DETECTION OF ULCERATIVE COLITIS SEVERITY AND ENHANCEMENT OF INFORMATIVE FRAME FILTERING USING TEXTURE ANALYSIS IN COLONOSCOPY VIDEOS

Ashok Dahal

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

December 2015

APPROVED:

JungHwan Oh, Major Professor
Bill Buckles, Committee Member
Yan Huang, Committee Member
Song Fu, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Mark Wardell, Dean of the Toulouse Graduate School

Dahal, Ashok. *Detection of Ulcerative Colitis Severity and Enhancement of Informative Frame Filtering using Texture Analysis in Colonoscopy Videos*. Doctor of Philosophy (Computer Science), December 2015, 74 pp., 12 tables, 34 figures, 52 numbered references.

There are several types of disorders that affect our colon's ability to function properly such as colorectal cancer, ulcerative colitis, diverticulitis, irritable bowel syndrome and colonic polyps. Automatic detection of these diseases would inform the endoscopist of possible sub-optimal inspection during the colonoscopy procedure as well as save time during post-procedure evaluation. But existing systems only detects few of those disorders like colonic polyps. In this dissertation, we address the automatic detection of another important disorder called ulcerative colitis. We propose a novel texture feature extraction technique to detect the severity of ulcerative colitis in block, image, and video levels. We also enhance the current informative frame filtering methods by detecting water and bubble frames using our proposed technique. Our feature extraction algorithm based on accumulation of pixel value difference provides better accuracy at faster speed than the existing methods making it highly suitable for real-time systems. We also propose a hybrid approach in which our feature method is combined with existing feature method(s) to provide even better accuracy. We extend the block and image level detection method to video level severity score calculation and shot segmentation. Also, the proposed novel feature extraction method can detect water and bubble frames in colonoscopy videos with very high accuracy in significantly less processing time even when clustering is used to reduce the training size by 10 times.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Dr. JungHwan Oh for his constant support and encouragement throughout my doctoral degree. His excellent guidance and readiness to help was one of the key factors accomplishing this huge achievement. I also thank my committee members Dr. Bill Buckles, Dr. Yan Huang, and Dr. Song Fu for providing me valuable suggestions during my degree. This dissertation would not have been possible without the contributions of my advisor and PhD committee members. I also acknowledge the valued feedbacks of Dr. Wallapak Tavanapong (Iowa State University, Ames, IA) and Dr. Piet C. de Groen (Mayo Clinic, Rochester, MN) towards my research and publications. Also, I would like to thank my advisor and the Department of Computer Science and Engineering for providing me continuous financial support throughout my degree. I would also like to thank my previous lab mates Dr. Nawarathna and Dr. Muthukudage and current lab mate Ali Alammari for their help in various stages of my degree.

Finally, I am forever thankful to my family for their love, inspiration, and belief. This journey would not have been possible without their prayers and encouragements. I am especially grateful to my father Tunga Nath Dahal, mother Hamala Dahal, and sister Shusila for giving me the opportunity to peruse my PhD degree. I also like to thank my uncles and aunts, father in-law, mother in-law, grandparents as well as cousins for their constant love and support. Last but not the least, I would like to express my heartfelt thanks to my loving wife, Bigya Pokhrel Dahal for her endless support during difficult times. She was always there for me motivating and encouraging to reach the finish line and ultimately helped me achieve this accomplishment.

Table of Contents Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1: INTRODUCTION

Colonoscopy is the preferred screening modality for prevention of colorectal cancer---the second leading cause of cancer-related deaths in the US [1]. As the name implies, colorectal cancers are malignant tumors that develop in the colon and rectum. The survival rate is higher if the cancer is found and treated early before metastasis to lymph nodes or other organs occurs. To prevent death due to this disease, the current Medicare guidelines suggest that each US citizen undergo colonoscopy at least once every 10 years starting at age 50. Colonoscopy is a complex procedure. It consists of two phases: an *insertion phase* and a *withdrawal phase*. During the insertion phase, a flexible endoscope (a flexible tube with a tiny video camera at the tip) is advanced under direct vision via the anus into the rectum and then gradually into the cecum (the most proximal part of the colon) or the terminal ileum. During the withdrawal phase, the endoscope is gradually withdrawn. The camera generates a video signal of the interior of the human colon, which is displayed on a monitor for real-time analysis by the physician. The purpose of the insertion phase is to reach the cecum or the terminal ileum. Careful mucosa inspection and diagnostic or therapeutic interventions such as biopsy, polyp removal, etc., are performed during the withdrawal phase. The inspection should be thorough of the colon mucosa and reach the end of colon indicated by the appearance of the appendix, ileocecal valve, or the small bowel mucosa. Colonoscopy has contributed to a marked decline in the number of colorectal cancer related deaths. However, recent data suggest that there is a significant (4-12%) miss-rate for the detection of even large polyps and cancers [2]. The miss-rate may be related to the experience of the endoscopist and the location of the lesion in the colon, but no prospective studies related to this have

been done thus far. In 2006, American Society for Gastrointestinal Endoscopy (ASGE) and American College of Gastroenterology (ACG) issued guidelines for quality colonoscopy. The guidelines suggest that (1) on average the withdrawal phase during a screening colonoscopy should last a minimum of 6 minutes and (2) visualization of cecum anatomical landmarks such as the appendiceal orifice and the ileocecal valve should be documented [2]. Nevertheless, there was no measurement method to evaluate the endoscopist's skill and the quality of a colonoscopic procedure.

1.1 Motivation and Significance

To address this critical need, Dr. JungHwan Oh, his colleagues, Dr. Wallapak Tavanapong (Iowa State University, Ames, IA) and Dr. Piet C. de Groen (Mayo Clinic, Rochester, MN), and his previous and current students have been investigating automated procedure quality measurement system [3] by adapting some algorithms and software developed with the support of the NSF funded Endoscopic Multimedia Information System (EMIS) project, Mayo Clinic and university research grants. As a result, the research team has in place hardware and software for collecting annotated colonoscopy videos, and images that can immediately be used for education activities (presentations, teaching of fellows, manuscripts, etc.), real-time blurry frame detection which includes a method to evaluate images without any reference frame in real time, real time detection of maximum intubation, which includes methods to identify reliable motion vectors, camera motion shots, and the end of the insertion phase, real-time polyp detection, and providing feedback in real time, which can be used by physicians and quality control committees to evaluate and improve the quality of colonoscopy in their institutions.

1.2 Problems Addressed in Dissertation

There are several types of disorders that affect our colon's ability to function properly such as Colorectal Cancer, Ulcerative Colitis, Diverticulitis, Irritable Bowel Syndrome, Colonic polyps and other abnormalities. As discussed in the previous section, the automated procedure quality measurement system can provide Colonic polyp detection only at the moment among these disorders. We would like to add a function to handle one of the important disorders called 'Ulcerative Colitis'. We propose a novel method to detect the severity of Ulcerative Colitis. We have investigated its detection in both block level, image level as well as video level. Besides, we have investigated methods to detect water and bubble frames from colonoscopy videos. Existing non-informative frame detection methods fails to detect water and bubble frames as non-informative ones because of edge structures present in the images. Accurately detecting and discarding water and bubble frames can improve the performance of the automated feedback system.

1.2.1 Severity of Chronic Ulcerative Colitis in Colonoscopy Videos

We propose a novel method to detect the severity of Ulcerative Colitis. The severity includes five classes which are 'severe', 'moderate', 'mild', 'scar', and 'normal'. We introduce a novel feature extraction algorithm based on accumulation of pixel value differences, which provides better accuracy, and at faster speed than the existing methods. Since there is no one type of texture feature providing reasonable accuracies for all five classes, we propose a hybrid approach in which a new proposed feature based on the accumulation of pixel value differences is combined with an existing feature such

as LBP [4]. Hence, our contributions are: (a) to introduce a novel feature extraction algorithm which is more than two times faster than existing algorithms such as LBP, and (b) to propose a hybrid method for classification of multiple classes with significantly improved accuracy.

1.2.2 Water and Bubble Detection to Enhance the Informative Frame Filtering

A fundamental step of the automated feedback system is to distinguish non-informative frames from informative ones. Existing non-informative frame detection methods fails to detect water and bubble frames as non-informative ones because of edge structures present in the images. We propose a novel method for water and bubble frame detection based on image texture focusing on accumulation of pixel value differences. We compare it with other existing texture based algorithms in terms of accuracy and execution time. To further reduce the execution time, we investigate different clustering methods for our training datasets. The proposed method performs very well in terms of accuracy and execution speed with or without clustering at a faster execution time. Therefore, our main contribution is to propose a novel method which can detect water and bubble frames with very high accuracy in significantly less processing time even when clustering is used to reduce the training size by almost a factor of 10.

1.3 Organization of Dissertation

The reminder of the dissertation is organized as follows. Chapter 2 describes the ulcerative colitis severity detection in colonoscopy video frames. Chapter 3 discusses the enhancement of informative frame filtering by water and bubble detection in colonoscopy

videos. Finally, chapter 4 gives some discussion and concluding remarks as well as future direction of the research topics discussed in this dissertation.

CHAPTER 2: DETECTION OF ULCERATIVE COLITIS SEVERITY IN COLONOSCOPY VIDEO FRAMES[1]

2.1 Introduction

Ulcerative Colitis (UC) is a chronic inflammatory disease characterized by periods of relapses and remissions affecting more than 500,000 people in the United States [1]. The therapeutic goals of the UC are to first induce and then maintain disease remission. Endoscopic disease severity may better predict future outcomes in UC. However, currently there are no validated clinical scoring systems that have been consistently utilized in UC clinical trials. Randomized controlled UC trials have used one of nine different clinical scoring systems to determine therapeutic efficacy [5-7]. Almost all UC patients with deep, extensive ulcers underwent colectomy (93%) compared to only 23% with only superficial ulcers present during colonoscopy [8]. Among patients with newly diagnosed moderate to severe UC requiring an initial course of systemic corticosteroids, absence of mucosal healing at 3 months was an independent predictor of more intensive future medical therapy, hospitalizations and colectomies [9]. Hence, it is very significant to detect the severity of UC for better management of UC disease and reduce its overall impact.

However, it is very difficult to evaluate the severity of UC objectively because of non-uniform nature of symptoms associated with UC, and large variations in their patterns

[1] Parts of this chapter have been previously published, either in part or in full, from A. Dahal, J. Oh, W. Tavanapong, J. Wong, and P. C. de Groen (2015). Detection of Ulcerative Colitis Severity in Colonoscopy Video Frames in 13th International Workshop on Content-Based Multimedia Indexing (CBMI), 2015, pp. 1-6, DOI: 10.1109/CBMI.2015.7153617.

[10]. To address this, we objectively measure and classify the severity of UC presented in optical colonoscopy video frames based on the image textures. For the evaluation of the severity of UC, we use four UC classes such as ‘severe’, ‘moderate’, ‘mild’ and ‘scar’ [11], and one ‘normal’ class as seen in Figure 2.1. It is clear that any of these five classes does not have a dominant color, so color based approaches are not reliable. Also, these five classes do not possess any specific shape. Thus, the shape-based approaches are not suitable either. However, image textures could be an option for detecting the severity of UC since the UC images consist of various textures. We have experimented and evaluated various popular texture features such as Higher Order Local Auto Correlations (HLAC) [10, 12], Local Binary Pattern (LBP) [4, 13, 14], Gabor filter banks [15], Leung-Malik filter banks [16], a modified version of LBP [17], the

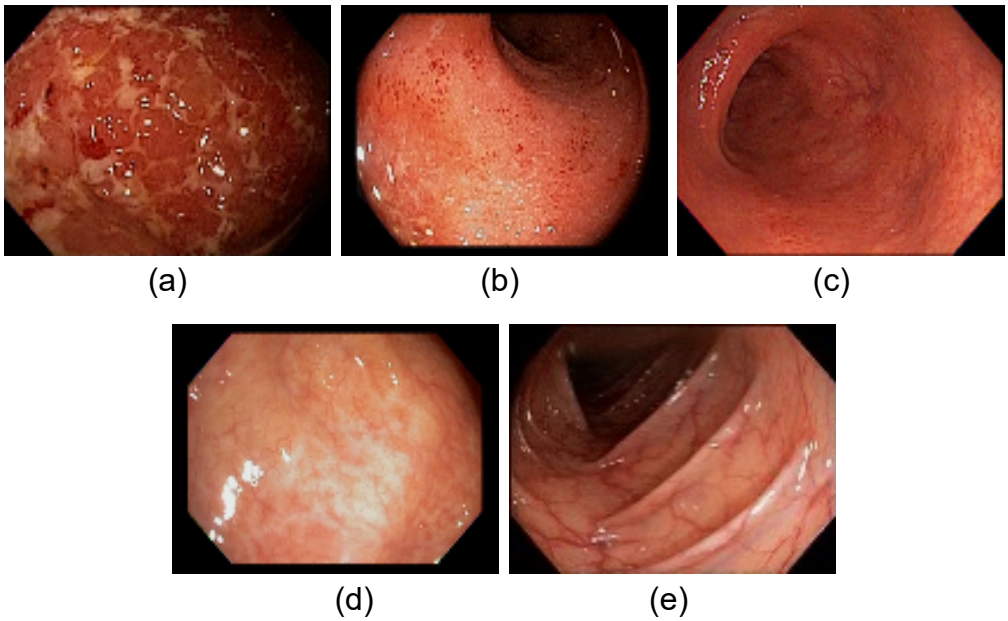


Figure 2.1 Images in different classes of UC. a) severe, b) moderate, c) mild, d) scar, and e) normal.

traditional texture features (i.e., Contrast, Correlation, Energy, Homogeneity, etc.) based on Gray-Level Co-Occurrence Matrix (GLCM) [18] as well as MPEG-7 texture features

[19]. Based on our experiments, none of the existing algorithms provides reasonable accuracy for all five classes of UC images within a training set we have created. We observed that, most features work well for 'mild' class, LBP works well for 'scar' and 'normal' classes, but any of existing methods did not provide reasonable accuracy for 'severe' and 'moderate' classes. More detailed explanation will be provided in the experimental section later. We introduce a novel feature extraction algorithm based on accumulation of pixel value differences, which provides better accuracy for 'severe' and 'moderate' classes, and at faster speed than the existing methods. Since there is no one type of texture feature providing reasonable accuracies for all five classes, we propose a hybrid approach in which a new proposed feature based on the accumulation of pixel value differences is combined with an existing feature such as LBP.

Hence, our contributions are: (a) to introduce a novel feature extraction algorithm which is more than two times faster than existing algorithms such as LBP, and (b) to propose a hybrid method for classification of multiple classes with significantly improved accuracy. The remainder of this chapter is organized as follows. Related work is presented in Section 2.2. The proposed technique is described in Section 2.3. In Section 2.4, we discuss our experimental setup and results. Finally, Section 2.5 presents some concluding remarks.

2.2 Related Work

Not much research has been done related to automated detection of UC disease features in colonoscopy videos. There are several literatures dealing with the texture detection and analysis in different types of images ranging from medical to non-medical images. There are literatures related to wireless capsule endoscopy (WCE) but WCE and

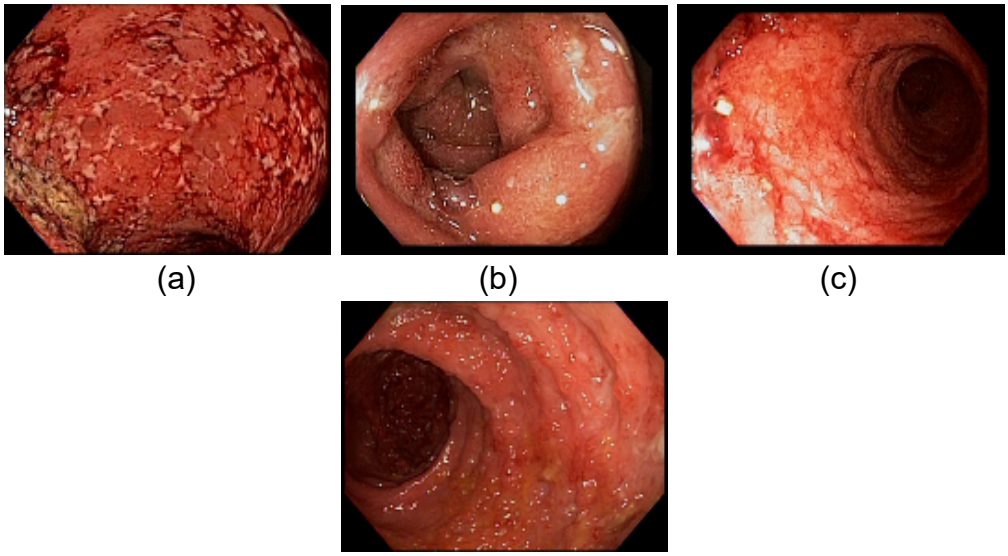
colonoscopy images are somehow different. In Wireless Capsule Endoscopy, research has be done to detect ulcers with the use of color as well as texture features. Li and Meng have proposed ulcer detection in WCE images by using chromaticity moments [20]. Their approach mainly focused on color features in HSI color space. Recently [21] has proposed another ulcer detection in WCE using bag-of-word model and feature fusion technique using LBP and SIFT features. Similarly, Li et al [22] have proposed ulcer detection in WCE by combining several classifiers as hybrid model. In terms of colonoscopy, authors in [10] classified the UC images by extracting Higher Order Local Auto Correlations from the saturation channel in HSV color space. This method considers the whole image for the feature extraction and classification, and very few images (total 27) were used for training and testing. We could not reproduce a similar accuracy as described in [10] using our larger dataset. The comparison with our method will be provided in the experimental section.

An automatic method of colitis detection in abdominal CT (Computerized Tomography) scans is proposed where the UC and non-UC images are detected by Gabor filter banks using k-means clustering and histograms from the codebook generated previously [23]. This method is close to [24] where texture based abnormality detection is applied to endoscopy video frames using Leung-Malik (LM) filter banks and local binary patterns (LBP). But in both of these works, images are classified into only two classes (i.e., normal and abnormal). Our problem involves multiple classes (5 classes) and these classes are very close to one another making it very difficult for accurate classification.

2.3 Blocks Extraction Methodology

The UC textures are not uniform throughout the image resulting in a significant number of variations in the textures. This makes the texture detection very challenging because we have to deal with several different variations of the same class. Figure 2.2 shows the images in the same severe class with very different textures. It can be seen that the severe texture vary from a video frame to another. Also, it can be seen that the textures are not present throughout the image. The pattern is similar in other classes as well. Therefore, it is better to extract the features based on blocks rather than the entire image. We divide the UC images (720x480 pixels) into a number of blocks in which each block is 128x128 pixels in size. We experimented with various block sizes such as 32x32, 64x64, 128x128, and 256x256 pixels and empirically determined the block size of 128x128 pixels to be optimal for capturing unique textures, and computationally efficient.

We observed that block size was very important aspect of the optimal feature extraction. Too small block size resulted in too similar textures even for different classes because they were not able to capture unique textures to differentiate two unlike classes. Similarly, too big block size had similar effect because of lack of distinguishing texture features. For better capturing of the non-uniform textures, we allow an overlap for block division, which means one block overlaps 50% horizontally and vertically with its neighboring blocks. The reason we do overlapping of block is to not miss the textures which lie in between (either vertically or horizontally) of two different blocks.



(a) (b) (c)

(d)

Figure 2.2 Four different textures in same severe class. Similarly, other classes (images not shown) also have different variations in their textures.

2.3.1 Blocks Filtering and Normalization

UC images contain black borders and specular reflections which means some of the extracted blocks would contain black borders and specular reflections if processed without filtering. Also, some of the blocks may contain both very dark region and very bright region within a block making texture feature inconsistent. We filter out these types of blocks so that only good blocks like Figure 2.3 (d) are passed through the feature extraction process. To filter out blocks, first we separate red, green, and blue channels from the original RGB block and process each channel separately.

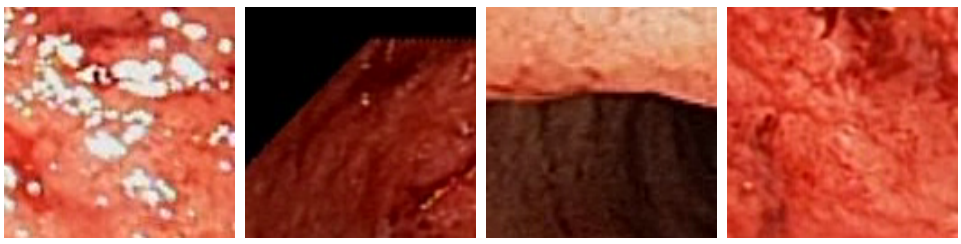
The actual values of the thresholds used for block filtering are summarized in Table 2.1, which are determined experimentally. The threshold values are determined one at a time using the entire images (total 207). Once the block filtering is done each used RGB block such as shown in Figure 2.3 (d) are converted into grayscale blocks for further processing. To make the grayscale properties consistent throughout the procedure, we

normalize the block by subtracting the minimum grayscale value from each pixel in the block.

We experimented with other preprocessing steps on our blocks apart from the normalization but it affected negatively with the accurate classification. We used homomorphic filtering to correct the non-uniform illumination of UC frames. Homomorphic filtering normalizes the brightness and increases the contrast across the image simultaneously [25]. Applying this filter prior to feature extraction reduces the likelihood of extracting erroneous features because of uneven illumination. But after applying this filter, we observed that the features for 'mild' and 'moderate' blocks were too similar resulting in misclassifications. So, we opted to use only grayscale normalization as preprocessing step.

Table 2.1 Block filtering parameters and thresholds used in UC experiments. These values are obtained based on 207 training images.

Parameters	Threshold values
Specular Reflection Pixel	$Specular_{Thld} = 0.8$
Specular Pixel Percent	$SPP_{Thld} = 20\%$
Black Border Pixel	$BP_{Thld} = 0.05$
Black Border Pixel Percent	$BBPP_{Thld} = 5\%$
Block Standard Deviation	$BSD_{Thld} = 0.3$



(a) (b) (c) (d)

Figure 2.3 Discarded blocks due to a) Specular Reflection, b) Black Borders, and c) High Standard Deviation. d) Example of a used block.

2.3.1.1 Blocks with Specular Reflection

Figure 2.3 (a) is an example of block with specular reflection. To filter out those blocks, first we separate red, green, and blue channels from the original RGB color in each block and normalize them so that the intensity value range for each channel becomes between 0 and 1.

Let $I(x_i, y_j)$ represents the pixel intensity at i^{th} row and j^{th} column of the block I with size M x N where M represents number of rows and N represents number of columns. The total number of specular pixels is calculated as

$$\text{SpecularPixels} = \sum_{i=1}^{M} \sum_{j=1}^{N} S\left(I(x_i, y_j)\right) \qquad (2.1)$$

$$\text{where, } S\left(I(x_i, y_i)\right) = \begin{cases} 1, \text{ if } I(x_i, y_i) > Specular_{Thld} \ \forall \text{ channels} \\ 0. \text{ Otherwise} \end{cases}$$

Here, $Specular_{Thld}$ is set based on Table 2.1 and is calculated experimentally. Once all specular pixels are found, we calculate its percentage over the entire block. If this percentage is greater than a threshold (SPP_{Thld}), we discard the block from further processing.

2.3.1.2 Blocks with Black Borders

First, blacks pixels are determined similar to specular pixels as given in equation (2.1) but key difference is that $S(I(x_i,y_i))$ is 1 if $I(x_i,y_j) < BP_{Thld}$ for all channels and 0 otherwise. Here, BP_{Thld} is set based on Table 2.1. If all RGB channel values of a pixel are less than a threshold (BP_{Thld}), it will be considered as a black pixel. If the black pixel percentage of a block is greater than a threshold ($BBPP_{Thld}$), we discard it. Figure 2.3 (b) is an example of discarded black border block.

2.3.1.3 Blocks with High Standard Deviation

Some of the blocks like Figure 2.3 (c) have very high uneven illumination which may provide incorrect characteristics of textures. The uneven illumination is characterized by calculating the standard deviation of the gray values of all the pixels in the block. We discard those blocks by thresholding with standard deviation. If a standard deviation of a block given by equation (2.2) is greater than a threshold (BSD_{Thld}), we discard it.

$$BlockSD = \sqrt{\frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(I(x_i, y_j) - \mu\right)^2} \qquad (2.2)$$

where, $\mu = \frac{1}{M*N}\sum_{i=1}^{M}\sum_{j=1}^{N} I(x_i, y_j)$

2.4 Proposed Feature Extraction Method

We consider a window of 3x3 pixels as shown in Figure 2.4, in which an average of the absolute differences between the center pixel (P_c) and its eight neighbors (P_1 ~ P_8) is calculated using equation (2.3), where n = 8.

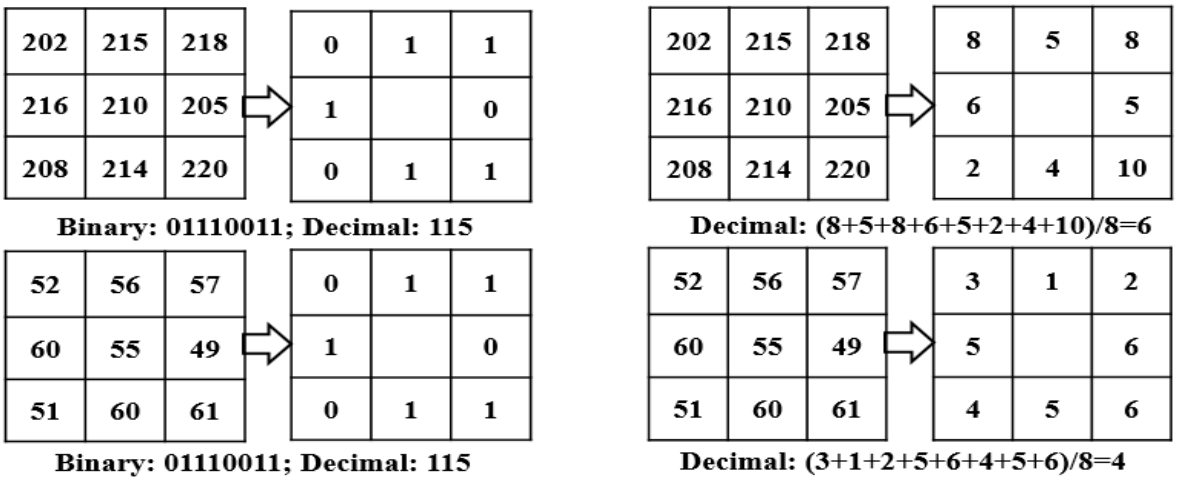
$$DIFF(P_c) = \frac{1}{n}\sum_{k=1}^{n} |P_k - P_c| \qquad (2.3)$$

P_1	P_2	P_3
P_4	P_c	P_5
P_6	P_7	P_8

Figure 2.4 A 3x3 pixel neighborhood; P_c represent the center pixel and all others are its neighbors.

This process is repeated over the entire block as 3x3 window (Figure 2.4) slides across the block one pixel at a time. One reason why we consider 3x3 window is to provide a fair comparison with LBP where its best accuracy is achieved with this window size [4]. A

major difference between our new texture feature and other existing methods is that it considers not only patterns of pixel differences but also what center pixels are associated with them. A comparison of between LBP and DIFF is shown in Figure 2.5. As seen, LBP considers which pixel value is larger among a center and its neighbors, but does not consider how much larger whereas DIFF does retain the pixel value difference when it compares a center with its neighbors. In other words, DIFF considers actual difference of center pixel value and neighbors for texture feature whereas LBP only considers the sign of center pixel and neighbors for texture feature. Similarly, DIFF considers the contrast of local image texture whereas LBP does not consider contrast of local image texture.



(a) LBP (b) DIFF

Figure 2.5 Difference in feature extraction method between proposed DIFF vs existing method Local Binary Pattern (LBP). DIFF considers contrast of local image texture whereas LBP does not considers the contrast of local image texture.

For a block of 128x128 pixels, 15,876 (126x126) DIFF values are generated after excluding border pixels. Minimum and maximum possible values of DIFF are 0 and 255, respectively. Also, possible values of P_c are between 0 and 255. Therefore, the number of occurrences of these values can be represented as a matrix (256x256) in which its

columns and rows represent different DIFF and P_c values, respectively as shown in figure 2.6. Now, the texture of a block is represented by 65,536 (256x256) numbers.

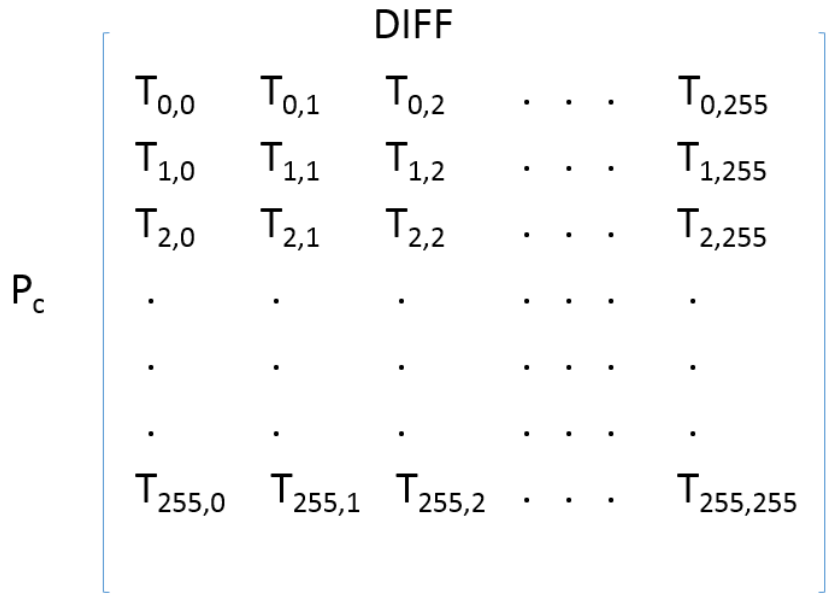


Figure 2.6 256 x 256 matrix representation of DIFF textures. Column values represents the DIFF values given by equation 2.3; i.e., $T_{m,n}$ represents the frequency for DIFF value n with center pixel value m. Row values represent the center pixel value. This is the maximum possible size of the texture representation of a block.

We can significantly reduce this huge texture size by quantization. The quantization of the center pixel (P_c) values is straightforward. Since it has 256 values, it can be quantized into any number by dividing by 2^m (m = 1, 2, …, 8). In our case we quantize into 16 values ($256/2^4$), 8 value ($256/2^5$) and so on. The quantization reduces the size of the feature vector and thereby accelerates the performance whereas too much quantization may generate unreliable features depending on the nature of textures in the images as seen in the results in experimental section.

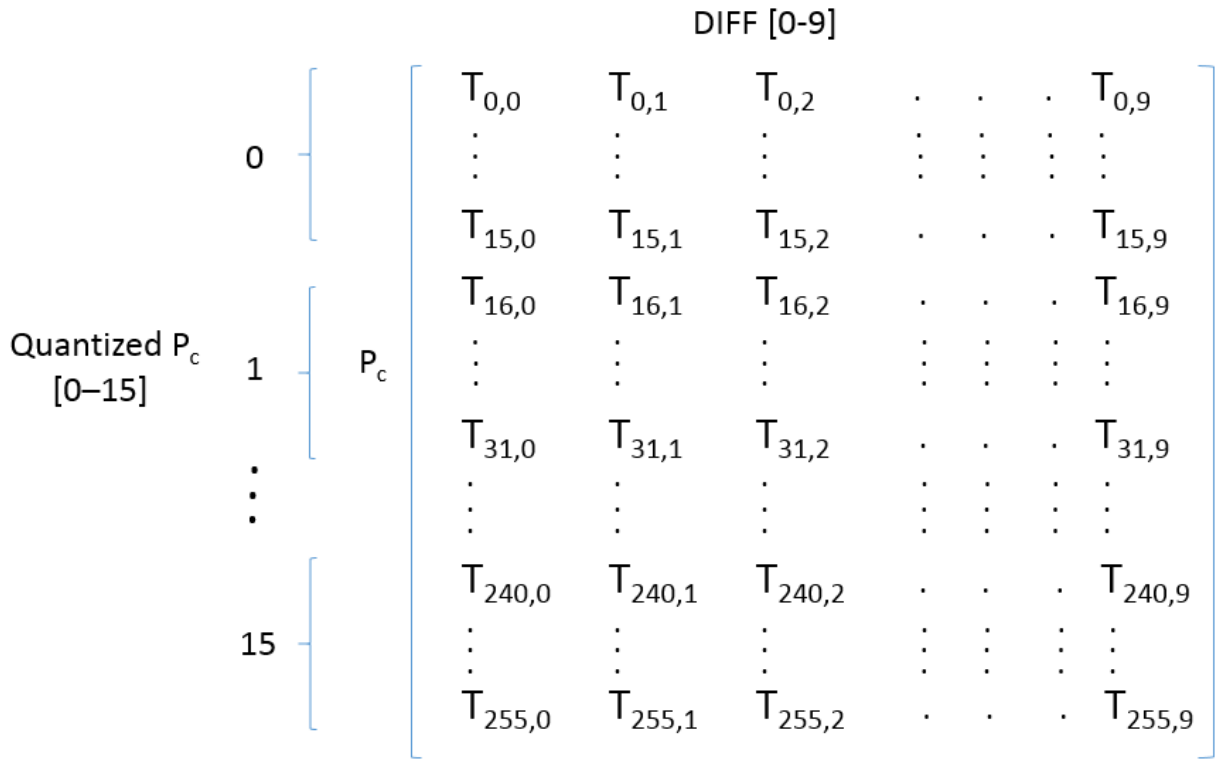


Figure 2.7 Quantization of original 256 x 256 texture. In this DIFF_16_10 texture, we have 16 groups of center pixels and first 10 DIFF values [0-9]. The 256 different center pixels are quantized into 16 groups each containing equal range of values. Center pixels values [0-15] goes to group 1, [16-31] goes to group 2 and so on. This way original 256 x 256 matrix is reduced to new 16 x 10 matrix which results in 160 bin size feature vector.

DIFF values generated by equation (2.3) for the blocks of our colonoscopy images are typically less than 50, and mostly less than 10 based on the observation of entire UC images (207 images in total). In fact, the first 10 DIFF values (i.e., the first 10 bins in the histogram) represent more than 95% DIFF values of the entire block for all UC classes. The reason is that the pixel value differences in a 3x3 window are very small since neighboring pixels are very similar. This feature of our DIFF method is key for significantly reduced feature vector size. This is where our proposed method excels in terms of processing time as less time is spent extracting the features from images. It will be shown

in the experimental section that how we could achieve competitive accuracy even keeping our feature vector size very low and gain significant speed performance.

We consider some combinations of quantized center pixel (P_c) values with quantized DIFF values as new features such as 'DIFF_16_10' with 16 center pixel (P_c) values and 10 DIFF values. Similarly, 'DIFF_16_50' with 16 center pixel (P_c) values and 50 DIFF values, 'DIFF_1_10' with one center pixel (P_c) value and 10 DIFF values, and so on. The selection of DIFF values and center values for a particular feature is dependent on the characteristics of textures in the blocks. For example, we observed that 'severe' and 'moderate' classes which have more non-uniform and random textures need more center values to be able to extract distinguishable features. The results can be seen in the experimental section. This feature method can be applied to other non-medical images as well if processing time is the key. It can be especially utilized with the system with real-time performance requirement.

2.5 Proposed Hybrid Method

Before we discuss the hybrid method, we summarize the overall procedure first. It has two main phases: Training and Testing. For Training, each input image in all five classes is divided into a number of blocks, and the block filtering and normalization are applied. A selected feature is computed for all blocks, and used to train a KNN (k-nearest neighbors) classifier [26] with k=1. KNN is one of simplest machine learning algorithms and quite accurate as compared to others for our medical dataset. An image block is classified by a majority vote of its k nearest neighbors. If k = 1, then the image block is simply assigned to the class of that single nearest neighbor. We experimented with different values of k, but found k=1 giving best results for our dataset. We also tested

other classifiers such as Decision Tree and Naïve Bays [26], but their results were worse than KNN and we exclude their results.

For Testing, a test image is divided into number of blocks, and the same block filtering and normalization as used in Training are applied to all blocks in the test image. Using the trained KNN classifier from the training phase, we determine for each block to which class it belongs. Lastly, we calculate the UC class probability of each test image by dividing the number of blocks for each class by the total number of blocks. The UC class with the largest probability value is selected as the UC class of the test image. In other words, the class of a test image is the class that most blocks of that image belong to. If there is a tie, the more severe UC class is selected as its class. For example, if there is the same number of 'moderate' and 'mild' blocks in a test image, it is classified as 'moderate'.

As mentioned earlier, any one feature could not provide acceptable accuracies for all five classes. We propose a simple hybrid method which provides a better accuracy. Here, we are discussing a hybrid of our DIFF with LBP as an example. In the Training phase, we train two KNN classifiers: one for LBP and the other for DIFF. In Testing, a test image is evaluated by the two classifiers in which each classifier provides the UC class that the test image belongs to with its probability value. The results of the two classifiers either are the same UC class or two different UC classes, we take the result with the greater probability value. Figure 2.8 shows step by step flow chart of the process for the hybrid classification. As the number of features increases, the computation cost also increases. But the combination of more features does not always result in better accuracy and it is also computationally expensive. We mainly focus on combination of two features,

but any combination of multiple features is possible. As mentioned earlier, the number of features needed for the hybrid will also depend upon the texture types of the images.

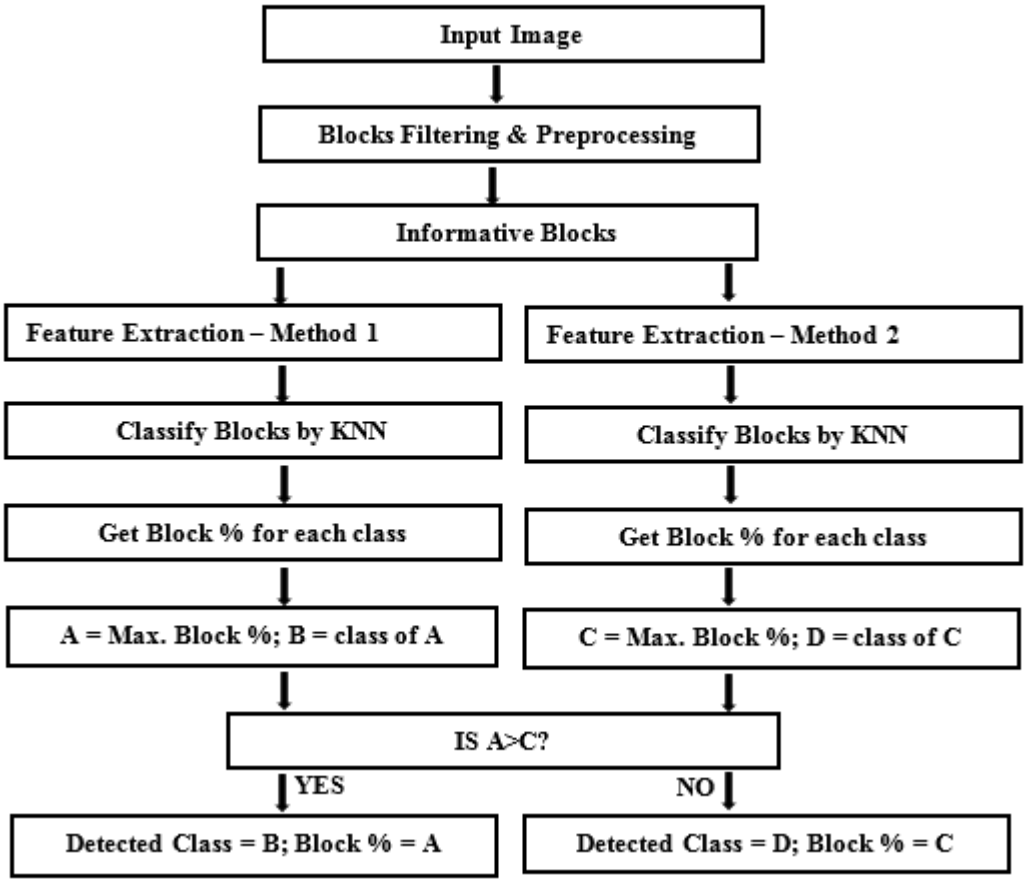


Figure 2.8 Detection method in hybrid approach. Two different feature methods are used and classified individually. The best result among two is picked as the final classification result.

2.6 Experiments

In this section, we assess the effectiveness of the single features including the proposed feature (DIFF), and the hybrid methods with multiple features. All experiments were conducted on a Windows 7 64-bit PC with Intel i7 2.8GHZ processor and 6GB RAM using MATLAB R2014a. For Training and Testing, we used 10-fold cross validation [26]. We chose this validation because manually dividing images into training and testing

makes the results inconsistent for different sets of images. Another reason for opting to 10-fold cross validation is due to limited training and testing images. For each of the 10 cross validations, we get 10% images from each class for testing, and all remaining images are used to train the KNN classifier which classifies the test images and their blocks. When all images of a class are evaluated for a cross validation, we calculate the image level accuracy and block level accuracy for the class using equation (2.4) and (2.5) below.

$$ImageAccuracy = \frac{TP_{Image}}{N_{Image}} x\ 100\% \quad (2.4)$$

$$BlockAccuracy = \frac{TP_{Block}}{N_{Block}}\ x\ 100\% \quad (2.5)$$

Here, TP_{Image} is the total number of correctly classified images (which means the image of a UC type is matching with its actual type), similarly TP_{Block} is the total number of correctly classified blocks. FN_{Image} is the total number of incorrectly classified images (which means an image of one UC type is mistakenly classified as another UC type or normal type), similarly FN_{Block} is the total number of incorrectly classified blocks.

$N_{Image} = TP_{Image} + FN_{Image}$, and $N_{Block} = TP_{Block} + FN_{Block}$.

Table 2.2 Colonoscopy images and blocks used in the experiments. The images were annotated by domain experts. The blocks are the only good blocks after filtering out unnecessary blocks in the preprocessing stage.

Severity type	No. of Images	No. of Blocks
Severe	40	1,500
Moderate	45	1,698
Mild	50	1,886
Scar	22	685
Normal	50	1,949
Total	207	7,718

Once a cross validation is completed, the same process is repeated for the remaining cross validations, and the final result is obtained as the average of all 10 cross validations. So, all the results presented below are average of 10 fold cross validations.

2.6.1 Single Features – Existing Feature Methods

For the single feature comparisons, we compare Higher Order Local Auto Correlations (HLAC) [10], four versions (LBP256, LBP59, LBP10, and Local variance method (LOCAL_VAR256)) from the original Local Binary Pattern [4], Gabor filter banks (GABOR) [15], Leung-Malik filter banks (LM) [16], a modified version of LBP (MOD_LBP) [17], the traditional textures (Contrast, Correlation, Energy, and Homogeneity) based on Gray-Level Co-Occurrence Matrix (GLCM) [18], and MPEG-7 based texture features (MPEG-7_HTD (Homogeneous Texture Descriptor) and MPEG-7_EHD (Edge Histogram Descriptor)) [19]. Brief explanation of each of these texture feature methods are presented below.

2.6.1.1 Local Binary Pattern (LBP)

LBP (Local Binary Pattern) is a widely used method that describes a local texture patterns. LBP works in a 3x3 pixel block of an image with one center pixel and its 8 neighbors. Although it can be generalized to any size and any neighbors, we only focused on a 3x3 neighborhood because it provides better accuracy according to [4], and it is computationally less expensive than larger neighborhoods. The LBP label of the center pixel is obtained by thresholding neighborhood pixels with the gray value of the center pixel, multiplying with power of 2, and summing them up as indicated in equation (2.6). There are $2^8 = 256$ possible values for LBP labels.

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} S(g_p - g_c) 2^p \quad (2.6)$$

where S(a) = 1 if a ≥ 0; S(a) = 0 if a < 0; P is the number of neighbors in a circular neighborhood or radius R; g_p and g_c represent gray values of neighbor pixel p and center pixel c respectively.

LBP256 contains 256 bins (histogram size) obtained by equation (2.6). The number of bins can be reduced to 59 by considering uniform patterns only. LBP59 contains 59 bins where the first 58 bins are 58 uniform patterns, and the last bin is everything else. Similarly, LBP10 is obtained based on rotation invariant uniform LBP. Uniformity is measured based on U value which is number of spatial transitions (bitwise 0/1) in the pattern. LBP operator which is both rotation invariant and uniform with U value 2 can be obtained using equation 2.7. It contains 10 bins where the first 9 bins contain the 9 rotation invariant uniform patterns, and the last bin contains all remaining 'non-uniform' patterns.

$$LBP_{P,R}^{ri\mu 2} = \begin{cases} \sum_{p=0}^{P-1} S(g_p - g_c) & if\ U(LBP_{P,R}) \leq 2 \\ P+1 & otherwise \end{cases} \quad (2.7)$$

where

$$U(LBP_{P,R}) = |S(g_{p-1} - g_c) - S(g_0 - g_c)| + \sum_{p=1}^{P-1} |S(g_p - g_c) - S(g_{p-1} - g_c)|$$

The local variance method (LOCAL_VAR256) is obtained also as described in [4], which takes the mean of 8 neighbors, and subtracts the mean from each of those neighbors. For our experiment, we set the number of bins to 256. A modified version of LBP (MOD_LBP) [17] slightly modifies the traditional LBP by multiplying the binary values with the squared difference of neighboring pixel and mean of the 3x3 neighborhood pixels. The total number of bins can remain 256 for this as well.

2.6.1.2 Gabor Filter Banks

Gabor filters are used for texture discrimination in various types of images [27]. Gabor filters are also widely used in pattern analysis applications [15]. We considered Gabor filter banks with 80-bin feature vectors (GABOR). First, we obtain 40 different Gabor filters (5 scales and 8 orientations), and convolute each filter with the input block to get 40 different response matrices. After that, we obtain Local Energy and Mean Amplitude by using the response matrices. Local Energy is calculated by summing up the squared values of a response matrix. Similarly, Mean Amplitude is calculated by summing up the absolute values of a response matrix. The [1x40] feature vector from Local Energy and [1x40] feature vector from Mean Amplitude is merged to obtain [1x80] Gabor feature for each image block.

2.6.1.3 LM Filter Bank

We include another popular filter bank called Leung-Malik (LM) [16] which is multi set, multi orientation filter bank with 48 filters. Although LM filter banks are not rotationally invariant, their accuracy is very good [24]. It consists of first and second derivatives of Gaussians at 6 orientations and 3 scales making a total of 36 filters, 8 Laplacian of Gaussian (LOG) filters, and 4 Gaussian filters. Similar Local Energy and Mean Amplitude as in GABOR feature are computed to make a 96 bin feature vector for each block. Only one of the Mean Amplitude or Local Energy could be used as feature vector but we observed that combination of two was generating better accuracy so we opted for combined feature vector.

2.6.1.4 Higher Order Local Auto Correlations

Higher Order Local Auto Correlations (HLAC) [12] is also evaluated where the primitive features are obtained by computing the sums of the products of the gray scale values of the corresponding pixels with 25 local 3x3 masks. HLAC features are used in various areas of image analysis ranging from face recognition to gesture recognition to natural object recognition [12, 28, 29]. One big advantage of HLAC features is they work in mask patterns which is less computation heavy as compared to interpolation [30]. Since our goal is to extract feature in real-time, HLAC feature is very suitable for faster feature extraction.

2.6.1.5 Gray-Level Co-Occurrence Matrix

Traditional texture features based on Gray-Level Co-Occurrence Matrix (GLCM) [18], which shows the relationships between adjacent pixels are also included for the comparisons. The texture segmentation by using different orientations of GLCM are proposed in [31]. Each texture are processed with normalization as well as noise removal. Principle component analysis is used for dimensionality reduction. For our experiment we use four commonly used texture features (Contrast, Correlation, Energy, and Homogeneity).

Contrast which is also known as ‘sum of squares variance’ measures how contrast the image block is. Here, the pixels similar to each other are given the weight zero. On the other hand, homogeneity weights values by the inverse of the contrast weight with decreasing the weights exponentially away from the diagonal. Energy which is also known as uniformity is calculated as square root of Angular Second Moment (ASM). ASM is

simply the sum of squares of the values from GLCM table. The Correlation measures the linear dependency of gray levels of neighboring pixels. By combining these 4 features, GLCM feature vector of size 4 is constructed for each block.

2.6.1.6 MPEG-7 Descriptor

MPEG-7 based texture extraction is also considered in the experiment. We have used Homogeneous Texture Descriptor (HTD) and Edge Histogram Descriptor (EHD) [19]. MPEG7_HTD is composed of a 62 bin feature vector. The first two are the mean and the standard deviation of the image block. The rest are the energy and the energy deviation of the Gabor filtered responses. MPEG7_EHD represents local edge distributions in the image block and is represented by 80 (16 sub-images per image, 5 bins i.e., one bin per each edge type (vertical, horizontal, two diagonals, and non-directional edge) per sub-image) bin feature vector.

2.6.2 Summary of Proposed and Existing Features

Table 2.3 lists the different variations of proposed features and existing features evaluated during the experiments. It can be seen that the feature vector size varies a lot based on different features and their versions and the computation time depends on the feature vector size. Our goal is to get the maximum accuracy possible by using minimum feature vector size without sacrificing the accuracy of the feature algorithm. For our proposed feature method, the feature vector size varies from 10 to 800 but the accuracy does not varies that much which is a good thing; and one important reasons our proposed feature is superior than the existing one in terms of accuracy and speed even though feature vector size is relatively small .

Table 2.3 Summary of different proposed and existing features methods. The feature vector size depends on the feature method algorithm and its variation. The computation time also depends on the feature vector size.

Feature Method	Variants	Feature Vector Size
DIFF	DIFF_1_10	10
	DIFF_1_50	50
	DIFF_8_10	80
	DIFF_16_10	160
	DIFF_16_50	800
LBP (Local Binary Patterns)	LBP10	10
	LBP59	59
	LBP256	256
Local Variance (LOCAL_VAR)		256
Modified LBP (MOD_LBP)		256
Gabor Filter Banks		80
Leung-Malik (LM) Filter Banks		96
Higher Order Local Auto Correlations (HLAC)		25
Gray Level Co-occurrence Matrix (GLCM)		4
MPEG-7	MPEG7-HTD (Homogeneous Texture Descriptor)	62
	MPEG7-EHD (Edge Histogram Descriptor)	80
Discrete Fourier Transform (DFT)[2]		256

[2] Although DFT feature method is discussed and evaluated in chapter 3, it is included here for the comparison purpose with other existing feature methods.

2.6.3 Results with Colonoscopy Dataset

In this section, we present the results of our proposed method and existing methods with our colonoscopy dataset. The test set contains 207 images from five different classes (‘severe’, ‘moderate’, ‘mild’, ‘scar’ and ‘normal’) provided by our domain expert. The images were taken from a collection of several videos. For convenience, we call this dataset ‘UC dataset’. The details of images and corresponding blocks for each class are shown in Table 2.2. Overall, we compare 16 different single features discussed above including our DIFF_1_10, DIFF_1_50, DIFF_8_10, DIFF_16_50 and DIFF_16_10. The detailed results can be found in Table 2.4, which are the averages from the 10-fold cross validations.

The best image level accuracies for ‘severe’ and ‘moderate’ classes are achieved by DIFF_16_50 and DIFF_16_10, which are generating very similar accuracies. For ‘mild’ class, the best accuracy was achieved by several different features. The best image level accuracy for ‘scar’ is achieved by LBP10, DIFF_1_10, and DIFF_1_50. And the best image level accuracy for ‘normal’ classes is achieved by LBP10. The best accuracy for all classes is achieved by LBP10, but it is less than 84%. A little bit less, but similar accuracies (80~82%) are achieved by DIFF_1_10, DIFF_1_50, DIFF_16_50 and DIFF_16_10. Overall these three features perform similarly in the image level. At the block level, LBP10 is a little better than the others only for ‘scar’ and ‘normal’ images. Figure 2.11 shows some of the examples of misclassified images by single features such as LBP10 and DIFF_16_10.

2.6.4 Results with General Datasets

For more objective comparisons of single features (DIFF_1_10, DIFF_16_10 and LBP10), we evaluated these with two popular texture datasets in [32] (called CUReT) and [33] in which their ground truths are known. For convenience, we call the dataset in [32] as ‘dataset1’, and the dataset in [33] as ‘dataset2’. ‘dataset1’ includes five texture types (bread, concrete, loofa, skin, and sponge) in which each type has 45 images. Similarly, ‘dataset2’ includes five texture types (T01, T04, T07, T08, and T09) in which each type has 40 images. Both datasets were tested in similar fashion as described above for testing UC images.

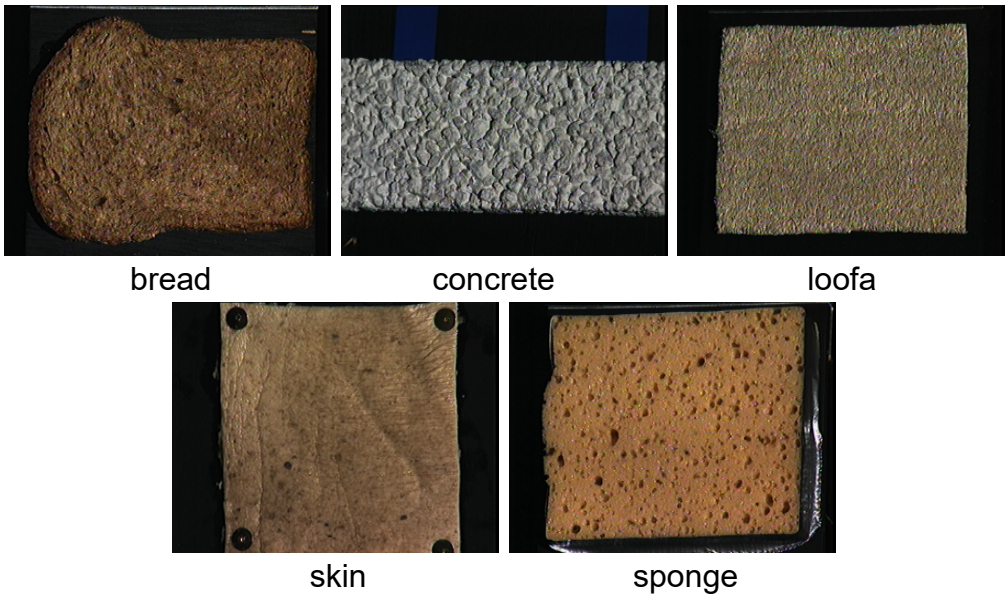


Figure 2.9 Five image types selected for dataset1. These image types are selected at random from the pool of several images.

For ‘dataset1’ (225 images with 5,525 blocks), the average image and block level accuracies for DIFF_16_10 were 99.6% and 97.8% respectively, whereas they were 99.1% and 95.9% for LBP10. Similarly, DIFF_1_10 resulted in 94.1% and 83.6% respectively for ‘dataset1’. For ‘dataset2’ (200 images with 2,400 blocks), the average

image and block level accuracies for DIFF_16_10 were 97.5% and 90.5% respectively, whereas they were 99.0% and 93.1% for LBP10. Similarly, DIFF_1_10 resulted in 93.5% and 83.0% respectively for ‘dataset2’. As the results show, DIFF_16_10 performs

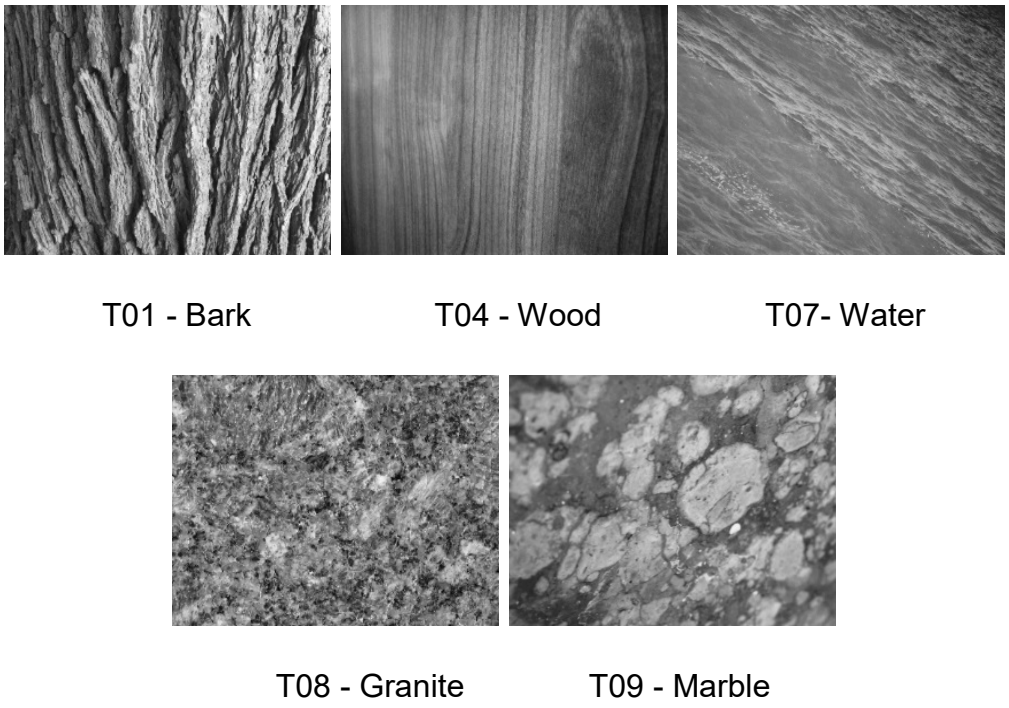


Figure 2.10 Five image types used for dataset2. These images were also selected at random from the pool of several images.

as good as LBP10 which is one of the most popular texture detection methods. But more importantly, our DIFF_16_10 is better than LBP10 in terms of execution speed. More details about the computation costs will be discussed later.

This shows that our proposed DIFF methods not only works for medical images but also with other well-known dataset images. Because of its low cost of computation and high accuracy for feature extraction and classification, it can be used in different fields of image processing and computer vision. As discussed in proposed feature extraction method section, we can quantize the DIFF features to best suit the image types of

consideration. We will show in chapter 3 that how versatile the DIFF features can be for other types of images even for the same colonoscopy image domain.

2.7 Results of Hybrid Methods with Multiple Features

For the hybrid approaches, we combine the best performing features such as LBP10, DIFF_1_10, and DIFF_16_10. Even though the accuracies of GABOR, LM, and MPEG7_HTD are less than those of the best performing single features as seen in Table 2.4, we include them in the hybrid approaches for comparison purposes. The hybrid approaches improve both image and block level accuracies significantly.

By using the combination of DIFF_16_10 and LBP10, we achieved 90.1% image level and 68.7% block level accuracies. We tested the combination of DIFF_16_10 and DIFF_1_10 with 84.1% image level and 61.3% block level accuracies. Other combination DIFF_1_10 with MPEG7_HTD generated 86.1% image level and 62.8% block level accuracies. We also tested the combinations of three feature methods involving DIFF_16_10, LBP10, MPEG7_HTD, LM, and GABOR. Again, their results are similar with two feature methods, but their execution cost is higher (data not shown). We also tested the combinations of other features, but we did not include the results here because there were very few differences. In terms of image and block level accuracies as well as execution time, the combination of DIFF_16_10 and LBP10 is best for our UC images. This is because the DIFF_16_10 works well for ‘severe’, and ‘moderate’ classes whereas LBP10 works well for ‘scar’ and ‘normal’ classes. Also, we observed that almost all hybrid methods works well for ‘mild’ class.

Table 2.4 Image and Block level accuracies (Unit: %) where IL = Image Level accuracy and BL = Block Level accuracy. The result is the average of 10 fold cross validations. The results after the bold horizontal line are for hybrid methods.

Features	Severe		Moderate		Mild		Scar		Normal		Average	
	IL	BL	IL	BL	IL	BL	IL	BL	IL	BL	IL	BL
LBP10	70.0	50.3	73.0	51.0	92.0	64.4	90.0	76.0	94.0	82.1	83.8	64.8
LBP59	57.5	46.5	77.0	56.7	90.0	62.1	81.7	61.3	92.0	65.4	79.6	58.4
LBP256	65.0	47.4	81.0	56.3	92.0	60.9	66.7	58.6	88.0	63.2	78.5	57.2
GABOR	77.5	48.1	75.0	52.5	78.0	46.3	71.7	48.1	56.0	42.3	71.6	47.5
HLAC	47.5	37.0	63.0	44.1	88.0	49.4	66.7	48.8	72.0	50.5	67.4	46.0
MOD_LBP	70.0	43.5	74.5	43.3	68.0	46.4	61.7	43.3	74.0	47.9	69.6	44.9
LOCAL_VAR256	72.5	49.7	72.5	51.1	88.0	58.4	71.7	56.6	82.0	65.4	77.3	56.2
GLCM	70.0	40.7	82.5	42.2	74.0	38.5	55.0	39.6	50.0	38.6	66.3	39.9
LM	77.5	45.1	79.0	51.1	92.0	47.6	66.7	47.1	62.0	42.8	75.4	46.7
MPEG7_HTD	75.0	57.5	89.5	57.4	80.0	53.5	75.0	53.5	64.0	46.5	76.7	53.7
MPEG7_EHD	25.0	34.7	76.5	62.6	92.0	61.9	33.0	22.7	54.0	41.9	50.2	44.7
DIFF_1_10	62.5	45.2	74.5	47.9	92.0	62.7	90.0	63.8	88.0	69.4	81.4	57.8
DIFF_1_50	65.0	45.3	68.0	48.4	92.0	63.2	90.0	63.7	90.0	69.2	81.0	58.0
DIFF_8_10	77.5	54.3	77.0	58.0	84.0	55.7	71.7	50.4	78.0	58.3	77.6	55.3
DIFF_16_10	82.5	56.1	94.0	60.2	92.0	54.5	61.7	42.5	72.0	54.5	80.4	53.5
DIFF_16_50	82.5	56.4	94.0	60.5	92.0	54.8	61.7	42.5	72.0	54.5	80.4	53.7
DIFF_16_10+LBP10	87.5	59.9	96.0	63.8	92.0	65.6	85.0	74.0	90.0	80.4	**90.1**	**68.7**
DIFF_16_10+DIFF_1_10	75.0	54.5	89.0	62.0	94.0	64.1	76.7	57.6	86.0	68.1	84.1	61.3
DIFF_1_10+LM	65.0	48.6	79.0	55.8	92.0	63.6	78.3	57.7	90.0	68.6	80.9	58.9
DIFF_1_10+MPEG7_HTD	75.0	56.4	85.0	59.7	92.0	65.3	86.7	64.0	92.0	68.4	86.1	62.8
DIFF_16_10+LBP10 +GABOR	87.5	63.6	91.5	65.5	98.0	69.0	85.0	74.7	92.0	81.6	90.8	70.9
DIFF_16_10+MPEG7_HT D +LM	85.0	62.5	86.5	66.0	94.0	60.4	68.3	49.1	78.0	56.4	82.4	58.9

Table 2.5 Contribution of DIFF_16_10 and GABOR with LBP10 as hybrid methods. DIFF_16_10 contributes more in the hybrid method DIFF_16_10+LBP whereas GABOR feature method contributes less in LBP10+GABOR hybrid method.

Class	DIFF_16_10+LBP10		LBP10+GABOR	
	DIFF_16_10	LBP10	LBP10	GABOR
Severe	62.5	37.5	67.5	32.5
Moderate	68.8	31.2	60.0	40.0
Mild	32.0	68.0	80.0	20.0
Scar	9.0	91.0	100.0	0.0
Normal	16.0	84.0	96.0	4.0

Table 2.5 shows the contribution of feature methods while deciding the final classification in the hybrid approaches. It is seen that DIFF_16_10 decides 'severe' and 'moderate' classes on average 62.5% and 68.8% respectively in DIFF_16_10+LBP10 hybrid. On the other hand, majority of the classes were decided by only LBP10 in LBP10+GABOR which shows GABOR is not significant in the hybrid of LBP10+GABOR.

(a) (b) (c)

Figure 2.11 Examples of some of the misclassified images: a) 'severe' image misclassified as 'moderate' by LBP10, but classified correctly by DIFF_16_10, b) 'scar' image misclassified as 'normal' by DIFF_16_10, but classified correctly by LBP10, and c) 'mild' image misclassified as 'moderate' by LBP10 as well as DIFF_16_10.

Figure 2.11 shows some of the misclassified examples by DIFF_16_10 and LBP10 features when they are used as hybrid. We observed that the accuracy level of these methods varies based on class types. For example – 'severe' and 'moderate' class were

less accurately detected my LBP10 whereas ‘scar’ and ‘normal’ images were more accurately detected by LBP10. On the other hand, ‘mild’ class was the most accurately detected class by both methods.

One of the reasons LBP10 was not able to differentiate between ‘severe’ and ‘moderate’ class consistently is because of its inability to extract distinguishable features. By observing the entire blocks of ‘severe’ and ‘moderate’ classes from the training set, we found that the average of feature vectors for ‘severe’ and ‘moderate’ class were very similar as seen in Figure 2.12. The graph of ‘severe’ and ‘moderate’ class almost overlap which means that there is high probability that a classifier may incorrectly classify a ‘severe’ image into a ‘moderate’ image and vice versa. On the other hand, the average feature vector for ‘mild’, ‘scar’, and ‘normal’ classes do not overlap which means the chance of misclassification is lower for those classes and the experimental results shown in Table 2.4 exactly demonstrate this fact.

In the hybrid method, we are taking the maximum probability from two feature methods which means that there is a chance of picking an incorrect classification. For example, let’s say method 1 classifies an image M incorrectly into class C. Similarly, method 2 classifies the same image correctly into class D. Since we are using method 1 and method 2 as hybrid approach, we pick the result from the method which classifies with higher probability. So, if the probability of method 1 is higher than method 2, result from method 1 is picked and image is misclassified. This is the main reasons for misclassification in hybrid method and a drawback of our hybrid approach. We tried to tackle this problem by introducing more feature methods in the hybrid approach and take the result based on the majority voting between multiple methods. But we encountered

the similar problems where majority methods could also incorrectly classify. Also, more methods means more processing time which affected the computation cost.

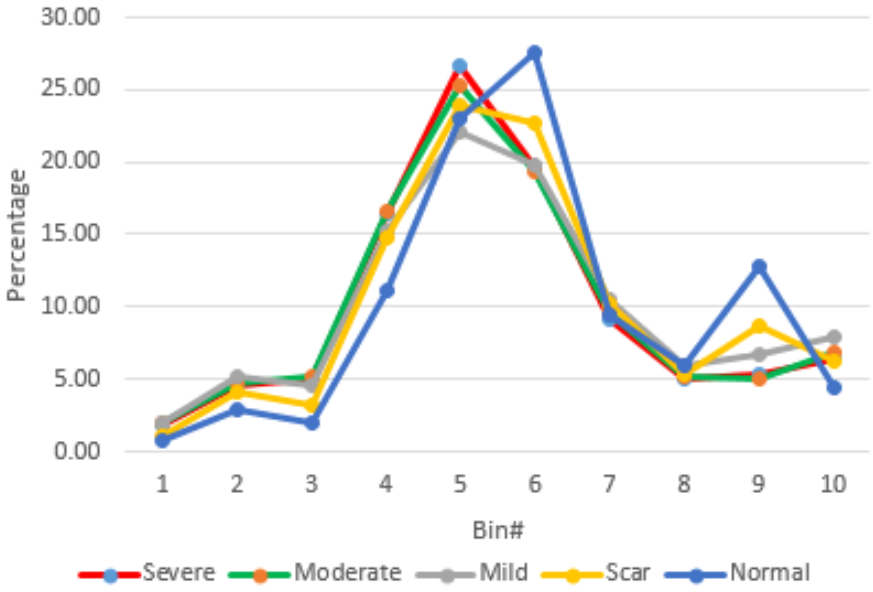


Figure 2.12 The average frequency of each bins in LBP10 for 5 classes. Here, 'severe' and 'moderate' class almost overlap causing higher misclassification among them.

Figure 2.13 and 2.14 show the image level and block level comparisons for our proposed feature method and some of the existing feature method as histograms. It can be seen that the hybrid method significantly increases the classification accuracy. We observed that the accuracy does not always increase in hybrid methods if the combination of hybrid methods are not optimal. Also, some of the feature methods are computationally expensive which results in computationally expensive hybrid methods. Our goal is to maximize the accuracy by keeping the execution speed low. The combination of our DIFF_16_10 and LBP10 was able to achieve that goal as shown in the results.

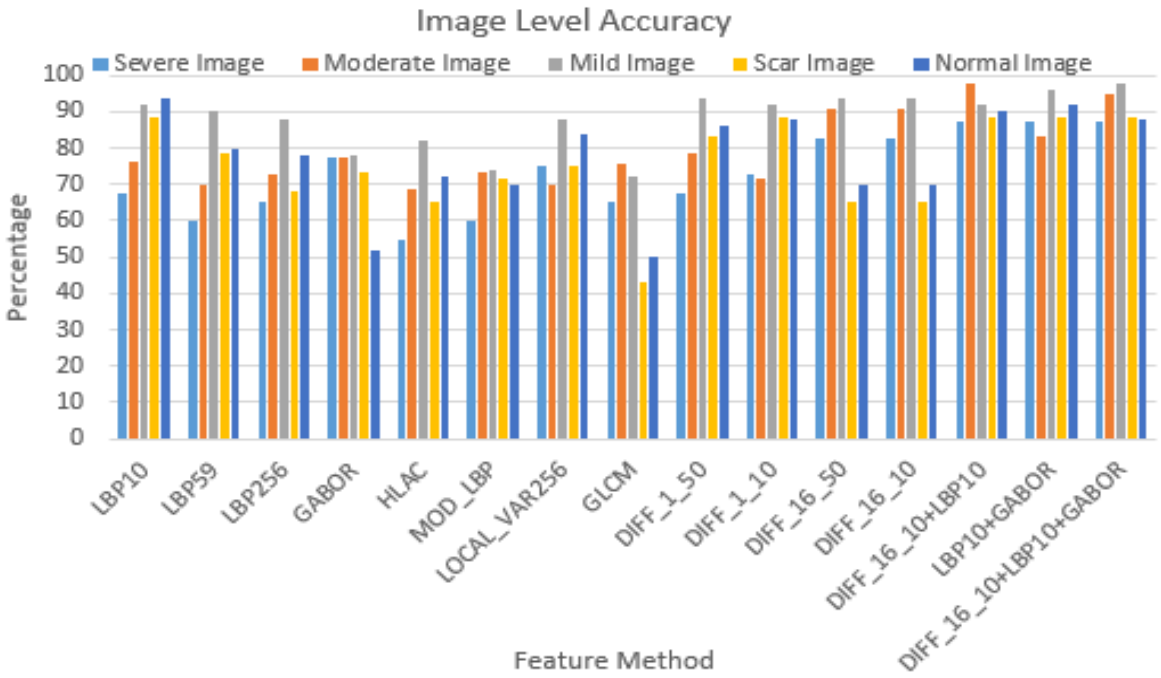


Figure 2.13 Image level accuracy of the average of 10-fold test results for single features and hybrid approaches.

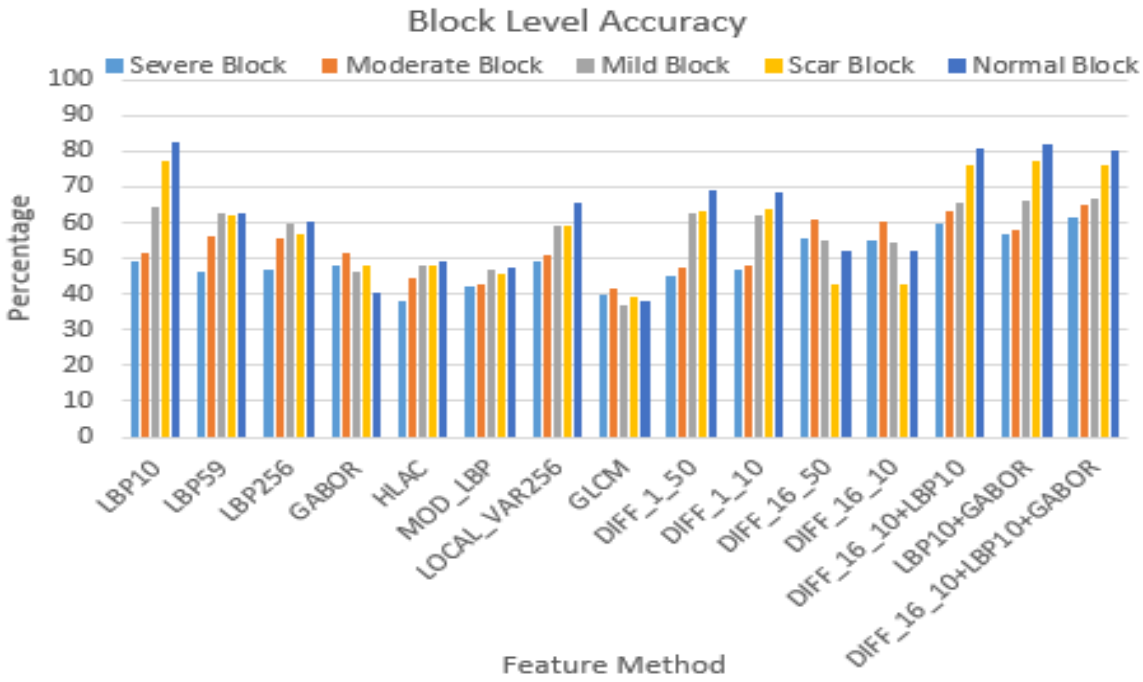


Figure 2.14 Block level accuracy of the average of 10-fold test results for single features and hybrid approaches.

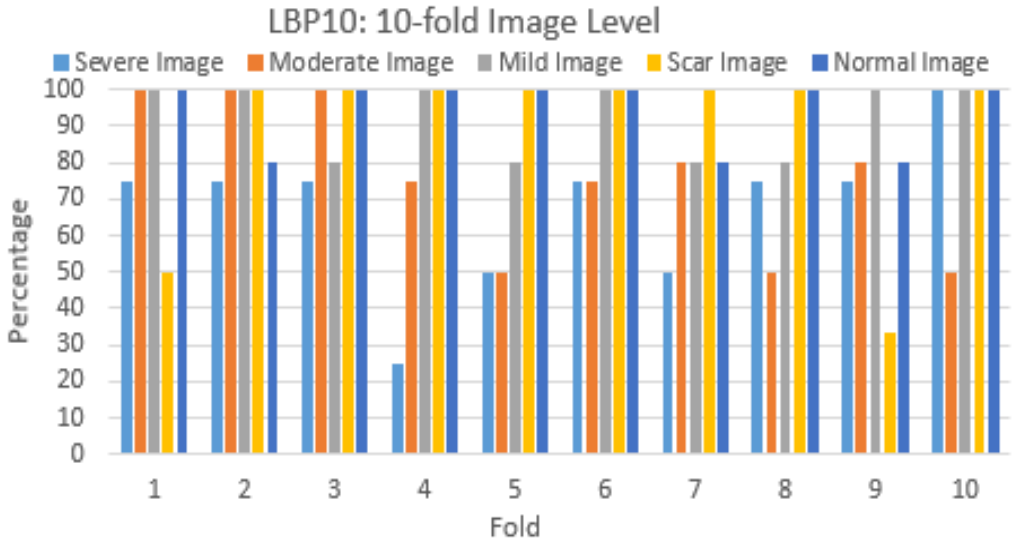


Figure 2.15 10-fold test results for LBP10 Image Level

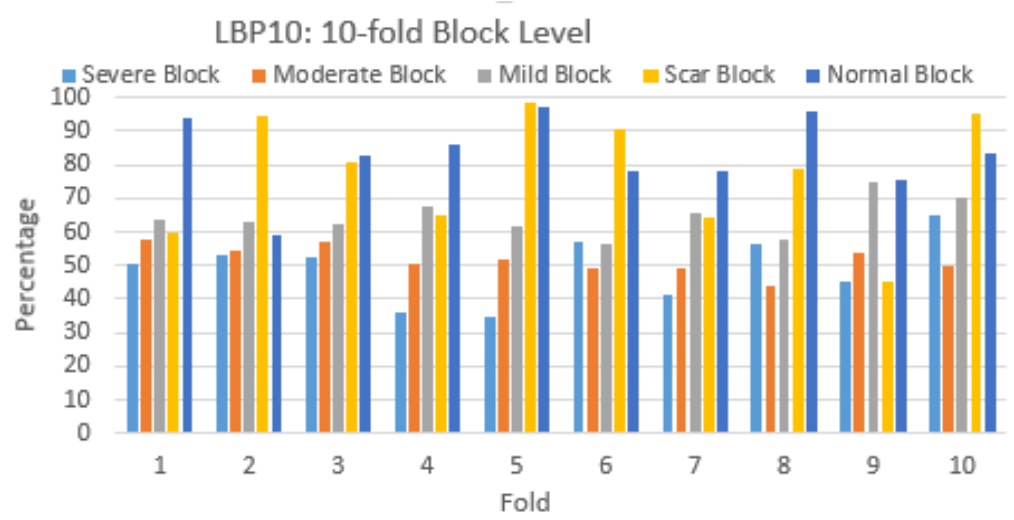


Figure 2.16 10-fold test results for LBP10 Block Level

Figure 2.15 and 2.16 show the 10-fold block level and image level classification accuracy for LBP10 feature method. It can be seen the LBP10 works well for ‘mild’, ‘scar’ and ‘normal’ classes but not so good for ‘severe’ and ‘moderate’ classes. This further proves that LBP10 cannot distinguish effectively between ‘severe’ and ‘moderate’ classes just like it is illustrated by graph in Figure 2.12.

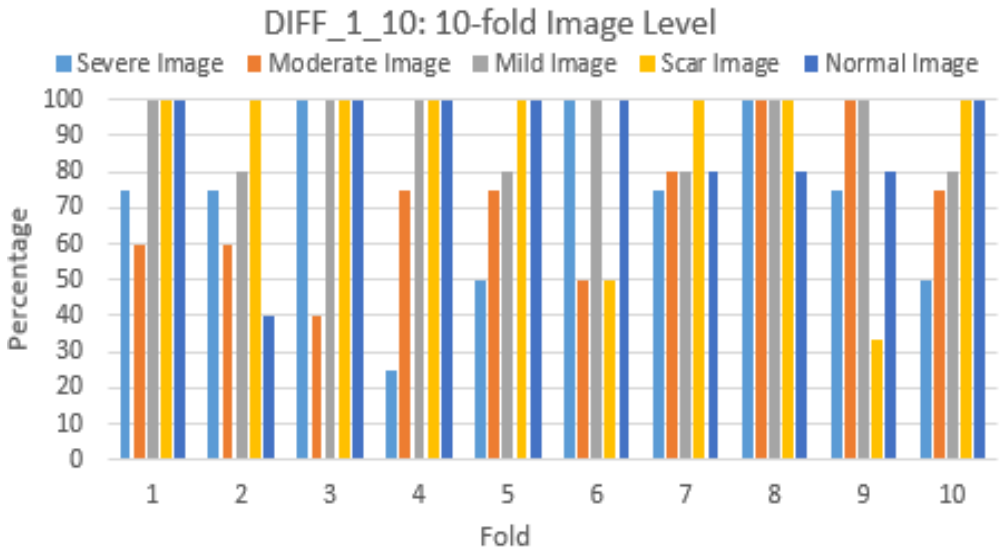


Figure 2.17 DIFF_1_10 Image Level

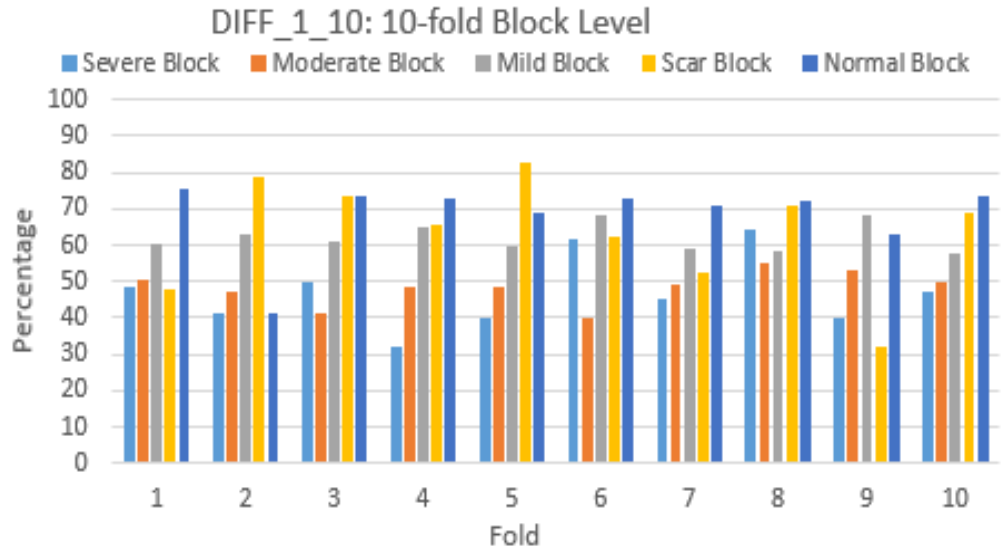


Figure 2.18 10-fold test results for DIFF_1_10 Block Level

Figure 2.17 and 2.18 show image and block level 10-fold accuracy for DIFF_1_10 feature method. It shows that the overall accuracy is not good for ‘severe’ and ‘moderate’ classes as compared to others. In fact, the results of DIFF_1_10 and LBP10 are very close for overall classes. This is the main reason DIFF_1_10 did not improve the accuracy when used in hybrid approaches.

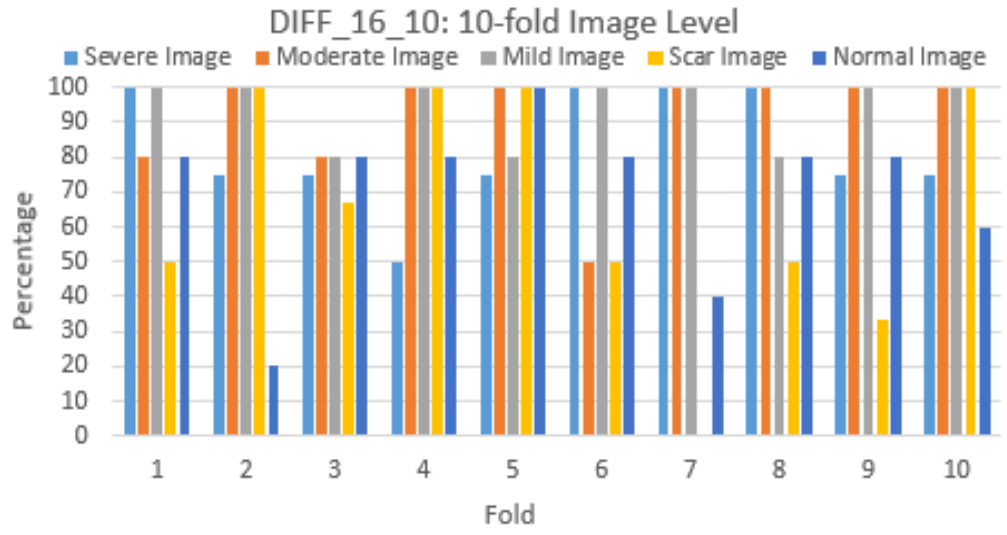


Figure 2.19 10-fold test results for DIFF_16_10 Image Level

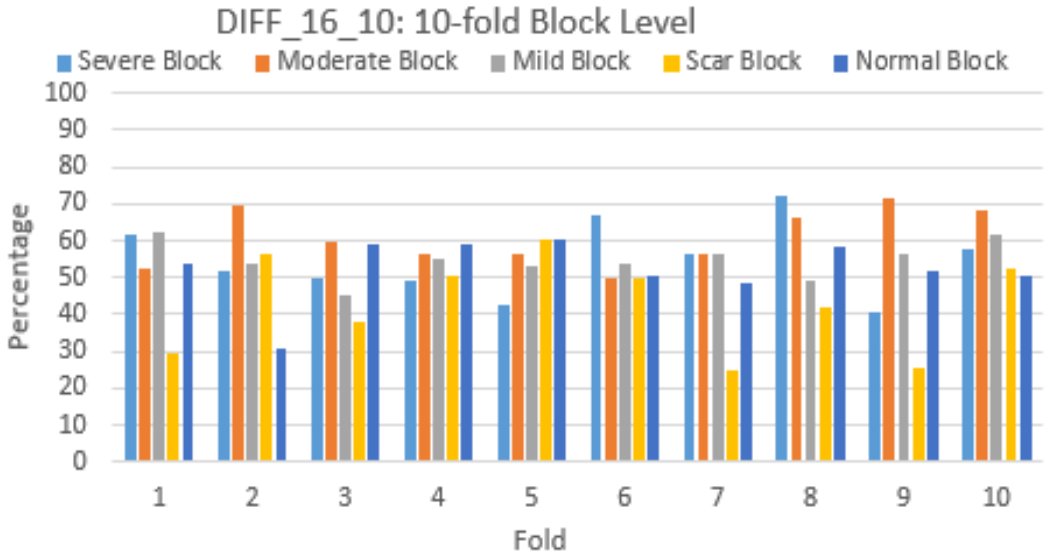


Figure 2.20 10-fold test results for DIFF_16_10 Block Level

Figure 2.19 and 2.20 show that DIFF_16_10 works comparatively better for 'severe' and 'moderate' class than previously discussed LBP10 and DIFF_1_10 methods. Since the results of DIFF_16_10 show better accuracy with 'severe' and 'moderate' class, it is used in DIFF_16_10+LBP 10 hybrid approach.

2.8 Computation Cost Comparison

As mentioned above, we compare the computation costs for some of the best performing single features and hybrid methods in this subsection. The computation cost which is the average execution time of 10-fold validation is computed for training and testing separately on 'UC dataset'. As shown in Table 2.6, both DIFF_1_10 and DIFF_16_10 are at least 2x faster than others during the training phase. Testing includes the feature extraction of testing blocks and their classifications. In terms of speed, our DIFF_1_10 and DIFF_16_10 outperform the existing methods by a huge margin in the testing phase as well. Any hybrid method that includes our DIFF_1_10 and DIFF_16_10 is significantly faster than any other hybrid method, even without considering the computation cost of the training, which is a one-time cost. This shows that our feature method (DIFF) can significantly reduce the execution time for both single feature and hybrid approaches without compromising the accuracy.

Table 2.6 Average of 10-fold computation cost (unit: seconds). Only the best performing feature methods are considered for computation cost.

Feature Method	Training	Testing
DIFF_1_10	**41.1**	**6.0**
DIFF_16_10	47.4	12.7
LBP10	134.2	16.2
MPEG-7_HTD	358.1	72.5
LM	1,751.8	156.6
GABOR	4,074.2	430.8

2.9 Conclusion

It is very difficult to evaluate the severity of Ulcerative colitis (UC) objectively because of non-uniform nature of symptoms associated with UC, and large variations in their patterns. To address this, we objectively measure and classify the severity of UC presented in optical colonoscopy video frames based on image textures. To extract distinct textures, we use a hybrid approach in which a new proposed method (DIFF) based on the accumulation of pixel value differences is combined with an existing method such as LBP. Therefore, our contributions are development of a new texture feature that works well for ‘severe’ and ‘moderate’ UC classes, and to combine this new texture feature with an existing feature to achieve significantly better overall accuracy with significantly less processing time for all UC disease grade classes. The experimental results show that the hybrid method, which can easily be modified for further improvement, already can achieve more than 90% overall accuracy.

Because of the computational efficiency of our DIFF single feature as well as hybrid method, it can be used for other image domains as well especially if the processing time is the key. This image level classification concept can be applied to video level severity score calculation as well as shot segmentation. We plan to extend this work further to video level which works as a feedback system for real-time colonoscopy procedure. Also, the classification criteria can be modified to better reflect the severity of disease rather than the mathematical maximum probability.

CHAPTER 3: ENHANCING INFORMATIVE FRAME FILTERING BY WATER AND BUBBLE DETECTION IN COLONOSCOPY VIDEOS[3]

3.1 Introduction

Colonoscopy is an endoscopic technique that allows a physician to inspect the mucosa of the human colon. It has contributed to a marked decline in the number of colorectal cancer related deaths [1]. However, recent data suggest that there is a significant (4-12%) miss-rate for the detection of even large polyps and cancers [2]. To address this, some research have been conducted investigating an 'automated feedback system' which informs the endoscopist of possible sub-optimal inspection during colonoscopy in order to improve the quality of the actual procedure being performed [3, 35].

A fundamental step of this system is to distinguish non-informative frames from informative ones. An informative frame in a colonoscopy video can be broadly defined as a frame which is useful for convenient naked-eye analysis of the colon mucosa (Figure 3.1). A non-informative frame has the opposite definition where we can not see the colon wall crearly (Figure 3.2). In general, non-informative frames can be considered out-of-focus frames. Informative and non-informative frames can be loosely termed as clear and blurry frames, respectively. An accurate algorithm for this informative frame filtering (IFF) [36-38] has been developed, which is firstly to detect the presence of such vivid lines, and secondly to measure the amount of curvaceous connectivity they possess.

[3] Parts of this chapter have been already published, either in part or in full, from A. Dahal, J. Oh, W. Tavanapong, J. Wong, and P. C. de Groen (2015). Enhancing Informative Frame Filtering by Water and Bubble Detection in Colonoscopy Videos in the proceedings of the International Conference on Health Informatics & Medical Systems, pp. 24-30, July 2015.

Then, with a carefully chosen threshold, frames which exhibit more curvaceous connectivity and classify them as informative, and vice-versa are identified.

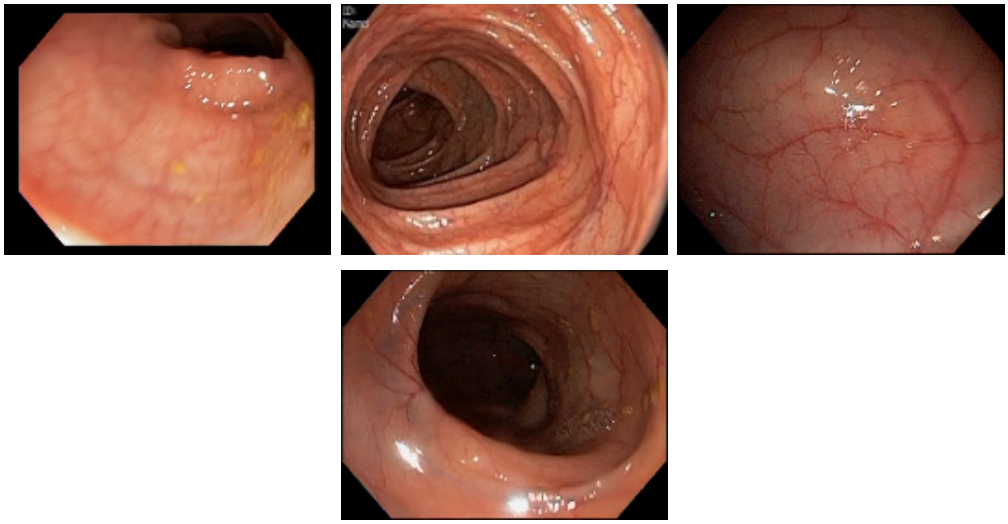


Figure 3.1 Examples of Informative Frames or Clear Frames. The colon wall is clearly visible in these images.

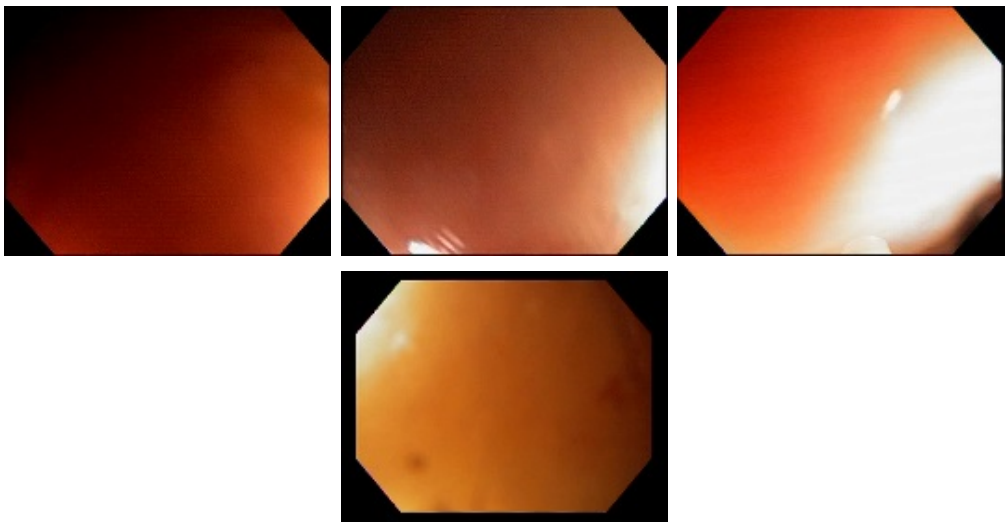


Figure 3.2 Examples of Non-Informative Frames or Blurry Frame. Colon mucosa is not visible in these images.

Figure 3.3 shows some frames having water and bubbles, which do not carry any useful visual information of colon mucosa. These frames need to be classified as non-informative. However, most IFF algorithms [36, 39] classify them as informative since they have clear edges and are in-focus. These types of frames are caused by water injection

for cleaning purpose during the colonoscopy procedure, and need to be discarded from further processing. We define a frame as water or bubble frame if more than 50% of the frame is covered with water or bubble. We call the frames in Figure 3.3 (a-b) 'water' frames, and the ones in Figure 3.3 (c-d) 'bubble' frames for convenience. Based on our observation with 100 colonoscopy videos, the percentage of these frames varies from 5.6% to 20.7% and 9.7% on average. Accurately detecting and discarding water and bubble frames can improve the performance of the 'automated feedback system' mentioned earlier.

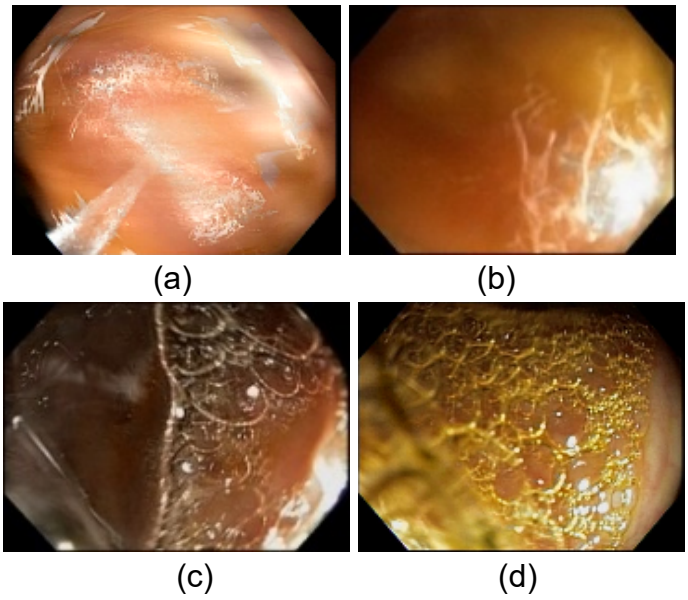


Figure 3.3 Examples of Water/Bubble frames: (a) and (b) Water Frames, (c) and (d) Bubble Frames. Even though colon mucosa is not visible in these images, they have significant amount of edges which result in incorrect classification as clear frames by existing IFF algorithms.

In this chapter, we propose a novel method for water and bubble frame detection based on image texture focusing on accumulation of pixel value differences. We compare it with other existing texture based algorithms in terms of accuracy and execution time. To further reduce the execution time, we investigate different clustering methods. The

proposed method performs very well in terms of accuracy and execution speed with or without clustering. More detailed explanation of accuracy and execution speed of the method is described in the experimental section. Therefore, our main contribution is to propose a novel method which can detect water and bubble frames with very high accuracy in significantly less processing time by using efficient clustering mechanism.

The remainder of this chapter is organized as follows. Related work is presented in Section 3.2. The proposed technique is described in Section 3.4. In Section 3.5, we discuss our experimental setup and results. Finally, Section 3.6 presents some concluding remarks.

3.2 Related Work

To the best of our knowledge, water and bubble frame detection in colonoscopy videos has not been investigated before. The most closely related work is [36] but it has some limitations as mentioned before. Recently, a new non-informative frame filtering method based on difference of Gaussian filtering has been proposed [39] but it has similar limitation which is that very clear water and bubble frames can be classified as informative. The clustering of non-informative frames in GI endoscopy videos is proposed for manifold learning to create structured manifolds from complex endoscopic videos [40]. Color and texture based features (mean, standard deviation, entropy, etc.) are extracted to classify the colon status as either normal or abnormal using Principle Component Analysis to reduce the size of features [41].

One of most popular texture detection method in images are based on textons. Textons are first introduced by Julesz [42] more than 30 years ago. In [43] algorithms are

designed to partition the grayscale image into different segments based on brightness and texture. Besides, there exist several texture detection techniques. The most commonly used ones are: Higher Order Local Auto Correlations (HLAC) [12], LBP (Local Binary Patterns) [4], Gabor filter banks [15], Leung-Malik filter banks [16], the traditional texture features (i.e., Contrast, Correlation, Energy, Homogeneity, etc.) based on Gray-Level Co-Occurrence Matrix (GLCM) [18], MPEG-7 texture features [19], Gaussian Markov random field (GMRF) [44] as well as Discrete Fourier Transform (DFT) [45].

Bejakovic et al [46] uses MPEG-7 descriptor along with GLCM features to differentiate fluids such as blood and intestinal juices as well as extraneous matter such as food and bubbles in WCE. Vilarino et al [47] proposed technique uses Gabor filters to automatically detect the intestinal juice. But as the experimental results show that Gabor filters are very computation expensive. Because of this, it cannot be used in real-time systems. All of these methods are competitive in terms of accuracy but their execution speeds vary a lot. We present the evaluation method and results of most of these existing algorithms. Also, we will compare our proposed method and these existing methods in Section 3.5 in terms of both ‘with clustering’ and ‘without clustering’ method.

3.3 Preprocessing

The goal of preprocessing stage is to filter out unnecessary blocks which may result in inconsistent results. Also, it normalizes the good blocks that are being processed so that the feature vector is consistent for all the variations of the blocks. The preprocessing steps are similar as explained in section 2.3.1. There are some notable differences and the major one is the specular reflection. Specular reflection are widely

present in water and bubble frames and is one of the key texture to differentiate them with other normal frames. Because of this reason we do not filter out specular reflection blocks from the preprocessing step. They are treated as good blocks and are moved to further processing. The black border blocks and blocks with high standard deviation are discarded based on the threshold values given in the Table 2.1.

3.4 Feature Extraction Methods

We compare our proposed feature extraction method explained in section 2.4 with several existing feature methods discussed in section 2.6.1. Most of the feature methods from section 2.6.1 are adapted. From our proposed method, we are using DIFF_1_10, DIFF_1_50, DIFF_2_10, DIFF_8_10, and DIFF_16_10. For the proposed feature methods, we are not considering DIFF_16_50 which results in 800 bin feature vector. From our experiments in chapter 2, we concluded that this feature method take a lot of computation time and it does not improve the accuracy at all.

In addition to these, we are considering a new variant called DIFF_2_10 having 20 size feature vector. This smaller size feature vector is computationally efficient and does not sacrifice the accuracy. The reason it did not work well for UC severity is that it could not generate the distinguishing feature vectors which could distinguish between five different UC classes effectively. This is because of the reason we discussed in chapter 2 that too much quantization may deteriorate the performance of feature method depending upon the texture patterns. From the existing feature methods from chapter 2, we are adapting LBP10, LBP59, LBP256, LOCAL_VAR, GLCM, HLAC, GABOR, LM, MPEG7_HTD, MPEG7_EHD. We are not considering MOD_LBP because of its high

computation cost and relatively low accuracy for UC images. We are also considering a new feature method based on Discrete Fourier Transform which will be discussed next.

3.4.1 Discrete Fourier Transform

For additional existing features, we have explored Discrete Fourier Transform (DFT) [45] based feature. DFT is used in different field of image processing such as image compression, image filtering and texture analysis. First, we get DFT of the input block using Fast Fourier Transform (FFT) algorithm [45]. FFT is a fast computation algorithm for DFT. To reduce the feature vector size, we take the mean and standard deviation of each row of the resultant block, and use them as features. In this way, a block is represented by a 256 bin feature vector with 128 means and 128 standard deviations.

3.5 Evaluation Method

Evaluation method has mainly two phases: Training and Testing. For Training, each input image is divided into a number of blocks, and the block filtering and normalization are applied as discussed in preprocessing section. A selected feature is computed for all blocks, and it is used to train a KNN (k-nearest neighbors) classifier [26] with k=1. We experimented with different values of k, but found k=1 giving best results for our dataset. We also tested other classifiers such as CART (Classification and Regression Tree) Decision Tree and SVM (Support Vector Machine) with linear kernel in MATLAB. Their comparison results will be discussed later.

For Testing, a test image is divided into number of blocks with the same block size used in Training. The same block filtering and normalization as used in Training are applied to all blocks in the test image. Using the trained KNN classifier, we determine for

each block to which type it belongs. Lastly, we calculate the probability of each type i.e. water/bubble or normal by dividing the detected number of blocks for each type by the total number of blocks processed for that image. If the test image has at least 50% water/bubble blocks, it is classified as a water/bubble frame. Otherwise, it is classified as normal frame. The rationale behind 50% threshold is that the colon mucosa is mostly hidden in an image covered with water/bubble by more than half. These types of frames will negatively affect the automatic feedback system if they are not filtered out.

3.6 Experiments

All experiments were conducted on a Windows 7 64-bit PC with Intel i7 2.8GHZ processor and 6GB RAM using MATLAB R2014a. The training images were provided by domain experts with annotations. To select testing images for water/bubble and normal class, we gathered several colonoscopy videos, extracted frames from them and randomly picked the test images. We tried our best to collect the images with different illuminations, colors, and noise levels so that it represents the vast majority of colonoscopy images and videos including the ones taken from different endoscopes.

We present our results using commonly used performance metrics [48]: Recall (or Sensitivity) (R), Specificity (S), Precision (P), and Accuracy (A). They are based on Table 3.1 of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). Precision (P) computed as the ratio of correctly classified positive instances from the predicted positives.

$$Precision(P) = \frac{TP}{TP+FP} \qquad (3.1)$$

Recall (or Sensitivity) (R) computed as the ratio of correctly classified positive instances.

$$Recall(R) = \frac{TP}{TP+FN} \quad (3.2)$$

Specificity (S) computed as the ratio of correctly classified negative instances.

$$Specificity(S) = \frac{TN}{TN+FP} \quad (3.3)$$

The accuracy (A) is the ratio of correctly classified instances.

$$Accuracy(A) = \frac{TP+TN}{TP+FP+FN+TN} \quad (3.4)$$

Table 3.1 Evaluation Metrics.

Actual	Predicted	
	Water/Bubble	Normal
Water/Bubble	TP	FN
Normal	FP	TN

The number of images and blocks used for training and testing are summarized in Table 3.2. We evaluate a large number of training and testing images in order to properly evaluate the computation cost of the different existing texture feature extraction methods and compare with different variations of our proposed texture feature extraction method.

Table 3.2 Description of number of images and blocks used in the experiments. The images are annotated by domain experts. The blocks used are only the good blocks after filtering out unnecessary blocks in the preprocessing stage.

Type	Training		Testing	
	Image	Block	Image	Block
Water + Bubble	588	22,049	288	10,522
Normal	599	21,296	284	10,456
Total	1,187	43,345	572	20,978

3.6.1 Evaluation Without Clustering

We evaluated a total of 15 features including 5 different versions of our DIFF features (DIFF_1_10, DIFF_1_50, DIFF_2_10, DIFF_8_10, and DIFF_16_10). Table 3.3 shows the results in terms of precision, recall, specificity, and accuracy for both image and block levels. Most feature methods are providing decent (i.e., 90-95%) image level accuracies. Some of the feature method such as GLCM and MPEG7_EHD did not perform well as compared to others.

Table 3.3 Image and Block level performance metrics without clustering (unit %)

Feature Method	Image				Block			
	Precision	Recall	Specificity	Accuracy	Precision	Recall	Specificity	Accuracy
DIFF_1_10	91.1	88.5	91.2	89.8	69.8	67.9	70.5	69.2
DIFF_1_50	90.9	89.9	90.8	90.3	70.0	67.9	70.8	69.1
DIFF_2_10	95.8	90.2	96.1	93.1	76.3	70.1	78.1	74.1
DIFF_8_10	96.0	91.6	96.1	93.8	77.0	72.0	78.4	75.2
DIFF_16_10	95.4	92.7	95.4	94.0	75.7	72.0	76.8	74.4
LBP10	88.8	91.3	88.3	89.8	72.8	72.7	72.7	72.7
LBP59	93.1	93.7	92.9	93.3	76.4	75.4	76.5	75.9
HLAC	91.6	94.4	91.2	92.8	69.9	72.9	68.4	70.7
GLCM	86.3	85.4	86.2	85.8	63.6	62.8	63.9	63.3
LOCAL_VAR	91.1	82.2	91.9	87.0	69.2	65.0	70.8	67.9
GABOR	94.1	93.7	94.0	93.5	74.0	74.7	73.6	74.2
LM	93.4	93.7	93.3	93.5	73.4	74.2	73.0	73.6
DFT	95.9	90.2	96.1	93.1	70.5	67.2	71.7	69.5
MPEG7_HTD	96.7	93.0	96.8	94.9	77.8	74.6	78.6	76.6
MPEG7_EHD	77.8	92.7	73.2	83.0	67.4	78.4	61.8	70.1

Our feature method DIFF_16_10 performed as good as the best performing feature method which is MPEG7_HTD. One of the reason GLCM did not perform well could be because of its smaller feature vector size. We observed that almost all of our feature methods (DIFF_1_50, DIFF_2_10, DIFF_8_10, and DIFF_16_10) performed on par with popular existing methods. Although all 4 performance metrics are equally important, we are mainly focused on the accuracy metric which gives the overall performance (both positive and negative classification) of the feature method tested.

The block level accuracy were similar to image level accuracy. DIFF_2_10, DIFF_8_10 and DIFF_16_10 performed on par with other feature methods in block level tests. GLCM and DFT are the worst performer in terms of block level. It should be noted that the image level result is more important for the classification because doctors only consider image level evaluation. DFT feature method is highly accurate with 93.1% even though its block level accuracy is only 69.5%. We will discuss the performance of feature methods with clustering and without clustering later along with their computation costs later. The computation cost is the main differentiator among the different feature methods because their accuracy are very similar. As seen in figure 3.4 and 3.5 the performance metrics are not that different when clustering is not used. The similar results hold for both image and block level evaluations. We will see how the performance fluctuates when clustering is used in the next sub section.

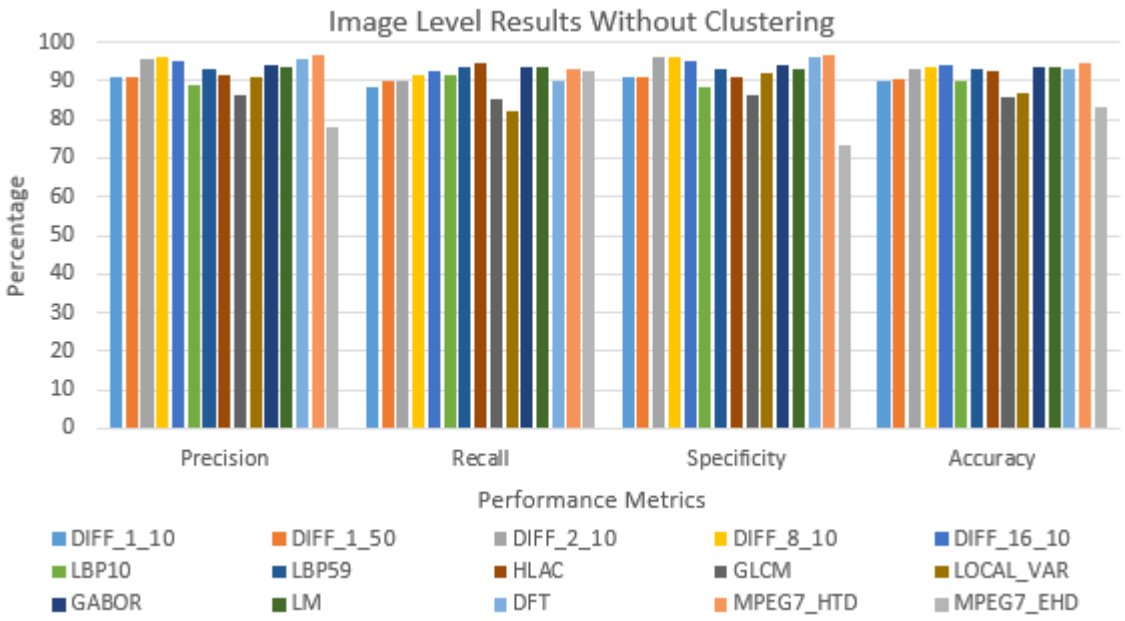


Figure 3.4 Image Level performance metrics without clustering

Figure 3.5 Block level performance metrics without clustering

3.6.2 Evaluation with Clustering

For more efficient and faster computing, we consider the clustering of the training blocks. To provide accurate detection of water and bubble frames, a huge number of

blocks in the training set need to be compared with the blocks in an unseen image. By the use of clustering, we can reduce the number of comparisons, which impacts the execution speed. A cluster has hundreds of feature vectors generated from hundreds of blocks. Instead of comparing with these hundreds of vectors, we can compare with one vector which is its centroid (i.e., mean). For the clustering purpose, we use K-means, K-medoids and Fuzzy C-means clustering [26]. But, first we need to find an optimal number of clusters.

We use the Elbow method which is simple but effective [49] where Within Cluster Sum of Squares (WCSS) is observed for different number of clusters. We ran K-means clustering for k = 10, 20, 30, …, K_{max}, where K_{max} equals 500 in our case, and the WCSS value is computed for each k. Our goal is to find the minimum value of k without sacrificing the accuracy of the classification.

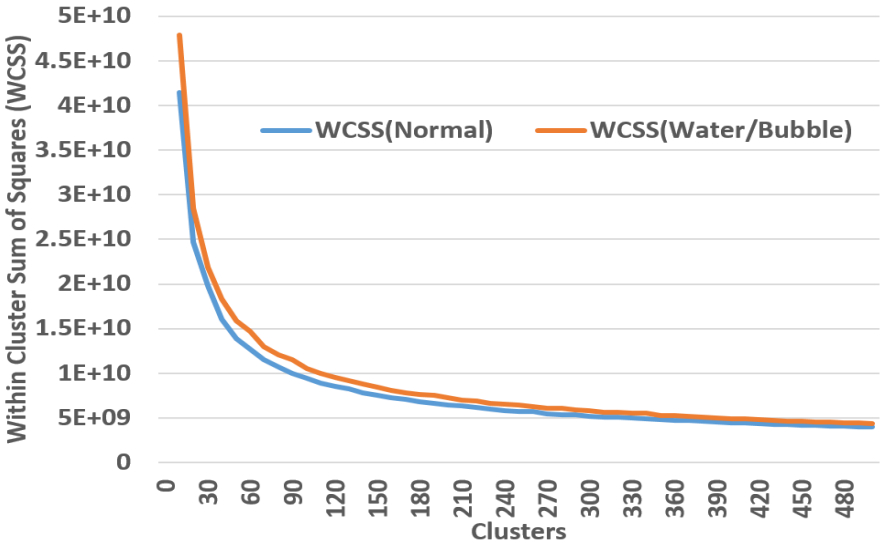


Figure 3.6 Optimal cluster estimation using Within Cluster Sum of Squares (WCSS). The plots almost overlap which means that we can use same number of clusters for both water/bubble and normal images.

We plot clusters (k) versus WCSS values. The optimal number of clusters is estimated by looking for k for which WCSS is not decreasing rapidly. Figure 3.6 shows the plot for water/bubble blocks as well as normal blocks in which DIFF_2_10 feature is used for computing WCSS values. The plot was obtained based on all of the training blocks for both water/bubble and normal images as listed in Table 3.3. As seen in the plot, after the k value around 50, the WCSS values do not decrease rapidly. And, after the k value around 300, the WCSS values change very slowly, which makes the graph almost flat. So, we can see that an optimal k value can be in the range from 50 to 300. Next, we find the optimal k from this range.

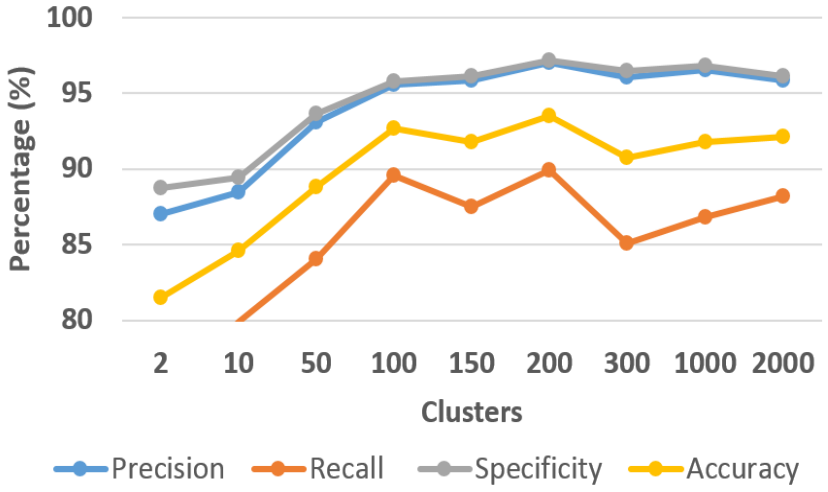


Figure 3.7 Image level accuracy for different numbers of clusters using DIFF_2_10 feature method. We limit the maximum number to clusters to 2000.

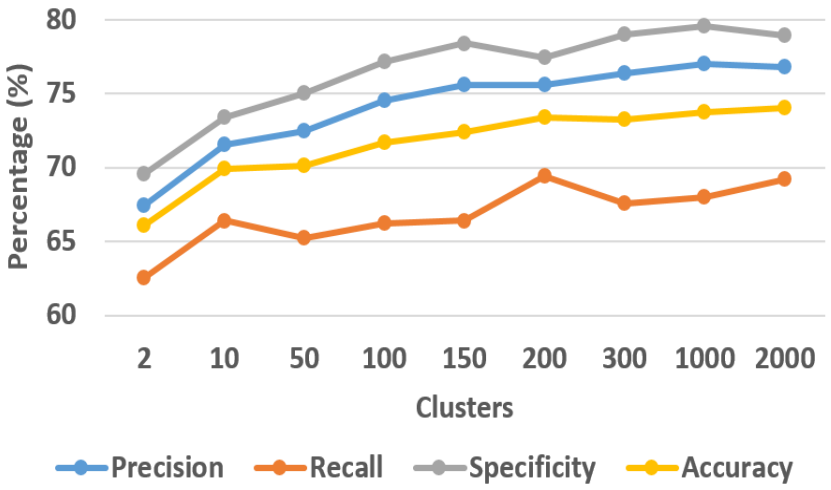


Figure 3.8 Block level accuracy for different numbers of clusters using DIFF_2_10 feature method. The maximum number of clusters was set to 2000 for block level test as well.

We evaluated several different k values. Figure 3.7 and 3.8 show the image and block level accuracies when K-means clustering is used with different cluster sizes and with DIFF_2_10 as feature method. It can be seen that the optimal image and block level accuracies are achieved at the cluster size of around 200 which falls in the estimated range by WCSS plot.

Table 3.4 show the results in terms of precision, recall, specificity, and accuracy for both image and block levels with clustering of size 200. As seen, the performances are degraded for the most of the features when compared with those without clustering. DIFF_1_10, DIFF_2_10, DIFF_8_10, DIFF_16_10, LBP59, and LM are still good (i.e., better than 90%). It can be seen that DIFF_2_10 retained its 93% accuracy even after the clustering. Accuracy was expected to go down when clustering was used to save the evaluation time. The previous best MPEG7_HTD decreased it accuracy from 94.9% to 86.7% which is more than 8 percentage points. It shows that the clustering is sensitive to feature methods. Our DIFF based feature is robust to clustering as shown in

the results that the accuracy remained intact even after clustering. We will show later that it is a key feature of our proposed feature method which dramatically improves the processing time without sacrificing the accuracy. We claim that DIFF_2_10 is our best choice since it is faster than the others and as well as it retains accuracy after clustering. Figure 3.9 and 3.10 shows the image and block level performance metrics with clustering. It can be seen that accuracy decreased for several feature methods significantly.

Table 3.4 Image and Block level performance metrics with clustering (unit %)

Feature Method	Image				Block			
	Precision	Recall	Specificity	Accuracy	Precision	Recall	Specificity	Accuracy
DIFF_1_10	90.7	85.4	91.2	88.2	70.7	66.6	72.2	69.4
DIFF_1_50	88.8	88.8	88.7	88.8	70.1	66.6	71.4	69.0
DIFF_2_10	97.0	89.9	97.1	**93.5**	75.5	69.4	77.4	73.4
DIFF_8_10	96.1	87.1	96.4	91.7	75.9	68.1	78.2	73.1
DIFF_16_10	95.0	87.5	95.4	91.4	74.7	69.4	76.3	72.9
LBP10	78.2	90.9	74.3	82.6	72.5	81.5	68.9	75.2
LBP59	90.5	89.5	90.4	90.0	79.2	76.1	79.9	78.0
HLAC	87.6	84.0	88.0	86.0	66.5	65.7	66.7	66.2
GLCM	78.9	87.1	76.4	81.8	62.3	63.2	61.5	62.4
LOCAL_VAR	89.2	77.4	90.4	83.9	70.5	65.2	72.5	68.9
GABOR	97.7	45.1	98.9	71.8	87.6	47.5	93.2	70.3
LM	95.0	93.7	95.0	94.4	71.7	74.4	70.5	72.4
DFT	96.5	78.1	97.1	87.5	74.3	61.5	78.6	70.0
MPEG7_HTD	81.5	95.1	78.1	86.7	67.2	75.4	62.9	69.2
MPEG7_EHD	66.1	95.8	50.3	73.2	60.5	74.2	51.3	62.8

Figure 3.9 Image Level Performance Metrics with Clustering

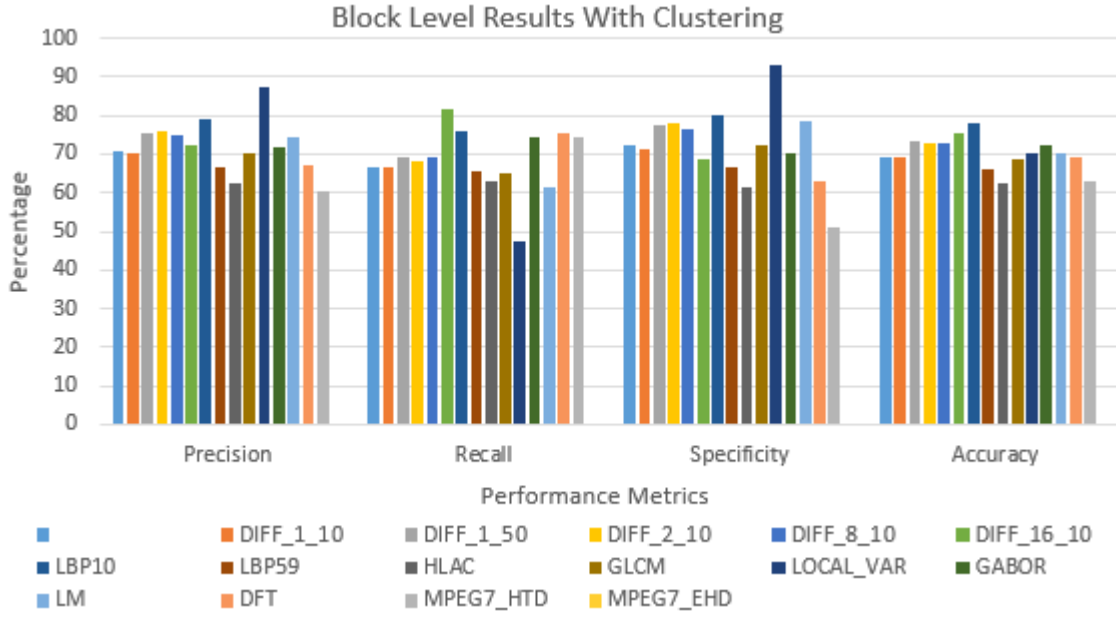


Figure 3.10 Block Level Performance Metrics with Clustering

We also evaluated our best performing feature DIFF_2_10 using different clustering algorithms and classifiers. We set the number of cluster to 200 as before and cluster our training blocks using K-medoids and Fuzzy C-Means as well as K-means

clustering algorithms [26]. For the classification we chose SVM and CART Decision Tree to compare with KNN [26].

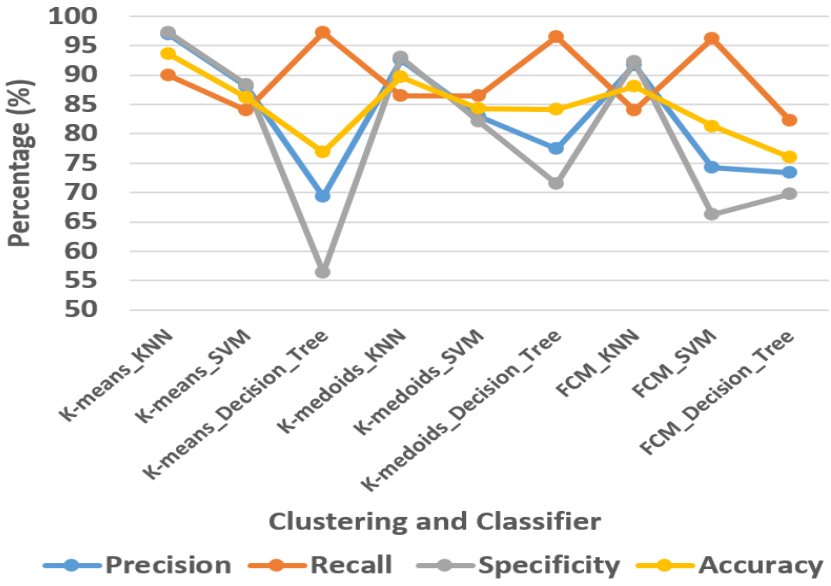


Figure 3.11 DIFF_2_10 Image level accuracies with three different clustering algorithms (K-means, K-medoids, and Fuzzy C-means) and three classifiers (KNN, SVM, and Decision Tree).

Figure 3.11 and 3.12 show the results of a total of nine combinations of the three clustering algorithms and three classifiers for the image and block levels, respectively. The main objective of this evaluation to observe the change in the accuracy when different clustering algorithms and classifiers are used. As seen, KNN with K-means clustering is the best among all in terms of accuracy. We observed that k-means and decision tree combination gives best recall percentage but it is not as good in terms of other metrics.

Figure 3.12 DIFF_2_10 block level accuracies with same three different clustering algorithms and same three classifiers.

3.7 Execution Speed Comparison

Computation cost is really important in a colonoscopy video processing system since a very large number of frames need to be evaluated. We compare the computation costs of some of the better performing features. Tables 3.5 and 3.6 show the results of the total computation costs for entire images of training and testing listed in Table 3.2 for both 'without clustering' and 'with clustering'. As seen, our DIFF based features are more than 2x faster than the others for the training phase in both 'with' and 'without' clustering evaluations. For the testing phase, our best performing feature DIFF_2_10 is significantly faster than all other similarly performing features. For example – per frame testing cost for DIFF_2_10 is 746.9/572 (these numbers are from Tables 3.5 and 3.2) = 1.3 seconds.

Table 3.5 Execution speed without clustering (unit: seconds). Not all feature methods are considered for execution speed comparison.

Features	Training	Testing	Total
DIFF_1_10	251.4	668.8	920.2
DIFF_2_10	278.6	746.9	1,025.5
DIFF_8_10	278.0	1,101.1	1,379.1
DIFF_16_10	283.1	1,562.9	1,846.0
LBP10	965.9	1,061.3	2,027.2
LBP59	857.3	1,251.6	2,108.9
HLAC	1,278.9	1,272.2	2,551.1
DFT	677.2	2,311.9	2,989.1
MPEG7_HTD	1,842.1	2,061.3	3,903.4
LM	8,565.0	4,941.6	13,506.6
GABOR	16,713.2	13,407.6	30,120.8

Table 3.6 Execution speed with clustering. Some of the feature methods dramatically improved the execution speed.

Features	Clustering	Training	Testing	Total
DIFF_1_10	775.6	244.0	624.5	1,644.1
DIFF_2_10	794.7	271.5	642.4	1,708.6
DIFF_8_10	1,167.0	327.3	674.5	2,168.8
DIFF_16_10	2,513.5	370.5	675.9	3,559.9
DFT	4,861.3	749.8	846.4	6,457.5
LBP10	1,254.2	916.3	959.9	3,130.4
LBP59	1,287.5	827.9	979.6	3,095.0
HLAC	1,763.7	1,281.2	1,172.4	4,215.3
MPEG7_HTD	4,003.2	2,109.9	1,605.0	7,718.1
LM	9,477.5	8,825.1	3,727.2	22,029.8
GABOR	18,408.8	17,142.5	8,532.2	44,083.5

Since all the implementations are done in MATLAB, the cost can be reduced significantly once implemented in C/C++. As mentioned earlier, the main benefit of clustering is to reduce the number of comparisons in the testing phase thereby reducing the computation cost. We observed that the computation cost improves dramatically in the testing phase for feature methods with a larger feature vector size like DIFF_16_10

(160 bin) and DFT (256 bin) as seen in Figure 3.14. For example – the testing time of DIFF_16_10 is reduced more than 2 times with clustering. Even without considering the one-time cost like clustering and training, classification using our DIFF based features is significantly faster than that using the other feature extraction methods. This shows that our DIFF based features are computationally efficient without sacrificing accuracy.

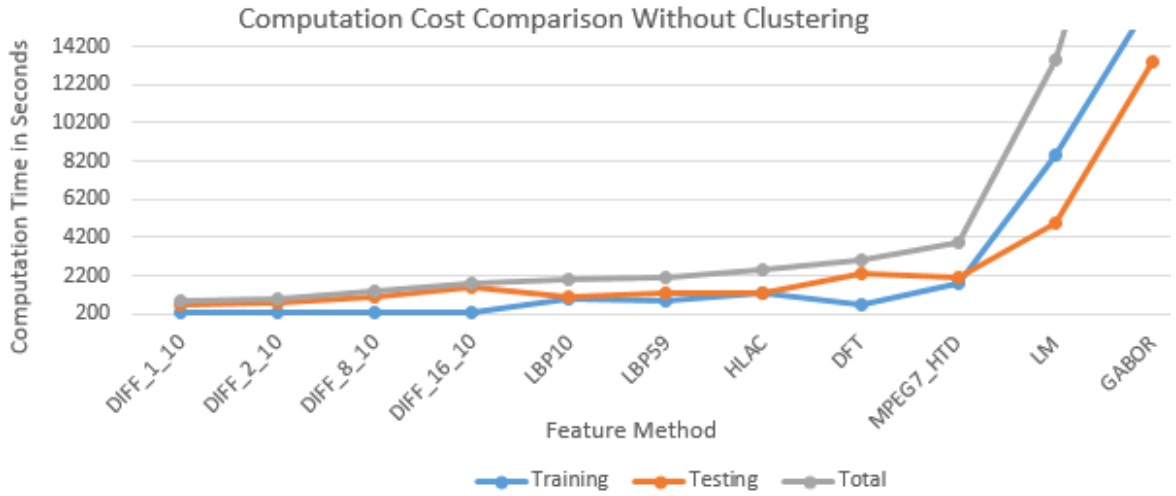


Figure 3.13 Plot of computation cost without clustering

Figure 3.13 shows the plot of computation cost without clustering. The most important time measure is during the testing phase because the training is only a one-time cost. All of our DIFF methods have lower processing time than other existing methods. As seen in figure 3.14 the computation cost decreased dramatically for some of the feature methods because during the classification we are comparing less number of blocks because of the use of clustering.

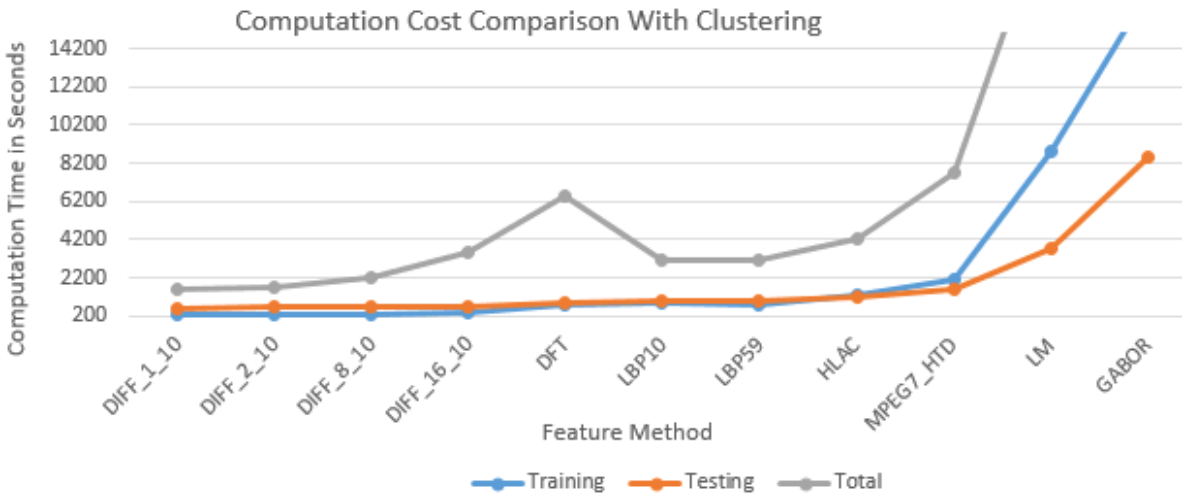


Figure 3.14 Plot of computation cost with clustering

3.8 Conclusion

To improve quality of colonoscopy, research have been conducted investigating an 'automated feedback system' which informs the endoscopist of possible sub-optimal inspection during the procedure. One of the basic steps of this system is to distinguish non-informative frames from informative ones. Existing methods for this cannot classify water/bubble frames (which do not carry any useful visual information of colon mucosa) as non-informative frames since they focus on image clarity not image semantic. To consider image semantic, we propose a novel image texture feature based on accumulation of pixel differences, which can detect water and bubble frames with very high accuracy and significantly less processing time. To reduce processing time even more, we employ clustering which can reduce the number of time-consuming comparisons. The experimental results show the proposed feature can achieve more than 93% overall accuracy in almost half of the time existing methods take.

CHAPTER 4: OVERALL CONCLUSION AND FUTURE WORK

There are several types of disorders that affects our colon's ability to function properly such as Colorectal Cancer, Ulcerative Colitis, Diverticulitis, Irritable Bowel Syndrome, Colonic polyps and other abnormalities. As discussed before, the automated procedure quality measurement system can provide Colonic polyp detection only at the moment among these disorders. We would like to add a functionality to handle one of the important disorders called Ulcerative Colitis. However, it is very difficult to evaluate the severity of Ulcerative colitis (UC) objectively because of non-uniform nature of symptoms associated with UC, and large variations in their patterns.

To address this, we objectively measure and classify the severity of UC presented in optical colonoscopy video frames based on image textures. To extract distinct textures, we use a hybrid approach in which a new proposed method (DIFF) based on the accumulation of pixel value differences is combined with an existing method such as LBP. Therefore, our contributions are development of a new texture feature that works well for 'severe' and 'moderate' UC classes, and to combine this new texture feature with an existing feature to achieve significantly better overall accuracy with significantly less processing time for all UC disease grade classes. The experimental results show that the hybrid method, which can easily be modified for further improvement, already can achieve more than 90% overall accuracy.

Recent data suggests that there is a significant (4-12%) miss-rate for the detection of even large polyps and cancers during the colonoscopy procedure. To improve quality of colonoscopy, an 'automated feedback system' which informs the endoscopist of possible sub-optimal inspection during the procedure have been investigated. One of the

basic steps of this system is to distinguish non-informative frames from informative ones. Existing methods for this cannot classify water/bubble frames (which do not carry any useful visual information of mucosa) as non-informative frames since they focus on image clarity not image semantic. To consider image semantic, we propose a novel image texture feature based on accumulation of pixel differences, which can detect water and bubble frames with very high accuracy and significantly less processing time. To reduce processing time even more, we employ clustering which can reduce the number of time-consuming comparisons. The experimental results show the proposed feature can achieve more than 93% overall accuracy in almost half of the time existing methods take.

The results also show that our DIFF based methods are hardly affected by the clustering. This is beneficial if we have very large dataset that needs to be trained and computation cost is important. Our DIFF based feature with clustering can be extended for any computationally intensive real-time systems. Depending upon the nature of the textures, different variants of DIFF feature can be used with different size of clusters. Last but not the least, the proposed feature method can be applied to any image domains as our results illustrated that it works accurately for other types of images as well.

We plan to extent this image classification into video based severity score calculation and shot segmentation. The severity score calculation and shot segmentation can help doctors during real-time colonoscopy procedure as well as post procedure assessments. Based on the colonoscopy videos, the severity score will help to measure the mucosa healing progress in the patients. Severity score calculation can be performed for whole colonoscopy video in certain frame per second rate. Various preprocessing steps can be applied to discard water, bubble, blurry, and stool frames from colonoscopy

video using our existing algorithms. Just like image based classification, each image from the video should be divided into number of blocks and feature is extracted from each block using proposed single feature or hybrid method involving DIFF and other existing methods. K-means clustering can be used for training blocks to reduce the ultimate classification cost and evaluation can be done based on KNN classifier. The severity score of each class (‘severe’, ‘moderate’, ‘mild’, and ‘scar’) as well as other metrics can be generated from each video. Shot detection algorithm can be implemented to segment different shots of the video based on the detected image severity. The optimized C/C++ code has be written to further improve the computation cost and meet the real-time requirements.

Based on our extensive experiments and exposure to hundreds of colonoscopy videos, we have realized that there are different variety of UC images having different textures, colors, orientations, contrasts and illuminations. So the traditional machine learning method is not feasible for large number of videos as training should be rebuilt for every new set of images. The current deep learning [51] trend seems very promising and the results are encouraging especially for image classifications [52]. To improve our detection system even further and to keep the system efficient, deep learning method can be applied to detect the UC images as well as water bubble images. Especially, the effectiveness of UC classification can be improved by using deep learning. We will explore deep learning as an extension of our current work.

REFERENCES

[1] *American Cancer Society, Colorectal Cancer Facts and Figures, 2015*.

[2] C. D. Johnson, J. G. Fletcher, R. L. MacCarty, J. N. Mandrekar, W. S. Harmsen, P. J. Limburg*, et al.*, "Effect of Slice Thickness and Primary 2D Versus 3D Virtual Dissection on Colorectal Lesion Detection at CT Colonography in 452 Asymptomatic Adults," *American Journal of Roentgenology,* vol. 189, pp. 672-680, 2007.

[3] S. R. Stanek, W. Tavanapong, J. Wong, J. Oh, R. D. Nawarathna, J. Muthukudage*, et al.*, "SAPPHIRE: A toolkit for building efficient stream programs for medical video analysis," *Computer Methods and Programs in Biomedicine,* vol. 112, pp. 407-421, 2013.

[4] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 971-987, 2002.

[5] P. Rutgeerts, W. J. Sandborn, B. G. Feagan, W. Reinisch, A. Olson, J. Johanns*, et al.*, "Infliximab for Induction and Maintenance Therapy for Ulcerative Colitis," *New England Journal of Medicine,* vol. 353, pp. 2462-2476, 2005.

[6] S. Lichtiger, D. H. Present, A. Kornbluth, I. Gelernt, J. Bauer, G. Galler*, et al.*, "Cyclosporine in Severe Ulcerative Colitis Refractory to Steroid Therapy," *New England Journal of Medicine,* vol. 330, pp. 1841-1845, 1994.

[7] D. Turner, C. H. Seow, G. R. Greenberg, A. M. Griffiths, M. S. Silverberg, and A. H. Steinhart, "A systematic prospective comparison of noninvasive disease

activity indices in ulcerative colitis," *Clin Gastroenterol Hepatol,* vol. 7, pp. 1081-8, Oct 2009.

[8] F. Carbonnel, A. Lavergne, M. Lemann, A. Bitoun, P. Valleur, P. Hautefeuille*, et al.*, "Colonoscopy of acute colitis. A safe and reliable tool for assessment of severity," *Dig Dis Sci,* vol. 39, pp. 1550-1557, July 1994.

[9] S. Ardizzone, A. Cassinotti, P. Duca, C. Mazzali, C. Penati, G. Manes*, et al.*, "Mucosal healing predicts late outcomes after the first course of corticosteroids for newly diagnosed ulcerative colitis," *Clin Gastroenterol Hepatol,* vol. 9, pp. 483-489.e3, Jun 2011.

[10] H. Nosato, H. Sakanashi, E. Takahashi, and M. Murakawa, "An objective evaluation method of ulcerative colitis with optical colonoscopy images based on higher order local auto-correlation features," *IEEE 11th International Symposium on Biomedical Imaging (ISBI),* pp. 89-92, May 2014.

[11] G. D'Haens, W. J. Sandborn, B. G. Feagan, K. Geboes, S. B. Hanauer, E. J. Irvine*, et al.*, "A review of activity indices and efficacy end points for clinical trials of medical therapy in adults with ulcerative colitis," *Gastroenterology,* vol. 132, pp. 763-86, Feb 2007.

[12] T. Kurita, N. Otsu, and T. Sato, "A face recognition method using higher order local autocorrelation and multivariate analysis," *Proceedings of 11th IAPR International Conference on Pattern Recognition Methodology and Systems,* pp. 213-216, Sep 1992.

[13] T. Ojala, K. Valkealahti, E. Oja, and M. Pietikäinen, "Texture discrimination with multidimensional distributions of signed gray-level differences," *Pattern Recognition,* vol. 34, pp. 727-739, 2001.

[14] M. Pietikäinen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," *Pattern Recognition,* vol. 33, pp. 43-52, 2000.

[15] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Identification Using Encrypted Biometrics," *Computer Analysis of Images and Patterns,* vol. 8048, pp. 440-448, Jan 2013.

[16] T. Leung and J. Malik, "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons," *International Journal of Computer Vision,* vol. 43, pp. 29-44, 2001/06/01 2001.

[17] A. Varghese, K. Balakrishnan, R. R. Varghese, and J. S. Paul, "Content Based Image Retrieval of Brain MR Images across Different Classes," *World Academy of Science, Engineering and Technology,* vol. 7, pp. 465-469, 2013.

[18] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural Features for Image Classification," *IEEE Transactions on Systems, Man and Cybernetics,* vol. SMC-3, pp. 610-621, 1973.

[19] M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy, "BilVideo-7: An MPEG-7-Compatible Video Indexing and Retrieval System," *IEEE MultiMedia,* vol. 17, pp. 62-73, 2010.

[20] B. Li and M. Q. H. Meng, "Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments," *Computers in Biology and Medicine,* vol. 39, pp. 141-147, 2009.

[21] Y. Lecheng, P. C. Yuen, and L. Jianhuang, "Ulcer detection in wireless capsule endoscopy images," *21st International Conference on Pattern Recognition (ICPR),* pp. 45-48, Nov 2012.

[22] L. Baopu, Q. Lin, M. Q. H. Meng, and F. Yichen, "Using ensemble classifier for small bowel ulcer detection in wireless capsule endoscopy images," *IEEE International Conference on Robotics and Biomimetics (ROBIO),* pp. 2326-2331, Dec 2009.

[23] W. Zhuoshi, Z. Weidong, L. Jianfei, W. Shijun, Y. Jianhua, and R. M. Summers, "Computer-aided detection of colitis on computed tomography using a visual codebook," *IEEE 10th International Symposium on Biomedical Imaging (ISBI),* pp. 141-144, April 2013.

[24] R. Nawarathna, J. Oh, J. Muthukudage, W. Tavanapong, J. Wong, P. C. d. Groen*, et al.*, "Abnormal image detection in endoscopy videos using a filter bank and local binary patterns," *Neurocomputing,* vol. 144, pp. 70-91, November 2014.

[25] H. G. Adelmann, "Butterworth equations for homomorphic filtering of images," *Computers in Biology and Medicine,* vol. 28, pp. 169-181, 6/1/ 1998.

[26] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*: Wiley-Interscience, 2000.

[27] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological Cybernetics,* vol. 61, pp. 103-113, 1989/06/01 1989.

[28] M. Kreutz, B. Völpel, and H. Janßen, "Scale-invariant image recognition based on higher-order autocorrelation features," *Pattern Recognition,* vol. 29, pp. 19-26, 1996.

[29] T. Kurita and S. Hayamizu, "Gesture recognition using HLAC features of PARCOR images and HMM based recognizer," *Third IEEE International Conference on Automatic Face and Gesture Recognition,* pp. 422-427, Apr 1998.

[30] T. Toyoda and O. Hasegawa, "Extension of higher order local autocorrelation features," *Pattern Recognition,* vol. 40, pp. 1466-1473, 2007.

[31] A. Rampun, H. Strange, and R. Zwiggelaar, "Texture segmentation using different orientations of GLCM features," presented at the Proceedings of the 6th International Conference on Computer Vision / Computer Graphics Collaboration Techniques and Applications, Berlin, Germany, 2013.

[32] K. J. Dana, B. V. Ginneken, S. K. Nayar, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *ACM Trans. Graph.,* vol. 18, pp. 1-34, 1999.

[33] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1265-1278, 2005.

[34] A. Dahal, J. Oh, W. Tavanapong, J. Wong, and P. C. de Groen, "Detection of ulcerative colitis severity in colonoscopy video frames," *13th International Workshop on Content-Based Multimedia Indexing (CBMI),* pp. 1-6, June 2015.

[35] J. Oh, S. Hwang, Y. Cao, W. Tavanapong, D. Liu, J. Wong*, et al.*, "Measuring Objective Quality of Colonoscopy," *IEEE Transactions on Biomedical Engineering,* vol. 56, pp. 2190-2196, 2009.

[36] V. P. Karri, J. Oh, W. Tavanapong, J. Wong, and P. C. d. Groen, "Effective and Accelerated Informative Frame Filtering in Colonoscopy Videos using Graphics

Processing Unit," *International Conference on Bio-inspired Systems and Signal Processing,* pp. 119-124, January 2011.

[37] J. Oh, S. Hwang, W. Tavanapong, P. C. de Groen, and J. Wong, "Blurry-frame detection and shot segmentation in colonoscopy videos," 2003, pp. 531-542.

[38] S. H. JungHwan Oh, JeongKyu Lee, W. Tavanapong, Piet C. de Groen, and Johnny Wong, "Informative Frame Classification for Endoscopy Video," *Medical Image Analysis,* vol. 11, pp. 110-127, February 27 2007.

[39] B. Munzer, K. Schoeffmann, and L. Boszormenyi, "Relevance Segmentation of Laparoscopic Videos," *IEEE International Symposium on Multimedia (ISM),* pp. 84-91, Dec 2013.

[40] S. Atasoy, D. Mateus, J. Lallemand, A. Meining, G. Z. Yang, and N. Navab, "Endoscopic video manifolds," *Med Image Comput Comput Assist Interv,* vol. 13, pp. 437-45, 2010.

[41] M. P. Tjoa and S. M. Krishnan, "Feature extraction for the analysis of colon status from the endoscopic images," *BioMedical Engineering OnLine,* vol. 2, pp. 9-9, 2003.

[42] B. Julesz, "Textons, the elements of texture perception, and their interactions," *Nature,* vol. 290, pp. 91-97, 03/12/print 1981.

[43] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *International Journal of Computer Vision,* vol. 43, pp. 7-27, 2001/06/01 2001.

[44] R. Chellappa and S. Chatterjee, "Classification of textures using Markov random field models," *IEEE International Conference on ICASSP Acoustics, Speech, and Signal Processing,* vol. 9, pp. 694-697, Mar 1984.

[45] P. Duhamel and M. Vetterli, "Fast fourier transforms: A tutorial review and a state of the art," *Signal Processing,* vol. 19, pp. 259-299, 1990.

[46] S. Bejakovic, R. Kumar, T. Dassopoulos, G. Mullin, and G. Hager, "Analysis of Crohn's disease lesions in capsule endoscopy images," *IEEE International Conference on Robotics and Automation,* pp. 2793-2798, 12-17 May 2009.

[47] F. Vilarino, P. Spyridonos, O. Pujol, J. Vitria, P. Radeva, and F. de Iorio, "Automatic Detection of Intestinal Juices in Wireless Capsule Video Endoscopy," *18th International Conference on Pattern Recognition* vol. 4, pp. 719-722, 2006.

[48] J. Han and M. Kamber, in *Data Mining: Concepts and Techniques*, ed: Morgan Kaufmann Publishers, 2001.

[49] D. J. Ketchen and C. L. Shook, "The Application of Cluster Analysis in Strategic Management Research: An Analysis and Critique," *Strategic Management Journal,* vol. 17, pp. 441-458, 1996.

[50] A. Dahal, J. Oh, W. Tavanapong, J. Wong, and P. C. d. Groen, "Enhancing Informative Frame Filtering by Water and Bubble Detection in Colonoscopy Videos," *Proceedings of the 2015 International Conference on Health Informatics & Medical Systems,* pp. 24-30, July 27-30 2015.

[51] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks,* vol. 61, pp. 85-117, 2015.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems* pp. 1106-1114, 2012.