. Romain, X. Dray, and B. Granado, “Toward embedded detection of polyps in WCE images for early diagnosis of colorectal cancer,” Int. J. Comput. Assist.

Radiol. Surg., vol. 9, no. 2, pp. 283–293, Mar. 2014.

[12] D. K. Iakovidis and A. Koulaouzidis, “Automatic lesion detection in wireless capsule endoscopy &amp;#x2014; A simple solution for a complex problem,” in 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 2236–2240.

[13] D. J. C. Barbosa, J. Ramos, J. H. Correia, and C. S. Lima, “Automatic detection of small bowel tumors in capsule endoscopy based on color curvelet covariance statistical texture descriptors,” in 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009, pp. 6683–6686.

[14] B.-P. Li and M. Q.-H. Meng, “Comparison of Several Texture Features for Tumor Detection in CE Images,” J. Med. Syst., vol. 36, no. 4, pp. 2463–2469, Aug. 2012

[15] J. Bernal, J. Sanchez, and F. Vilarino, “Impact of image preprocessing methods on polyp localization in colonoscopy frames,” in 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2013, pp. 7350–7354.

[16] E. J. Beattie, N. D. Bloom, and J. C. Harvey, *Thoracic surgical oncology*. Churchill Livingstone, 1992.

[17] A. Deverakonda, “Research &amp; reviews : journal of medical and health sciences.,” Res. Rev. J. Med. Heal. Sci., vol. 5, no. 5, p. , Aug. 2012.

[18] Tuceryan, M.,Jain,A.K. Texture Analysis.The Handbook of pattern recognition and computer vision .1998;2:207-248

[19] Chen He, D.,Wang,L. Simplified Texture Spectrum for Texture Analysis. Journal of Communication and Computer. 2010; 7(8):45-53

[20] Karkanis, S., Galouri, K., Maraoulis,D. Classification of Endoscopic Images Based on Texture Spectrum. University of Athens, Dept. of Informatics, TYPA Bldg. Panepistimiopolis, 15784 Athens, Greece

[21] Chen He, D.,Wang,L.Textural filters baded on the textural spectrum.pattern recognition society.1991;24(12):1187-1195.

[22] Maenpaa,T.,Pietikainen, H. Texture analysis with local binary Patterns. Department of Electrical and Information Engineering. 2004

[23] Eichkitz.G.C., Davies, J., Amtmann, J., Schreilechner, G.M., Groot, D.P. Grey level co occurrence matrix and its application to seismic data. 2015; 33.

[24] Huang ,X., Liu,X., Zhang, L. A Multichannel Gray Level Co-Occurrence Matrix for Multi/Hyper spectral Image Texture Representation. Remote sensing.2014;6:8424-8445.

[25] Liv ens,S. Sc heunders, P., G. Wouwer,V.D. D. Dyck, V. Wavelets for Texture Analysis.VisieLab, Department of Physics, University of Antwerp.

[26] Smola, A., Vishwanathan, S.V.N. Introduction to Machine Learning. Departments of Statistics and Computer Science Purdue University and College of Engineering and Computer Science Australian National University.2010.

[27] Hur, B.A, Weston.J. A User's Guide to Support Vector Machines. Department of Computer Science Colorado State University.

[28] Gunn,S.,R. Support Vector Machines for Classification and Regression. Faculty of Engineering science and Mathematics school of Electronics &computer science.

[29] M. Alizadeh, O. H. Maghsoudi, K. Sharzehi, H. Reza Hemati, A. Kamali Asl, and A. Talebpour, "Detection of small bowel tumor in wireless capsule endoscopy images using an adaptive neuro-fuzzy inference system.," J. Biomed. Res., vol. 31, no. 5, pp. 419–427, Sep. 2017.

[30] S.Balakrishama,A.Ganapathiraju.LDA theory .institute for signal and information processing, Department of electrical and computer engineering.

[31] Mayo Clinic, www.polyp2015.com, available since February 2015.

[32] https://www.medicalnewstoday.com/articles/155598.php

Appendix

A.1 Implementation code for LBP using SVM

```
clear all;
clc;
close all;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/'
num2str(i)],'.tiff'));
I = rgb2gray(a);
K=imresize(double(I),[ 540 540]);
features = extractLBPFeatures(K);
data=[data;features];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
SVMModel = fitcsvm(X,P,'Standardize',true,'KernelFunction','RBF',...
 'KernelScale','auto');
% test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
K=imresize(double(d),[ 540 540]);
features2= extractLBPFeatures(K);
data2=[data2;features2];
end
xtest=data2;
figure
Label = predict(SVMModel,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.2 Implementation code for GLCM using SVM

```
clear all
close all
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/'
num2str(i)],'.tiff'));
b=rgb2gray(a);
glcm = graycomatrix(b,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data=[data;feat];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
```

```
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
% SVMModel = fitclinear(X,P);
 SVMModel = fitcsvm(X,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
figure;
%   % test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
glcm = graycomatrix(d,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);

feat2=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data2=[data2;feat2];
end
xtest=data2;
Label = predict(SVMModel,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.3 Implementation code for TSH using SVM

```
clear all
close all
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/'
num2str(i)],'.tiff'));
b=rgb2gray(a);
texturespec = calcTextureSpectrum(b);
data=[data;texturespec];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
SVMModel = fitcsvm(X,P,'Standardize',true,'KernelFunction','RBF',...
     'KernelScale','auto');
figure;
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
% test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
texturespec = calcTextureSpectrum(d);
data2=[data2;texturespec];
end
xtest=data2;
Label = predict(SVMModel,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.4 Implementation code for WT using SVM

```
close all;
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/'
num2str(i)],'.tiff'));
b=rgb2gray(a);
N=1;
[C,S] = wavedec2(b,N,'haar');
B = reshape(S,1,6);
data=[data;B];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
SVMModel = fitcsvm(X,P,'Standardize',true,'KernelFunction','RBF',...
 'KernelScale','auto');
% SVMStruct = fitcsvm(X,P);
figure;
%   % test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
N=1;
[C,S] = wavedec2(d,N,'haar');
M= reshape(S,1,6);
data2=[data2;M];
end
xtest=data2;
Label = predict(SVMModel,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.5 Implementation code for combining two texture descriptors (function)

```
 function data = featuresfrom(algorithm,TrainingDataPath,imagefiles)
addpath(TrainingDataPath)

nfiles = length(imagefiles);    % Number of files found
data=[];
switch algorithm
    case 'GLCM'
        for ii=1:nfiles
            currentfilename = imagefiles(ii).name;
            currentimage = imread(currentfilename);
            b=rgb2gray(currentimage);
            glcm = graycomatrix(b,'Offset',[2 0],'NumLevels',16,'symmetric',true);
            stats = graycoprops(glcm);
            feat=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
            data=[data;feat];
        end
    case 'TS'
         for ii=1:nfiles
            currentfilename = imagefiles(ii).name;
            currentimage = imread(currentfilename);
            b=rgb2gray(currentimage);
            texturespec = calcTextureSpectrum(b);
            data=[data;texturespec];

         end
    case 'LBP'
        for ii=1:nfiles
            currentfilename = imagefiles(ii).name;
            currentimage = imread(currentfilename);
            b=rgb2gray(currentimage);
            K=imresize(double(b),[ 540 540]);
            features = extractLBPFeatures(K);
            data=[data;features];
          end

    case 'WT'
         for ii=1:nfiles
            currentfilename = imagefiles(ii).name;
            currentimage = imread(currentfilename);
            b=rgb2gray(currentimage);
            N=1;
            [C,S] = wavedec2(b,N,'haar');
            B = reshape(S,1,6);
            data=[data;B];
          end


end
end
```

A.6 Implementation code for LBP + WT using SVM

```
clc
clear
close all

%TrainingDataPath = uigetdir('','Load Training Dataset');
%Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);
%TestDataPath = uigetdir('','Load Test Dataset');
%Testimagefiles = dir([TestDataPath,'/*','.tiff']);
%GroundTruthPath = uigetdir('','Load CSV File');
%GroundTruthName = dir([GroundTruthPath,'/*','.csv']);
%featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);
%featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);

data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
I = rgb2gray(a);
K=imresize(double(I),[ 540 540]);
features = extractLBPFeatures(K);
data=[data;features];
end
featurefromLBPTraining=data;
data1=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
N=1;
[C,S] = wavedec2(b,N,'haar');
B = reshape(S,1,6);
data1=[data1;B];
end
featurefromWTTraining=data1;

XTraining=[featurefromWTTraining featurefromLBPTraining];
%G=csvread([GroundTruthPath,'/' GroundTruthName.name]);
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
SVMModel = fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
 'KernelScale','auto');

data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
K=imresize(double(d),[ 540 540]);
features2= extractLBPFeatures(K);
data2=[data2;features2];
end
featurefromLBPTest=data2;

data3=[];
for i=1:5000
```

```
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
N=1;
[C,S] = wavedec2(d,N,'haar');
M= reshape(S,1,6);
data3=[data3;M];
end
featurefromWTTest=data3;
%featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);
%featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);

XTest=[featurefromWTTest featurefromLBPTest];
Label = predict(SVMModel,XTest);
```

A.7 Implementation code for LBP + GLCM using SVM as a classifier

```
clc
clear
close all

%TrainingDataPath = uigetdir('','Load Training Dataset');
%Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);
%TestDataPath = uigetdir('','Load Test Dataset');
%Testimagefiles = dir([TestDataPath,'/*','.tiff']);
%GroundTruthPath = uigetdir('','Load CSV File');
%GroundTruthName = dir([GroundTruthPath,'/*','.csv']);
%featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);
%featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);

data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
I = rgb2gray(a);
K=imresize(double(I),[ 540 540]);
features = extractLBPFeatures(K);
data=[data;features];
end
featurefromLBPTraining=data;
data1=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
glcm = graycomatrix(b,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data1=[data1;feat];
end
featurefromGLCMTraining=data1;

XTraining=[featurefromGLCMTraining featurefromLBPTraining];

P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
```

```
  SVMModel = fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
K=imresize(double(d),[ 540 540]);
features2= extractLBPFeatures(K);
data2=[data2;features2];
end
featurefromLBPTest=data2;
data3=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
glcm = graycomatrix(d,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat2=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data3=[data3;feat2];
end
featurefromGLCMTest=data3;
%XTest=horzcat(featurefromGLCMTest,featurefromLBPTest);
XTest=[featurefromGLCMTest featurefromLBPTest];
Label = predict( SVMModel,XTest);
```

A.8 Implementation code for LBP + TSH using SVM

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);

featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);

featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/'num2str(i)],'.tiff'));
I = rgb2gray(a);
K=imresize(double(I),[ 540 540]);
features = extractLBPFeatures(K);
data=[data;features];
end
featurefromLBPTraining=data;
data=[];
for i=1:5000
```

```
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a)
texturespec = calcTextureSpectrum(b);
data=[data;texturespec];
end
featurefromTSTraining=data;

XTraining=[featurefromTSTraining featurefromLBPTraining];
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
SVMModel = fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
K=imresize(double(d),[ 540 540]);
features2= extractLBPFeatures(K);
data2=[data2;features2];
end
featurefromLBPTest=data2;
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/'num2str(i)],'.tiff'));
d=rgb2gray(c);
texturespec = calcTextureSpectrum(d);
data2=[data2;texturespec];
end
featurefromTSTest=data2;
 XTest=[featurefromTSTest featurefromLBPTest];
Label = predict(SVMModel,XTest)
```

A.9 Implementation code for WT+ GLCM + LBP using SVM

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);
featurefromGLCMTraining = featuresfrom('GLCM',TrainingDataPath,Trainingimagefiles);
featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromGLCMTraining featurefromLBPTraining featurefromWTTraining];

P=csvread([GroundTruthPath,'/' GroundTruthName.name]);

SVMModel = fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');

featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);
featurefromGLCMTest = featuresfrom('GLCM',TestDataPath,Testimagefiles);
featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);

XTest=[featurefromLBPTest featurefromGLCMTest featurefromWTTest];
label = predict(SVMModel,XTest);
```

A.10 Implementation code for GLCM + TSH using SVM

```
clc
clear
close all
% TrainingDataPath = uigetdir('','Load Training Dataset');
% Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);
%
% TestDataPath = uigetdir('','Load Test Dataset');
% Testimagefiles = dir([TestDataPath,'/*','.tiff']);
%
% GroundTruthPath = uigetdir('','Load CSV File');
% GroundTruthName = dir([GroundTruthPath,'/*','.csv']);
% featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);
% featurefromGLCMTraining = featuresfrom('GLCM',TrainingDataPath,Trainingimagefiles);
%
% XTraining=[featurefromGLCMTraining featurefromTSHTraining];
%
% G=csvread([GroundTruthPath,'/' GroundTruthName.name]);
% SVMModel =fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
%
% featurefromTSHTest = featuresfrom('TS',TestDataPath,Testimagefiles);
% featurefromGLCMTest = featuresfrom('GLCM',TestDataPath,Testimagefiles);
%
% XTest=[featurefromGLCMTest featurefromTSHTest];
% Label = predict(SVMModel,XTest);

data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
glcm = graycomatrix(b,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data=[data;feat];
end
featurefromGLCMTraining=data;

data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
texturespec = calcTextureSpectrum(b);
data=[data;texturespec];
end
featurefromTSTraining=data;

XTraining=[featurefromTSTraining featurefromGLCMTraining];
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
Z= csvread('E:/Important/GHM/GOD/methods/GroundTruth/K.csv');
SVMModel = fitcsvm(XTraining,P,'Standardize',true,'KernelFunction','RBF',...
    'KernelScale','auto');
data2=[];
for i=1:5000
```

```
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
glcm = graycomatrix(d,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat2=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data2=[data2;feat2];
end
featurefromGLCMTest=data2;
data2=[];
for i=5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
texturespec = calcTextureSpectrum(d);
data2=[data2;texturespec];
end
featurefromTSTest=data2;

% featurefromGLCMTest = featuresfrom('TS',TestDataPath,Testimagefiles);
% featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);

XTest=[featurefromTSTest featurefromGLCMTest];
Label = predict(SVMModel,XTest);
```

A.13 Implementation code for LBP using LDA

```
clear all;
clc;
close all;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
I = rgb2gray(a);
K=imresize(double(I),[ 540 540]);
features = extractLBPFeatures(K);
data=[data;features];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
MdlLinear = fitcdiscr(X,P);
% test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
K=imresize(double(d),[ 540 540]);
features2= extractLBPFeatures(K);
data2=[data2;features2];
end
xtest=data2;
figure
label = predict(MdlLinear,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.14 Implementation code for TSH using LDA

```
clear all
close all
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
texturespec = calcTextureSpectrum(b);
data=[data;texturespec];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
MdlLinear = fitcdiscr(X,P);
figure;
% test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
texturespec = calcTextureSpectrum(d);
data2=[data2;texturespec];
end
xtest=data2;
label = predict(MdlLinear,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.15 Implementation code for GLCM using LDA

```
clear all
close all
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
glcm = graycomatrix(b,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
feat=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data=[data;feat];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
MdlLinear = fitcdiscr(X,P);
figure;
%   % test data
data2=[];
for i=1:5000
c=imread(strcat(('E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
glcm = graycomatrix(d,'Offset',[2 0],'NumLevels',16,'symmetric',true);
stats = graycoprops(glcm);
```

```
feat2=[stats.Contrast stats.Correlation stats.Energy stats.Homogeneity];
data2=[data2;feat2];
end
```

A.16 Implementation code for WT using LDA

```
close all;
clc;
data=[];
for i=1:5000
a=imread(strcat(['E:/Important/GHM/GOD/methods/TTrain/' num2str(i)],'.tiff'));
b=rgb2gray(a);
N=1;
[C,S] = wavedec2(b,N,'haar');
B = reshape(S,1,6);
data=[data;B];
end
X=data;
P=csvread('E:/Important/GHM/GOD/methods/GroundTruth/W.csv');
MdlLinear = fitcdiscr(X,P);
figure;
%   % test data
data2=[];
for i=1:5000
c=imread(strcat(['E:/Important/GHM/GOD/methods/TTest/' num2str(i)],'.tiff'));
d=rgb2gray(c);
N=1;
[C,S] = wavedec2(d,N,'haar');
M= reshape(S,1,6);
data2=[data2;M];
end
xtest=data2;
label = predict(MdlLinear,xtest);
hold on;
plot(X(:,1),X(:,2),'ro','MarkerSize',12);
hold off
```

A.17 Implementation code for LBP + WT using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);
featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromGLCMTraining featurefromLBPTraining];

G=csvread([GroundTruthPath,'/' GroundTruthName.name]);


MdlLinear = fitcdiscr(XTraining,G);

featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);
featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);

XTest=[featurefromWTTest featurefromLBPTest];
label = predict(MdlLinear,XTest);
```

A.18 Implementation code for LBP + GLCM using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromGLCMTraining = featuresfrom('GLCM',TrainingDataPath,Trainingimagefiles);
featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromGLCMTraining featurefromLBPTraining];

G=csvread([GroundTruthPath,'/' GroundTruthName.name]);
```

```
MdlLinear = fitcdiscr(XTraining,G);

featurefromGLCMTest = featuresfrom('GLCM',TestDataPath,Testimagefiles);
featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);

XTest=[featurefromGLCMTest featurefromLBPTest];
label = predict(MdlLinear,XTest);
```

A.19 Implementation code for LBP + TSH using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);
featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromTSHTraining featurefromLBPTraining];

Y=csvread([GroundTruthPath,'/' GroundTruthName.name]);

MdlLinear = fitcdiscr(XTraining,Y);

featurefromTSHTest = featuresfrom('TS',TestDataPath,Testimagefiles);
featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);

XTest=[featurefromTSHTest featurefromLBPTest];
label = predict(MdlLinear,XTest);
```

A.20 Implementation code for WT + TSH using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);
featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromTSHTraining featurefromWTTraining];

W=csvread([GroundTruthPath,'/' GroundTruthName.name]);

MdlLinear = fitcdiscr(XTraining,W);

featurefromTSHTest = featuresfrom('TS',TestDataPath,Testimagefiles);
featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);

XTest=[featurefromTSHTest featurefromWTTest];
label = predict(MdlLinear,XTest);
```

A.21 Implementation code for GLCM+ TSH using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);
featurefromGLCMTraining = featuresfrom('GLCM',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromGLCMTraining featurefromTSHTraining];

W=csvread([GroundTruthPath,'/' GroundTruthName.name]);

MdlLinear = fitcdiscr(XTraining,W);
```

```
featurefromTSHTest = featuresfrom('TS',TestDataPath,Testimagefiles);
featurefromGLCMTest = featuresfrom('GLCM',TestDataPath,Testimagefiles);

XTest=[featurefromGLCMTest featurefromTSHTest];
label = predict(MdlLinear,XTest);
```

A.21 Implementation code for LBP + GLCM+ WT using LDA

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);
featurefromGLCMTraining = featuresfrom('GLCM',TrainingDataPath,Trainingimagefiles);
featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromGLCMTraining featurefromLBPTraining featurefromWTTraining];

G=csvread([GroundTruthPath,'/' GroundTruthName.name]);

MdlLinear = fitcdiscr(XTraining,G);

featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);
featurefromGLCMTest = featuresfrom('GLCM',TestDataPath,Testimagefiles);
featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);

XTest=[featurefromLBPTest featurefromGLCMTest featurefromWTTest];
label = predict(MdlLinear,XTest);
```

A.22 Implementation code for LBP + WT + TSH

```
clc
clear
close all

TrainingDataPath = uigetdir('','Load Training Dataset');
Trainingimagefiles = dir([TrainingDataPath,'/*','.tiff']);

TestDataPath = uigetdir('','Load Test Dataset');
Testimagefiles = dir([TestDataPath,'/*','.tiff']);

GroundTruthPath = uigetdir('','Load CSV File');
GroundTruthName = dir([GroundTruthPath,'/*','.csv']);


featurefromLBPTraining = featuresfrom('LBP',TrainingDataPath,Trainingimagefiles);
featurefromWTTraining = featuresfrom('WT',TrainingDataPath,Trainingimagefiles);
featurefromTSHTraining = featuresfrom('TS',TrainingDataPath,Trainingimagefiles);

XTraining=[featurefromLBPTraining featurefromWTTraining featurefromTSTraining];

G=csvread([GroundTruthPath,'/' GroundTruthName.name]);


MdlLinear = fitcdiscr(XTraining,G);

featurefromLBPTest = featuresfrom('LBP',TestDataPath,Testimagefiles);
featurefromWTTest = featuresfrom('WT',TestDataPath,Testimagefiles);
featurefromTSTest = featuresfrom('TS',TestDataPath,Testimagefiles);

XTest=[featurefromLBPTest featurefromWTTest featurefromTSTest];
label = predict(MdlLinear,XTest);
```