جـــامـعــة الـــشـرق الأوسـط
MIDDLE EAST UNIVERSITY
Amman - Jordan

Image Classification Based on Enhancement of Local Binary Pattern

تصنيف الصور استناداً إلى تحسين النمط الثنائي المحلي

Prepared By

Ahmad A. Suleiman

Supervisor

Dr. Mudhfar Al-Jarrah

Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

January, 2018

Authorization

I, Ahmad Abdelhamid Ekrayyem Sulieman, authorize the Middle East University to provide hard copies or soft copies of my Thesis to libraries, institutions, or individuals upon their request.

Name: Ahmad A. E. Sulieman

Date: 15/ 1/ 2018

Signature:

Examination Committee Decision

This is to certify that the thesis entitled **"Image Classification Based on Enhancement of Local Binary Pattern "**was successfully defended and approved on 15 / 1 / 2018.

Examination Committee Members	**Signature**
Dr. Mudhfar Al-Jarrah **(Supervisor)** Associate Professor, Department of Computer Science Middle East University (MEU)	
Dr. Abdelrahman Abu-Arqoub **(Member / Chairman)** Assistant Professor, Department of Computer Science Middle East University (MEU)	
Dr. Nijad Al-Najdawi **(External Examiner)** Associate Professor, Department of Computer Science Al-Balqa Applied University (BAU)	

Acknowledgment

First, I give thanks, praise to Allah for his mercy, and reconcile and for granting me knowledge, confidence, patience to pass this Master thesis successfully.

Also, I would like to express my thank to my thesis advisor. **Dr. Mudafar Al-Jarah** for the complete guidance throughout the thesis stages, and for the critical assistance in designing and proceeding the methodology of my research.

Also, I wish to place on record my deep and sincere gratitude to the previous advisor of my thesis **Dr. Ahmad Abu-Shareha**, for suggesting this field of research work and continue working with full enthusiasm. I gratefully acknowledge his kindness, patience, encouragement.

Finally, I thank all those, who have helped me directly or indirectly in the successful completion of my research work

Ahmad Suleiman

The Researcher

Dedication

To the one who always kept me in her prayers, the one who did not save any effort to assist me throughout my life, **My Beloved Mother**.

To **My Father**, who has been struggling to assure us a decent life, who taught me the acts of mannerism, who kept admonishing me by the trust and honesty.

To the wonderful **Sisters**.

To the gorgeous **Brother**.

To the light of my eyes, to the love of my life, to my heart, **My Daughter**.

I dedicate my effort

Ahmad Suleiman

Table of Content

Cover Page……………………..……………………………………………………….….I

Authorization.. II

Examination Committee Decision III

Acknowledgment..IVI

Dedication V

Table of Content...IVI

List of Tables X

List of Figures.. XI

List of Abbreviations XIV

Abstract...XVI

المُلخص.. XVIII

Chapter one: **Introduction** 1

1.1Introduction 2

1.2 Background of the study 3

1.3 Local Binary Pattern (LBP) 7

1.4 Problem Statement 9

1.5 Research Questions 10

1.6 Goal and Objectives 10

1.7 Motivation……………………………………...11

1.8 Contribution and Significance of the Research 11

1.9 Scope of the Study 11

1.10 Limitations of the Study 12

1.11 Thesis Outline 12

Chapter Two: Background and Literature Review 13

2.1Overview........ 14

2.2 Introduction........ 14

2.3 Background........ 14

2.3.1 Local Binary Pattern Mechanism 15

2.3.2 LBP Minimization........ 18

2.3.2.1 Uniforms Patterns 19

2.3.2.2 Rotation Invariant 20

2.4 Related Work 21

2.4.1 Increasing Discriminative Ability........ 22

2.4.2 Raising LBP Performance 27

2.4.3 Neighborhood Topology........ 33

2.4.4 Minimization LBP Size 37

2.5 LBP Variants Comparison........ 40

2.6 Classification Algorithms 43

2.6.1 Support Vector Machines 44

2.6.2 SVM Multi-class Classification........ 44

2.7 Summary........ 45

Chapter Three: Proposed Work **46**

3.1 Overview 47

3.2 Introduction 47

3.3 Diagonal Intersection LBP Descriptor 48

3.3.1 DILBP Neighborhood Topology 49

3.3.2 DILBP Codes Generation Process 50

3.3.3 Complexity Reduction of DILBP 52

3.3.4 DILBP Weights Assignment 53

3.3.5 Final Descriptor Construction 53

3.4 DILBP Pseudocode and Flowchart 54

3.5 Generated Patterns using DILBP 58

3.6 DILBP Characteristics 60

3.7 Summary 64

Chapter Four: Implementation and Results **65**

4.1 Overview 66

4.2 Introduction 66

4.3 Datasets 67

4.3.1 Fifteen Scene Dataset 67

4.3.2 Outex Dataset 68

4.3.3 CUReT Dataset 70

4.4 Implementation 71

4.5 Parameter Setting 72

4.6 Evaluation Metrics 73

4.7 Results 73

4.7.1 Results of Fifteen Scene Dataset 73

4.7.2 Results of Outex TC 12 Dataset 76

4.7.3 Results of CUReT Dataset 78

4.8 Summary 79

Chapter Five: Conclusion and Future Work 81

5.1 Conclusion 82

5.2 Future Work 84

References 85

List of Tables

Chapter Number. Table Number	Contents	Page
Table 2.1	Example of Uniform and Non-uniform LBP Patterns	20
Table 2.2	Comparison Between LBP Variations	40
Table 3.1	Converted DILBP Sub-Code to Upper and Lower Binary Code	52
Table 4.1	Parameter Settings of Fifteen Scene, Outex TC12 and CUReT Datasets	72
Table 4.2	Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes of Fifteen Scene Dataset	75
Table 4.3	Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes of Outex TC12 Dataset	77
Table 4.4	Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes of CUReT Dataset	79

List of Figures

Chapter Number. Figure Number	**Contents**	**Page**
Figure 1.1	Traditional Image Processing System Stages	3
Figure 1.2	Image Enhancement	5
Figure 1.3	Image Compression for JPEG Format	5
Figure 1.4	Image Segmentation	6
Figure 1.5	Image Classification	7
Figure 1.6	Example of How the LBP- Descriptor Works	8
Figure 1.7	Different Neighborhood System	9
Figure 1.8	Rotation Invariant Mapping	9
Figure 2.1	LBP Neighboring Pixels System	16
Figure 2.2	Example of Calculation the Differences in LBP	17
Figure 2.3	Binary Code to Decimal Code	18
Figure 2.4	Example of Histogram Construction	18
Figure 2.5	Image Primitives Captured by LBP Patterns	19
Figure 2.6	LBP Rotation Invariant	21
Figure 2.7	Example of the Improved LBP	23
Figure 2.8	Example of a Median LBP	24
Figure 2.9	Example of Calculation ELBP Codes	26
Figure 2.10	Framework of CLBP	27

Chapter Number. Figure Number	**Contents**	**Page**
Figure 2.11	Illustration Code Generation in LTP Descriptor	28
Figure 2.12	Example of Converting LTP Code to LBP Code	29
Figure 2.13	Example of Splitting a Quinary Code into Four LBP Codes	31
Figure 2.14	Examples of Elongate LBP	34
Figure 2.15	Example of Code Generation in MB-LBP	35
Figure 2.16	Scheme of ELBP4,4,3 Operators	35
Figure 2.17	Spiral Neighborhood Topology	36
Figure 2.18	LBP and CS-LBP Features for a Neighborhood of 8-Pixels	37
Figure 2.19	Example of CBP (8,1) Descriptor	39
Figure 3.1	Neighborhood Structure of 8-Pixels in DILBP	49
Figure 3.2	DILBP Neighboring System	49
Figure 3.3	Example of DILBP Patterns: (a) Decrement Pattern (b) Increment Pattern (c) Line Pattern (d) Insignificant Pattern	51
Figure 3.4	Example of Converting a Quaternary Code into Two Binary Codes	52
Figure 3.5	Example of Histograms (a) Histograms Construction for the Upper and Lower Pattern, (b) Two Histograms Concatenations	54
Figure 3.6	DILBP Flowchart Explain the Proposed DILBP Descriptor	55
Figure 3.7	Example of Line-End Pattern That are Captured by DILBP Descriptor	58

Chapter Number. Figure Number	**Contents**	**Page**
Figure 3.8	Example of Various Corner Shape Pattern That are Captured by DILBP Descriptor	59
Figure 3.9	Example of Various Edges Pattern That are Captured by DILBP	59
Figure 3.10	Example of Spot and Flat Pattern That are Captured by DILBP	60
Figure 3.11	Example of Lines Pattern That are Captured by DILBP	60
Figure 3.12	Illustration of DILBP's Illumination Insensitivity	61
Figure 3.13	Illustration of DILBP Noise Resistance	62
Figure 3.14	Illustration of DILBP's Rotation Invariance	63
Figure 3.15	Illustration of Encoding Significant Non-Uniform Pattern Using DILBP Descriptor	64
Figure 4.1	Examples of The Fifteen Scene Categories	68
Figure 4.2	Examples of Different 24 Classes in Outex TC12 Dataset	69
Figure 4.3	The 61 Classes in CUReT Dataset	70
Figure 4.4	Implementation Flowchart for Proposed DILBP	71
Figure 4.5	Accuracy Results of Descriptors in Basic and Rotation Invariant Modes Over Fifteen Scene Dataset	74
Figure 4.6	Accuracy Results of the Compared Descriptors in Basic and Rotation Invariant Modes Over Outex Tc12 Dataset	76
Figure 4.7	Accuracy Results of the Compared Descriptors in Basic and Rotation Invariant Modes Over CUReT Dataset	78

List of Abbreviations

Abbreviations	Meaning
ALG	Average Local Gray Level
BRISK	Binary Robust Invariant Scalable Keypoints
CBP	Centralized Binary Pattern
CLBP	Completed Local Binary Patterns
Co-HOG	Co-occurrence Histograms of Oriented Gradients
CS-LBP	Center-Symmetric Local Binary Patterns
FLBP	Fuzzy Local Binary Patterns
FREAK	Fast Retina Keypoint
HLBP	Hamming Local Binary Patterns
HOG	Histogram Oriented Gradients
ILBP	Improved Local Binary Pattern
LBP	Local Binary Pattern
LQP	Local Quinary Patterns
Lsym	Level of Symmetry
LTP	Local Ternary Patterns
MB-LBP	Multi-Block Local Binary Patterns
MLBP	Median Local Binary Patterns
MSER	Maximally Stable Extremal Regions
RLBP	Robust Local Binary Patterns

Abbreviations	Meaning
SIFT	Scale-Invariant Feature Transform
SLBP	Soft Local Binary Patterns
SURF	Speeded Up Robust Features
SVM	Support Vector Machine

Image Classification Based on Enhancement of Local Binary Pattern

Prepared By

Ahmad A. Suleiman

Supervisor

Dr. Mudhfar Al-Jarrah

Abstract

Image classification based on feature extraction from the image, by grouping images into categories, plays an important role in various computer application. Several approaches for extracting features from images have been presented in the literature. Local Binary Patterns (LBP) is one of the most common approaches due to its computational simplicity, its invariance to illumination changes and its reliability in image classification because it captures most of the essential visual features of the image. Although LBP has many advantages, however, it produces a large descriptor of 8-bits for each pixel and it is sensitive to image rotation. In this thesis, a new descriptor is proposed based on the original LBP, which is called Diagonal Intersection Local Binary Pattern (DILBP) that captures primitive properties of images such as edges, corners, line-ends, spots, flat areas, and other local characteristics such as lines primitives, which have not been addressed in the original LBP. The proposed Diagonal Intersection LBP (DILBP) uses a new technique to analyze the differences in intensity between the center pixel and its neighboring pixels, by comparing the center pixels with a pair of opposite pixels in the same diagonal. As a result, DILBP descriptor reduces the length of feature vector from 8-bit in the original LBP to 4-digit, which in turn reduces the vector length of histogram and reduces storage requirements for feature extraction from the image. Additionally, the proposed DILBP descriptor has other benefits such as maintaining image information, invariance to illumination change and rotation invariance. The proposed DILBP descriptor has been implemented and evaluated using the Fifteen Scene dataset, in addition to two public texture datasets: Outex dataset and CUReT dataset. Support Vector Machine (SVM) multi-class classifier is used in this thesis for classification purpose. The experimental results over Fifteen Scene dataset showed that the accuracy enhancement of proposed DILBP descriptor compared to the original LBP is around 1.5% and the enhancement over the CS-LBP is around 45.4%. In Outex TC 12 dataset, the accuracy enhancement of proposed DILBP descriptor compared to the original LBP is around 3.8% and the enhancement over the CS-LBP is around 15.3%. While in CUReT dataset, the accuracy enhancement of proposed DILBP descriptor compared to the CS-LBP is around 48.4% but is lower than the original LBP at a decreased rate of 2.9%. This thesis presents conclusions and suggestions for future enhancement of the proposed descriptor.

Keywords: Image Classification, Local Binary Pattern, Diagonal Intersection Local Binary Pattern, Rotation Invariance, Feature Extraction, Image Descriptor.

تصنيف الصور استنادا إلى تحسين النمط الثنائي المحلي

إعداد

أحمد عبدالحميد سليمان

إشراف

الدكتور مظفر الجراح

المُلخص

تصنيف الصور القائم على الخصائص المستخرجة من الصورة من خلال تجميع الصور إلى فئات، يلعب دورا هاما في تطبيقات الكمبيوتر المتعددة. وقد تم عرض العديد من النهج لاستخراج الخصائص من الصور في الأدب. النمط الثنائي المحلي (LBP), يعد واحد من الأكثر النهج شيوعا وذلك لبساطتها الحسابية، وثباته للتغير في الإضاءة وموثوقيته في تصنيف الصور وذلك لالتقاطة اغلب الخصائص المرئية الأساسية من الصورة. بالرغم من ان النمط الثنائي المحلي لديه العديد من المزايا، فإنه، مع ذلك، ينتج واصف بحجم طويل (8-بت) لكل بكسل، بالاضافة الى حساسيته في ما يخص دوران الصور. في هذه الأطروحة، تم اقتراح واصف جديد على أساس النمط الثنائي المحلي الأصلي، والذي يدعى التقاطع القطري النمط الثنائي المحلي (DILBP) الذي يلتقط الخصائص الاولية للصور مثل الحواف والزوايا، نهاية الخط، البقع، والمناطق المتجانسة وغيرها من الخصائص المحلية مثل خاصية الخطوط ، والتي لم يتم تناولها في النمط الثنائي المحلي الأصلي. يستخدم التقاطع القطري النمط الثنائي المحلي المقترح (DILBP) تقنية جديدة لتحليل الاختلافات في الكثافة بين بكسل المركز والبكسل المجاورة لها، من خلال مقارنة بكسل المركز مع زوج من البكسل المتقابلة في نفس القطر. ونتيجةً لذلك، يقلل الواصف المقترح DILBP من طول متجة الخاصية من 8 بت في النمط الثنائي المحلي الأصلي إلى

4 ديجت، الأمر الذي يؤدي إلى تقليل طول متجة الرسم البياني وبالتالي يؤدي إلى تقليل المساحة المطلوبة للخاصية المستخرجة من الصورة. بالإضافة إلى ذلك، فإن الواصف المقترح DILBP له فوائد أخرى مثل الحفاظ على معلومات الصورة، الثبات للتغيير في الإضاءة والدوران. تم تنفيذ الواصف المقترح (DILBP)، وتقييمه باستخدام مجموعة بيانات صور مشاهد طبيعية Fifteen Scene Dataset، بالاضافة الى مجموعتي بيانات عامتين هما مجموعة بيانات Outex TC12, ومجموعة بيانات CUReT. يستخدم ، شعاع الدعم الآلي (SVM) متعددة الطبقات في هذه الأطروحة لتصنيف وتمييز الصور. أظهرت النتائج التجريبية على مجموعة بيانات Fifteen Scene dataset أن تحسين دقة الواصف المقترح DILBP مقارنة مع LBP الأصلي هي ما يقارب 1.5٪ و التحسين على CS-LBP هي ما يقارب 45.4٪ . في مجموعة بيانات Outex TC12، تحسين دقة الواصف المقترح DILBP مقارنة مع LBP الأصلي هي حوالي 3.8٪ , في حين التحسين على CS-LBP حوالي 15.3٪. بينما في مجموعة بيانات CUReT، تحسين دقة الواصف المقترح DILBP مقارنة مع CS-LBP هي حوالي 48.4٪، ولكنها أقل من LBP الأصلي بمعدل انخفاض تقريبا 2.9٪. وتقدم هذه الأطروحة استنتاجات ومقترحات من أجل تحسين الوصف المقترح في المستقبل.

الكلمات المفتاحية: تصنيف الصور، النمط الثنائي المحلي، تقاطع قطري النمط الثنائي المحلي، ثابت الدوران، الميزة المستخرجة, واصف الصور .

Chapter one

Introduction

1.1 Introduction

Image represents a precise reflection of a scene in various fields, such as natural, medical and remote sensing. Although images have an important role in this life, it is difficult to store a huge number of captured images in its hard-format and it is impossible to modify and enhance such forms of images. Therefore, the image is digitized and transformed into a suitable form that can be saved in a computer memory or some storage devices. Image digitization is implemented by either scanning hard-copy images or by capturing the scene directly using a digital camera. Digital images are composed of multiple small pieces of colours, called pixels, which are arranged in a matrix with specific height and width. Consequently, the size of the image is specified by the number of its pixels.

To gain the most benefit of images, image processing techniques are applied to digital images, to enhance, modify or recognize the content of the images. One of the common image processing techniques is image classification, which categorizes image into groups based on their content. The precision and performance of the classification technique depends essentially on two components, the classification algorithm and the features that are extracted from the image, which represent the image properties (Sheha, Mabrouk, & Sharawy, 2012).

Local Binary Pattern (LBP) feature is an example of local features that was proven to be an efficient descriptor for its computational simplicity and its resistance to change in illumination (O'Connor & Roy, 2013). Improving the output of image classification by enhancing original LBP descriptor and reducing the size of LBP descriptor are the core motivations of this thesis.

1.2 Background of the study

Digital image processing consists of various techniques that collectively aim at enhancing the content of the image, extract its properties and recognize its content. In general, the traditional image processing system consists of six stages, as presented in the following order, and as illustrated in Figure 1.1 (Young, Gebrands, & Van, 1998).

1. **Image Collection:** is the process of obtaining the digital images by the optical sensor, such as camera photos or scanner.
2. **Image Pre-processing:** is the process of processing images using filters.
3. **Segmentation:** is the process of separating important information of image, such as separating an object from the background.
4. **Features Extraction:** is the extraction of features that represent the image properties either globally or locally.
5. **Classification:** is categorizing images in classes based on their content.
6. **Image Comprehension:** is creating a perception and understanding of the image.

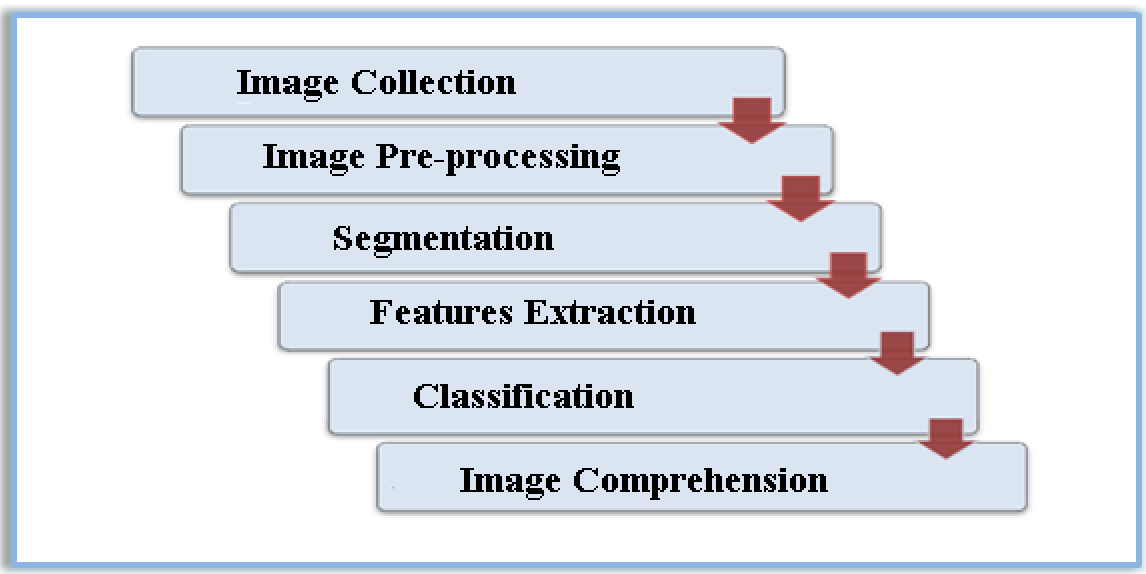


Figure 1.1: Traditional Image Processing System Stages

Digital Image processing fields are numerous, examples of these fields are: medicine (Bone scan, radiography, vascular imaging, and magnetic resonance), nuclear medicine, astronomy, remote sensing, and monitoring systems. Accordingly, image processing has been used in many industrial applications (Saxena, Sharma, & Sharma, 2016).

Digital Image processing has three levels depending on the purpose of processing. The first level called Low-Level, in this level the processing is performed at the pixel level, where the input and the output is a digital image (e.g. image enhancement). The second level called Middle-Level, the input of this level is an image and the output are features extracted from the image (e.g. edges, corners). Finally, the third level is High-Level, where the result is identification and recognition of the content of the input image such as, face recognition (Vadivambal & Jayas, 2015). Overall, the goals of image processing are varied, where some of them aim at enhancing the image or cutting significant parts of it and others aim at extracting features, which helps in recognizing objects in the image (Crnojević, Panić, Brkljač, Ćulibrk, Ačanski, & Vujić, 2014).

Digital image processing techniques are broad, the most important techniques that are used for digital image processing are as follows:

1. **Image Enhancement**: the aim of digital image enhancement is to improve the quality of the image. There are various technologies to enhance images, such as eliminating the noise that is caused by the vulnerability in camera or through image transmutation and storage, also reducing or removing blur from images. Figure 1.2 shows an example of image enhancement. Thus, the importance of enhancement is to correct and redistribute colors and lighting that implemented in different ways, such as increasing or decreasing contrast and brightness (Schettini & Corchs, 2010).

Figure 1.2: Image Enhancement (photofirstaid.co.uk).

2. **Image Compression:** image compression is an approach that decreases the volume of data in a certain image to reduce its storage capacity by discarding redundant or repetitive details. Image compression is very crucial because without compression it will be hard to distribute and share images because uncompressed images consume huge memory space. Figure 1.3 shows an example of image compression. Image compression has two approaches, lossy approach, and lossless approach. In the lossy approach, some details are missed through the compression manner such as JPEG format, while in the lossless scheme the image is compressed as much as possible so that no detail is lost in the image such as GIF format (Said & Pearlman, 1996).

Figure 1.3: Image Compression for JPEG Format

3. **Image Segmentation**: image segmentation is one of the significant techniques in the fields of image processing. The purpose of the segmentation is to extract the distinguishing parts of the image and separating them from the background, as shown in Figure 1.4. Segmentation is implemented using different methods such as, distinguishing objects edges, the similarity in color and other similar properties. Many processes such as face recognition can be achieved by separating these distinguishing parts and calculating their volume (Despotović, Goossens, & Philips, 2015).
4. **Image Classification:** image classification uses various techniques to group images based on their extracted features. Classification strategies use two stages of processing: training and testing stage. In the training stage, image features are extracted and descriptions for each group is then created. In the testing stage, features are also extracted and compared with the previously constructed description of each group, in order to determine the most suitable group to be selected. Figure 1.5 shows the process of image classification (Priyadarsini, Sivakumar & Student, 2016).

Figure 1.4: Image Segmentation

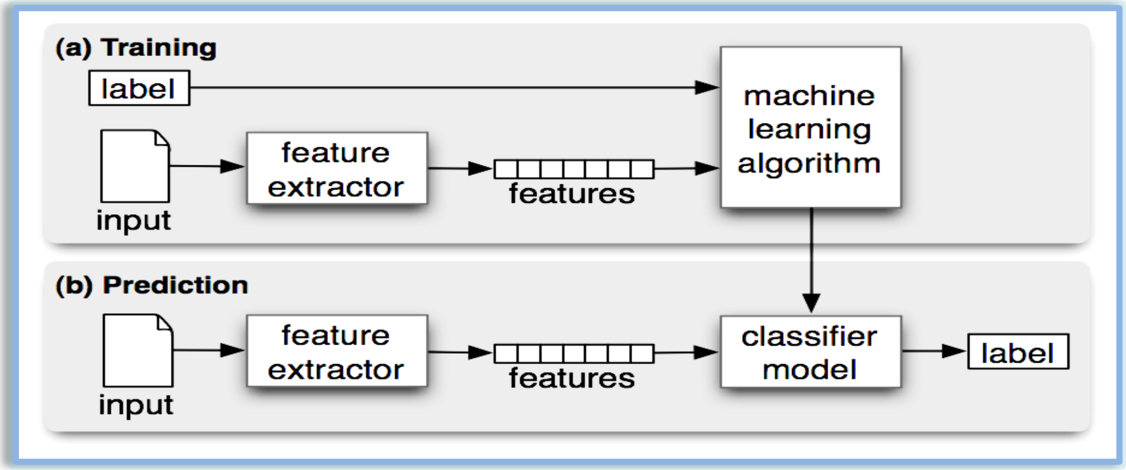


Figure 1.5: Stages of Image Classification

5. **Features Extraction**: features are properties that describe the content of the image in compact length. There are two types of features extracted from the images, these are, global features and local features. Global features are used to describe the whole image and are utilized in different application such as image retrieval, object detection, and image classification. Examples of global features are: Shape Matrices, Invariant Moments, Histogram Oriented Gradients (HOG) and co-occurrence histograms of oriented gradients (Co-HOG). Local features are used for describing patches in the image such as edge, corner, line end and spot, also are used for object recognition. Examples of a local feature are: SIFT, SURF, LBP, BRISK, MSER, and FREAK (Murphy, Torralba, Eaton, & Freeman, 2006).

1.3 Local Binary Pattern (LBP)

Local Binary Pattern (LBP) is an approach to extract local features from image by computing the local differences in intensity between the value of the center pixel and its surrounding pixel (neighbouring pixels). The aim of the local binary pattern is detecting different patterns using the differences between the neighbouring pixels and its centre pixel.

Patterns are then aggregated to describe the whole image. Consequently, LBP have been used with enormous applications such as texture classification (Ojala, Pietikainen, & Maenpaa, 2002), medical image annotation (Caputo, 2014), face recognition (Ahonen, Hadid, & Pietikäinen, 2004), fingerprint identification (Malathi & Meena, 2010) and medical image classification (Nanni Lumini, Brahnam, 2010). Usually, LBP descriptor is applied to grayscale images, so that each pixel has a value ranging from (0 to 255). The term $LBP_{P,R}$ is referring to original LBP descriptor where R is the radius of the circle and P is the number of neighbor pixels. Subsequently, the form of $LBP_{8,1}$ represents the simple form of the LBP with radius one and eight neighbors. Thus, in a simple form, the local binary pattern generates binary code with eight bits, where each bit has the value of either one or zero. The value one is assigned if the neighbor of the center pixel has a greater grey value than the center pixel and zero otherwise. The eight neighbors around the centre can be described with a code of eight bits. Figure 1.6 shows an example of the method of calculating the LBP descriptor.

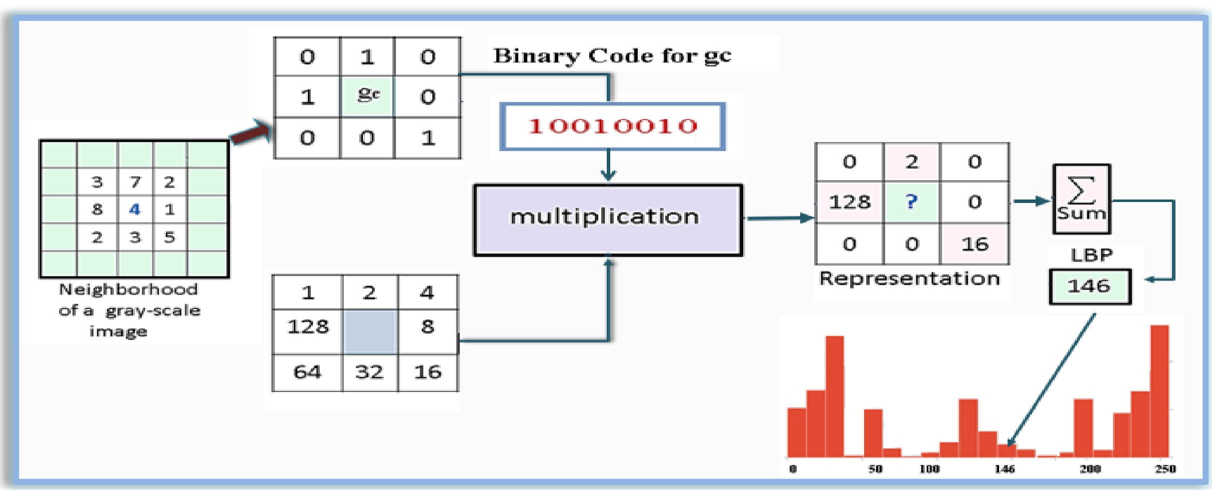


Figure 1.6: Example of How the LBP- Descriptor Works (Lindahl, 2007)

The generic LBP descriptor is not limited to represent only eight pixels with R value equal to one because using bilinear interpolation to make the LBP descriptor extendable to consider more neighborhood pixels with different radius values and cover wider regions as shown in Figure 1.7.

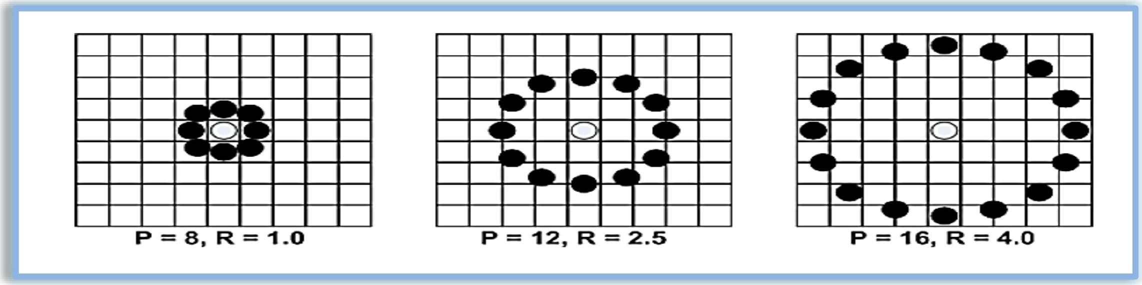


Figure 1.7: Different Neighborhood System (Ojala, Pietikäinen & Mäenpää, 2002)

Extracted features using LBP descriptor is invariant for image rotation. Rotation invariant means the ability to shift a bit of the binary code in a circular form to obtains the smallest value code, which generates a unique code for rotation invariant as shown in Figure 1.8.

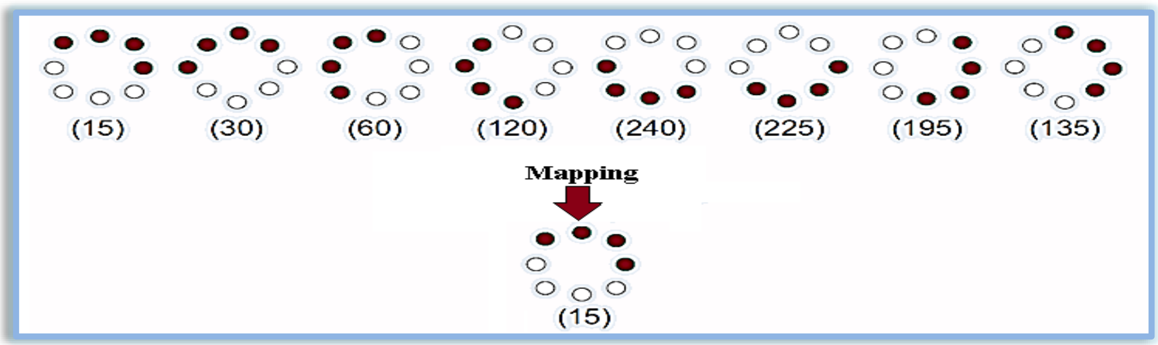


Figure 1.8: Rotation Invariant Mapping

1.4 Problem Statement

Several Local Binary Pattern (LBP) extensions were built based on original LBP descriptor. The goal of each new extension is to improve properties of the Local Binary Pattern. The main question in this thesis is how to obtain a robust descriptor, while preserving the invariant properties of LBP for illumination change and variance in rotation to improve the output of image classification and reducing the size of the original LBP descriptor to reduce the storage requirements.

1.5 Research Questions

The problem, addressed in this thesis, can be further clarified in the following questions:

1. How to modify the original LBP descriptor to reduce its size without degradation the discriminatory ability?
2. How to obtain the rotation and illumination invariance properties for the proposed LBP descriptor?
3. How to utilize the proposed LBP descriptor to enhance the output of image classification?
4. How to measure the discriminatory ability and prove the classification accuracy of the proposed LBP descriptor?

1.6 Goal and Objectives

The goal of this thesis is to propose a new extension based on the original LBP descriptor. This extension should increase the robustness of the original LBP by enhancing the rotation and illumination invariance properties. Furthermore, the proposed LBP descriptor should result in smaller feature vector compared to the original LBP.

The objectives of this research are as follows:

1. To reduce the size of the proposed descriptor by using sets of pixels rather than an individual pixel in comparing, analyzing, and generating codes without degradation the discriminatory ability.
2. To obtain rotation and illumination invariant in the proposed LBP descriptor.
3. To use the proposed LBP descriptor to enhance image classification.
4. To prove the discriminatory ability and classification accuracy of the proposed LBP descriptor in image classification, using public image datasets.

1.7 Motivation

The motivation of this thesis is to find an effective approach to classify the images into classes based on their content quickly and precisely. Automatic image classification overcomes human mistakes, which is considered the first motivation of this research, where the accuracy and efficiency of the classification procedure depend on the extracted image features. Original LBP is considered a robust and efficient descriptor although it generates long codes (8 bits) per pixel, thus, reducing the size of new descriptors is the second motivation of this thesis.

1.8 Contribution and Significance of the Research

Enhancing original LBP descriptor by reducing the length of the code for each pixel in the image is the main contribution of this thesis. As the code size is reduced, the number of bins in a histogram, which describes the whole image, will also be decreased. Also, the proposed descriptor uses a new way of computing the differences between pixels. Hence, this improvement is important for image classification, in which the space complexity is a significant issue.

1.9 Scope of the Study

The scope of this thesis is within analysis and classification of images based on the LBP approach. The work will involve investigating improvements of the original LBP descriptor for achieving better robustness against rotation and illumination change, and to improve discriminative ability of the proposed LBP. The work will perform analysis of public image datasets to measure the intended extensions of the LBP descriptor.

1.10 Limitations of the Study

The proposed descriptor is limited to greyscale images; hence color images require pre-processing to convert to greyscale. Also, the proposed descriptor is not fully insensitive to noise, as it is difficult to avoid sensitivity to noise fully and it is considered as a limitation in the proposed descriptor.

1.11 Thesis Outline

This chapter addressed digital image processing in general and provided an overview of the original LBP. The role of LBP descriptor in image classification is also presented in this chapter. Finally, the research problem, the objectives, limitations, and scope are also discussed. The rest of this thesis is organized as follows:

Chapter Two discusses and reviews the previous studies that are related to original LBP descriptor. The related works are classified and discussed comprehensively in a literature review. A comprehensive a literature review that clarified the extensions of LBP in term of advantages and dis advantages.

Chapter Three presents the new proposed descriptor in detail. Analyze the new proposed descriptor in term of the size, benefits, and limitations.

Chapter Four presents the implementation of the proposed descriptor. The results and its effectiveness are also discussed in this chapter.

Chapter Five will give a general summary of the thesis, summarizes the research findings and future works.

Chapter Two

Background and Literature Review

2.1 Overview

This chapter presents the factors, motivations, and details of the approaches that have extended the Local Binary Pattern (LBP) from 1996 until now. Section 2.2 presents an introduction to the Local Binary Pattern as a local feature. Section 2.3 explains the mechanism of reducing the size of the original LBP, these are, uniform and rotation invariant. In addition to the importance of LBP and its extensions, and how they help in classification of the image is also explained. Section 2.4 shows different LBP extensions, how they operate, their goals and a comparison between these extensions in terms of advantages and disadvantages. Section 2.5 presents overall comparison between LBP extensions. Section 2.6 provides a summary of this chapter.

2.2 Introduction

Image classification is one of the important fields in digital image processing, which relies essentially on the features extracted from the image to obtain an effective approach for image classification. As known, there are two types of features: global features and local features, which have been discussed in the previous chapter. In this chapter, local features are discussed. One of the most commonly utilized approaches that is used to extract local features from the image is the Local Binary Pattern (LBP).

2.3 Background

Local Binary Patterns (LBP) has emerged with the continuous efforts of developing an expressive local feature descriptor. LBP is a non-parametric descriptor that is used to capture the local structure of image effectively by converting an image into numeric labels (decimal number) that resulted from comparing every pixel in the image with its neighborhood and using these labels to build an image descriptor in a form of a histogram. (Huang, Shan, Ardabilian, Wang, & Chen, 2011). The popularity of the LBP comes from the fact that LBP is insensitive to the monotonic illumination changes, its computational simplicity, and it a robustness to describe and capture different local patterns in the image.

LBP has been used, tested, and verified in many applications such as, face image analysis, texture classification, image retrieval (Junding, Shisong, & Xiaosheng,2010), motion analysis (Pietikäinen, 2005), biomedical and aerial image analysis (Oliver, Lladó, Freixenet, & Martí,2007) and medical image classification (Huang, & Wang, 2007).

2.3.1 Local Binary Pattern Mechanism

To declare the mechanism of the local binary pattern, some basic concepts should be identified and clarified. Neighborhood pixels are represented by the set $\{P1, P2, \ldots, Pn\}$, where n denotes to the number of neighbour pixels, which are a collection of points equally distributed on a circle with distance R from the centre pixel.

The center pixel $\boldsymbol{c}$ is the matter core of the LBP and it is any pixel in the image. However, the pixel on the image border does not consider as center pixel because it is not completely covered by a circle of pixels. The points that do not fall within the center of pixels and located between neighborhood pixels is interpolated using bilinear interpolation, thus allowing for any radius and any number of points of the neighborhood. Figure 2.1 shows some examples of the neighborhood pixels.

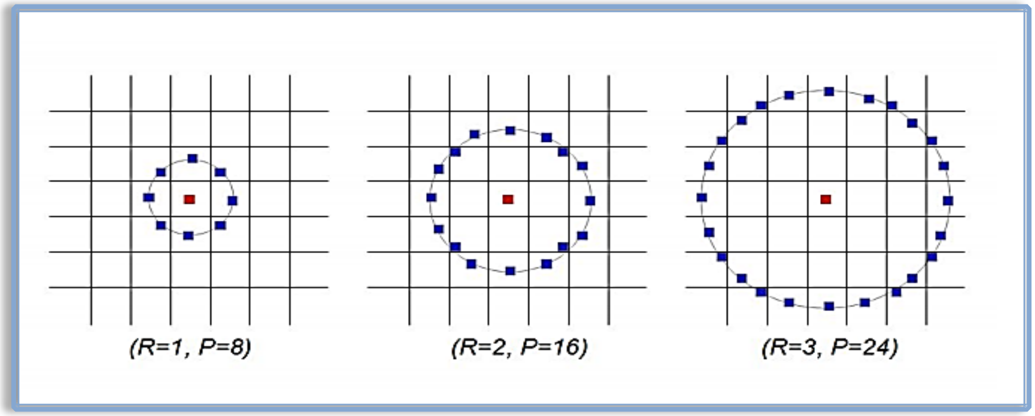


Figure 2.1: LBP Neighboring Pixels System (Huang et al., 2011)

Based on the previous concepts to calculate the LBP for grayscale image, the value of center pixel will be compared with the values of neighboring pixels, starting from one neighborhood pixel, and moving to the others in specific orientation, a clockwise or anti- clockwise orientation, that should be preserved for all other pixels. For instance, using $3x3$ block or patch will give eight pixels around the center pixel, this leads to eight comparisons in intensity with the center pixel and the output of comparisons will be converted into binary code as shown in Figure 2.2**.**

The threshold function is applied to the values were obtained from the difference between the value of the center pixel and the neighbors pixel values, the threshold is usually identified as zero, but sometimes is selected by the user or is selected under certain circumstances, if the value of the differences is greater or equal to the threshold value, the output of the thresholding will be one, while if the value is less than the threshold the output will be zero, this gives information about the type of generated patterns around the center pixel, such as being edge, corner or spot.

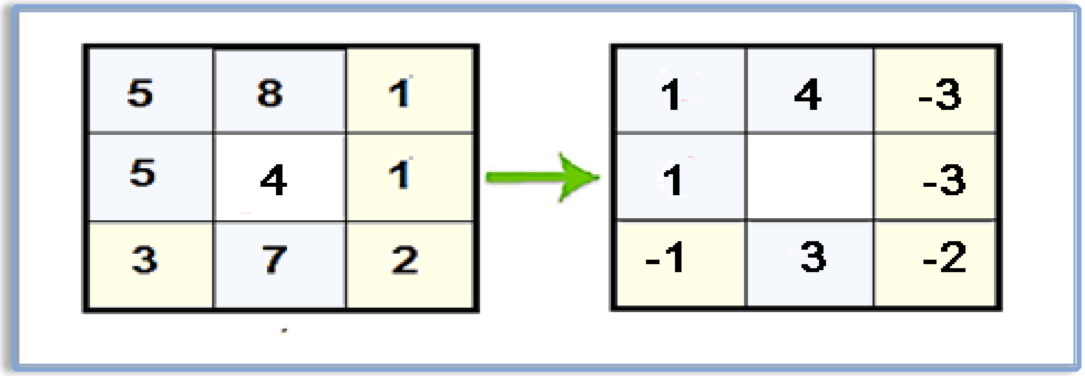


Figure 2.2: Example of Calculation the Differences in LBP (Rosebrock, 2015)

After the threshold is applied to the differences values, the output will be either zero or one. Given that each neighboring pixel has a value of zero or one, the aggregated values of all neighborhoods around a center pixel will form a binary code. This binary code of n-bits is then converted into decimal form to gain the final value for the center pixel. This conversion is executed by multiplying the value of each bit with 2^i where i is the index of the bit $\{0 \ldots n-1\}$, as given in Equation 2.1.

$$LBP_{P,R} = \sum_{i=0}^{i=p-1} s(\, g_i \; - \; g_c \,)2^i, \quad s(v)\begin{cases} 1 & v \geq 0 \\ 0 & v < 0 \end{cases} \qquad \textbf{(2.1)}$$

where v is the difference between intensity value of the neighbor pixel and the intensity value of the center pixel, R is the radius size, P is the number of neighboring pixels and $s(\mathrm{v})$ is a thresholding function. Figure 2.3 gives an example of Code Generation in LBP.

The values obtained using Equation 2.1 for each pixel in the image will be converted to the decimal number and then they are used to construct a histogram for the image. Thus, each pixel can contribute in one bin in a histogram, which represents descriptor of the entire image.

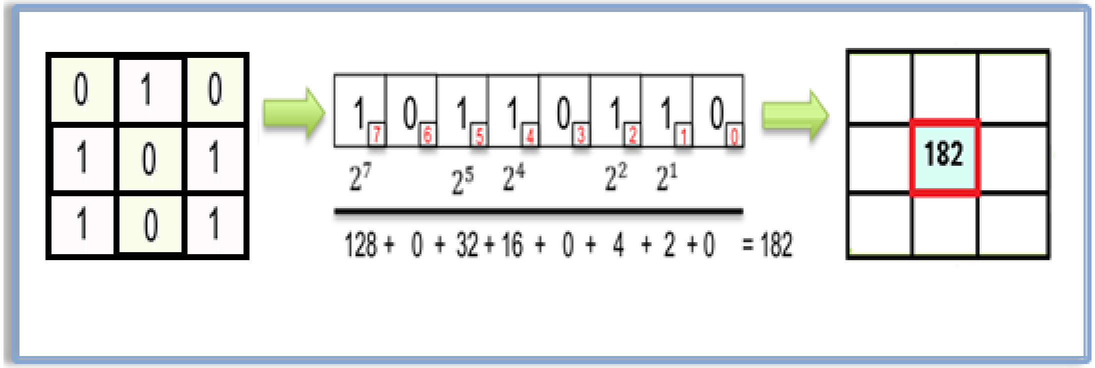


Figure 2.3: Example of Converting Binary Code to Decimal Number (Rosebrock, 2015)

For instance, when P is equal to eight this lead to generate 256 different patterns (2^8), and when P is equal to 12 there are 4096 different patterns (2^{12}) and so on. Therefore, each bin of the histogram can be regarded as a "micro-texton" encoded by LBP (Ahonen, Hadid, & Pietikäinen, 2004), as illustrated in Figure 2.4.

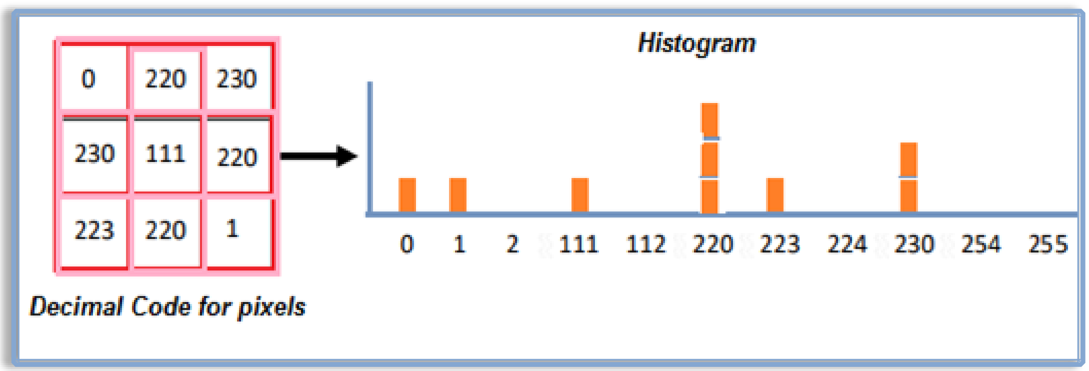


Figure 2.4: Example of Histogram Construction

2.3.2 LBP Minimization

Reducing the number of patterns generated by the Local Binary Pattern (LBP) is great interest for many researchers because this reduction makes LBP descriptor faster to be generated and more robust in detection primitive's properties in the image. In addition, dealing with a reduced number of patterns becomes easier and more flexible without losing information about the image primitives. This reduction can be implemented using two different algorithms, which are uniform patterns and rotation invariant.

2.3.2.1 Uniform Patterns

Binary codes which are created by LBP represents patterns in the image. Some patterns are more important than others as these patterns capture more significant primitives compared to others. Moreover, some of these patterns do not capture any primitives and might be produced as a matter of noise from other patterns as illustrated in Figure 2.5.

Based on the previous observations, patterns that are captured by LBP have been divided into two categories: uniform and non-uniform patterns. Patterns are defined as uniform if the number of transitions from one to zero or vice versa are less than or equal to two. On the contrary, patterns are considered as non-uniform patterns if the transitions are more than two. Uniform patterns considered more influential than the others and there is an equation to calculate the uniform pattern of LBP code as given in Equation 2.2 (Lindahl, 2011).

$$ULBP_{P,R} = \left| s(g_{p-1} - g_c) - s(g_0 - g_c) \right| + \sum_{p=1}^{p-1} \left| s(g_p - g_c) - s(g_{p-1} - g_c) \right| \qquad (2.2)$$

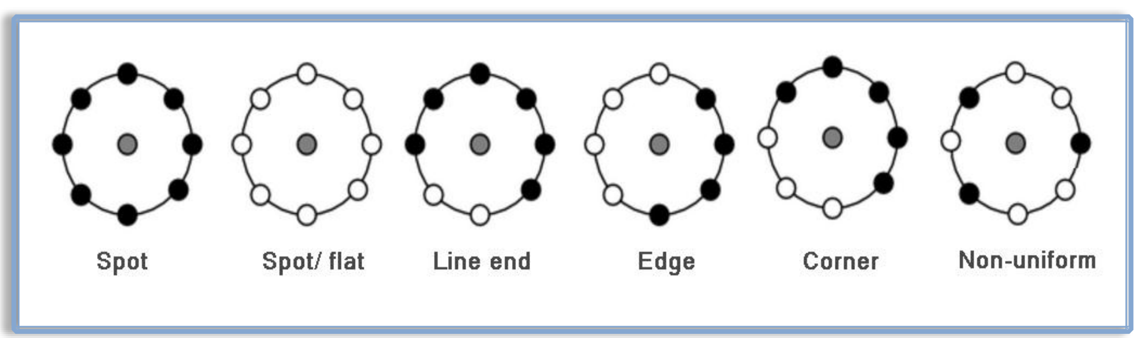


Figure 2.5: Image Primitives Captured by LBP Patterns

According to the Equation 2.2, $ULBP_{P,R} \leq 2$ refers to a uniform pattern, so the number of patterns is decreased to $P\ (P-1)+2$ uniform patterns, where P is the number of the neighbor pixels. Hence, the number of patterns for $LBP_{8,1}$ is 58 uniform patterns. As the non-uniform patterns are aggregated in a single bin, the final number of bins in a histogram is given by $P\ (P-1)+3$ bins. Table 2.1 illustration the uniform and non-uniform LBP patterns.

Table 2.1: Example of Uniform and Non-uniform LBP Patterns (Sánchez, 2010)

Circular Binary Pattern	Bitwise transitions	Uniform pattern
11111111	**0**	**Yes**
00001111	**1**	**Yes**
11001110	**3**	**No**
10101010	**8**	**No**
11111101	**2**	**Yes**
01001011	**6**	**No**
11000000	**1**	**Yes**

2.3.2.2 Rotation Invariance

One of the significant characteristics of the local binary pattern (LBP) is rotation invariant. To obtain invariance for rotation, the generated binary code by LBP is rotated in a circular way to the right continuously to get the lower value for binary code, this operation called bitwise shift operation, which applied on the generated binary code using ***ROR*** function. For instance, the binary code 11101000 is shifted to 01110100 then to 00111010 then to 00011101 and this will be the smallest value.

Rotation invariant codes are generated by implementing bitwise shift function as given in Equation 2.3.

$$LBP_{P,R}^{ri} = \min\{ROR(LBP_{P,R}, i) \mid i = 0, \dots, P-1\} \tag{2.3}$$

where ROR is a function for shifting the bits, thus, instead of producing 256 various patterns for LBP $_{8,1}$ only $\mathbf{36}$ different patterns, where the first-row patterns are considered as uniform patterns as illustrated in Figure 2.6.

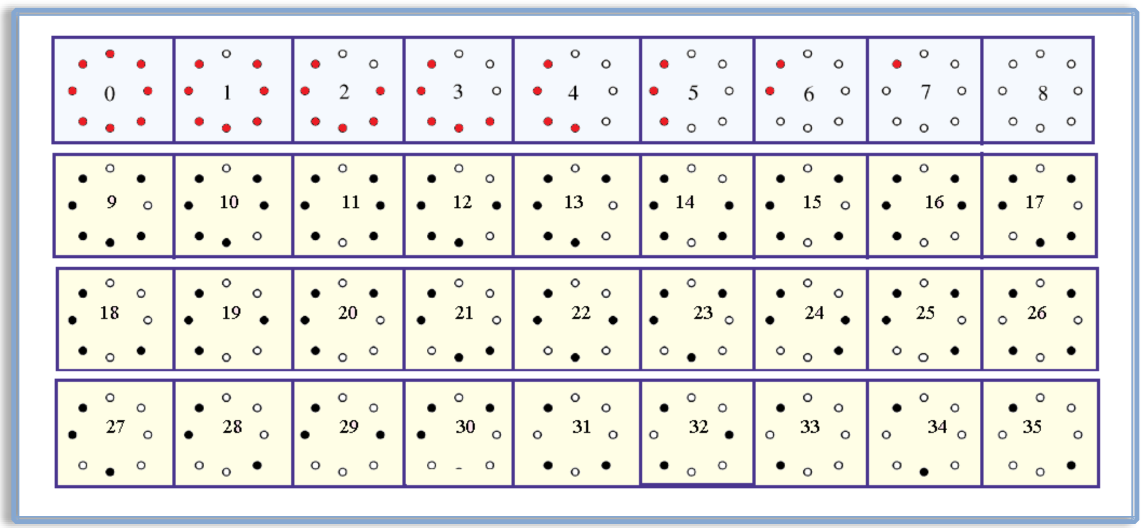


Figure 2.6: LBP Rotation Invariance

2.4 Related Work

Based on the literature, the original LBP is a powerful and efficient local feature descriptor due to its simplicity, efficiency, and flexibility. Besides, LBP can be easily modified, so that it becomes suitable for various types of problems.

The original LBP was first introduced by Ojala et al. (1996), and then several extensions and variants over the original LBP were proposed in the literature. All these extensions and variants have the same objectives, but they are implemented using different approaches in order to achieve the main goals, which are extracting robust and effective descriptors that are invariant to illumination changes, image rotation, noise effects and reducing the computation cost. The significant of reviewing the existing LBP variants approaches is to investigate the theory of building these extensions approaches and to highlight the weaknesses which must be addressed in the proposed work. The extensions of original LBP are divided into classes based on their objectives. Some of these extensions may be associated with more than one class. The classes of the LBP extensions are:

1. Increasing Discriminative Ability: which is performed by encoding an extra pattern or information.

2. Raising the Performance: which is performed by improving the robustness of LBP and reducing the noise sensibility.

3. The Neighborhood Topology: which is performed by selecting a suitable neighborhood set, the order of the neighbor pixels, its size, and the form of the neighborhood.

4. Minimization LBP Size: which is performed by minimizing the numbers of the potential patterns or integrating different patterns with each other.

2.4.1 Increasing Discriminative Ability

The original LBP extracts specific numbers of patterns to represent the local structure. Therefore, to improve the discriminative ability of the LBP, extra patterns or information are encoded.

Jin, Liu, Lu and Tong (2004) proposed a new descriptor called Improved Local Binary Patterns (ILBP) by modified the original LBP descriptor, to enhance the original LBP, increase its discriminative capability and describe more local information. The ILBP descriptor calculated the mean of the patch or block to be used as a threshold and then compared all pixels in the same block (including the central pixel) by that threshold, as given in Equation 2.4 and shown in Figure 2.7.

$$ILBP_{P,R} = \sum_{i=0}^{p-1} s(g_i - m)2^i + s(g_c - m)2^p, s(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (2.4)$$

where m is a mean of all pixels in the block and calculated as given in Equation 2.5.

$$m = \frac{1}{P+1}\left(\sum_{i=0}^{p-1} g_i + g_c\right) \quad (2.5)$$

The simple form of LBP $_{8,1}$ generated 2^8, which is equal to 256 patterns in a 3 ×3 block (patch), while ILBP generate 2^9, which is equal to 512 patterns. The idea of this technique is based on observing the effect of the central pixel, which gives extra information more than its neighbors pixel. ILBP gives satisfactory results that prove its discriminating ability.

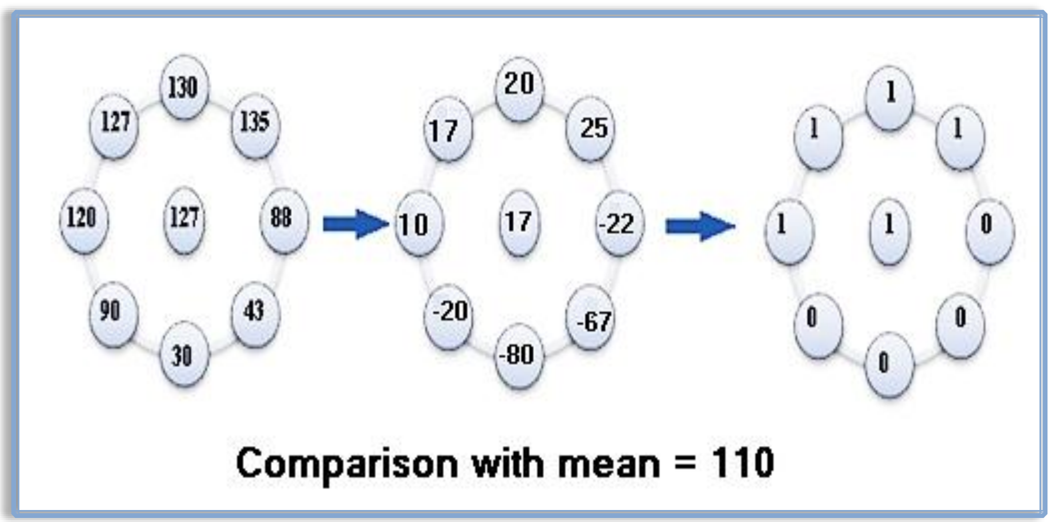


Figure 2.7: Example of the Improved LBP (ILBP)

Hafiane, Seetharaman, Palaniappan and Zavidovique (2008) presented a different approach based on calculating the differences between the pixels in the block and their median value, which called Median LBP (MLBP). This approach computed the differences values by comparing the pixels in the neighborhood with the median values of the block as given in Equation 2.6. Figure 2.8. shows an example of the process of Median LBP.

$$\boldsymbol{v_i} = g_i - LocalMedian\,, \quad \boldsymbol{v_{center}} = g_c - LocalMedian \tag{2.6}$$

The MLBP increased the size of the binary code through using the center pixel in calculation to add additional bit, which becomes more noise resistant than the original LBP. The median of the block is used as a threshold.

Yang and Wang (2007) improved the original LBP by using Hamming LBP approach to raise the discrimination power of the original LBP. The core idea of the Hamming LBP is to insert an additional step after thresholding. In this step, after dividing the patterns into uniform and non-uniform all the non-uniform patterns are transforming into uniform patterns rather than accumulating the non-uniform patterns into one bin in a histogram as in original LBP.

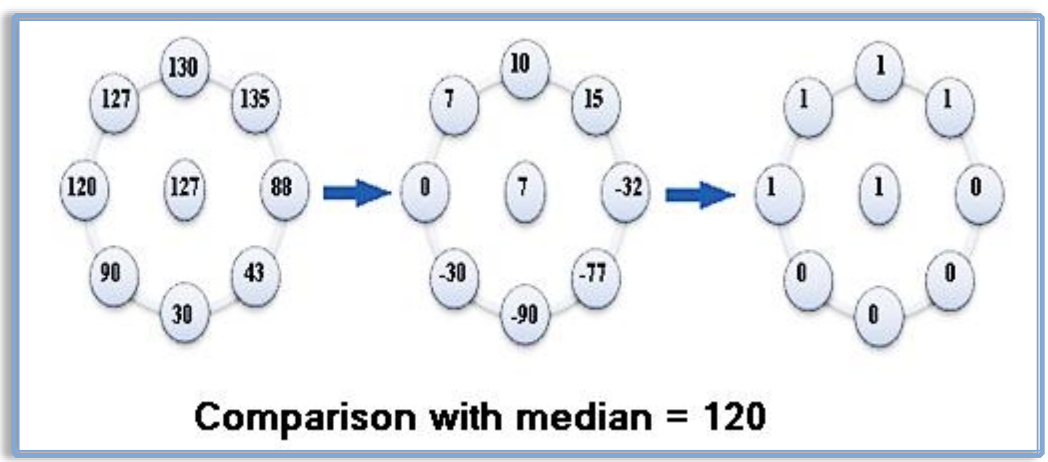


Figure 2.8: Example of a Median LBP

The non-uniform pattern is integrated into its closest uniform pattern by calculating the distances between the non-uniform pattern and all the uniform patterns using hamming distance, then the non-uniform pattern is transforming into the closest uniform pattern. For instance, the non-uniform pattern 10001110 is transformed into the uniform pattern 10001111 because its hamming distance is one, which is the least distance from any uniform pattern. In some cases, when the hamming distances between the non-uniform pattern and multiple uniform patterns are the same, the Euclidian distance for the non-uniform pattern and uniform patterns is computed and the minimum Euclidian distance between their decimal values will be chosen.

Huang, Wang and Wang (2007) proposed a new LBP extension to increase the discriminative power of the original LBP, this extension called Extended Local Binary Pattern (ELBP). The ELBP technique is based on the idea that each pattern can be encoded with more than one code using several layers. In the first layer L1, the code is calculated as original LBP, then the absolute difference between the grey values of the center pixel and neighborhood pixels are calculated as decimal number. The decimal number convert to binary and each bit in binary number contributed in the rest layers. Finally, each layer generates a binary code for the same pattern, which increases the discrimination ability for original LBP, because sometimes the first layer code does not give enough information for the pattern. Figure 2.9 illustration the layers of the ELBP, in which the first layer L1 represents the original LBP binary code that is equal to 11010011 in binary and 211 in decimal, while L2 represents absolute binary code that is equal to 01000000 in binary and 64 in decimal and so on.

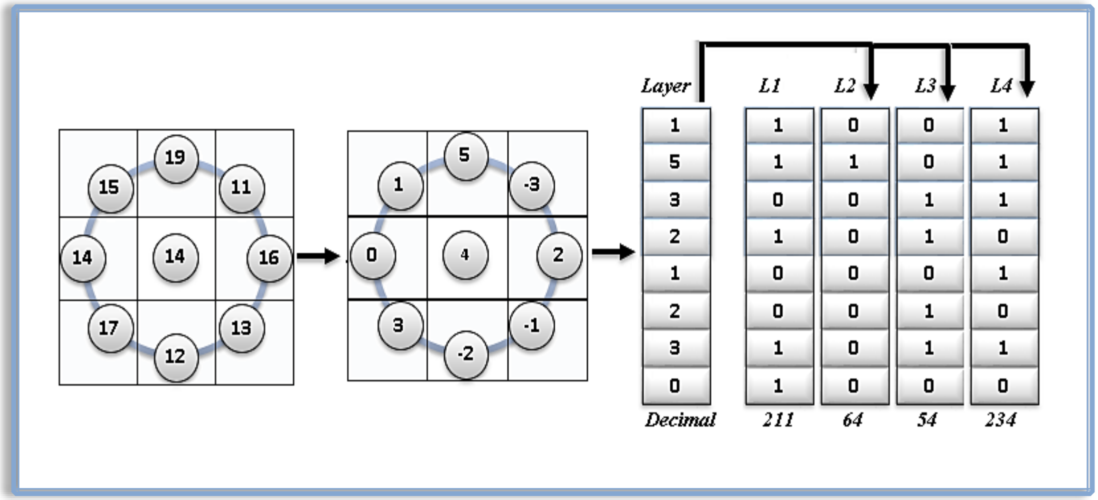


Figure 2.9: Example of Calculation ELBP Codes

Guo, Zhang and Zhang (2010) proposed a new approach, which is called Completed LBP (CLBP), by adding additional patterns to the original LBP patterns in order to improve the efficiency of original LBP. Every pixel in the image is expressed by three operations: CLBP- Sign (CLBP_S), CLBP-Magnitude (CLBP_M) and CLBP-Center (CLBP_C), these operations used the same neighborhood system that is used in the original LBP. CLBP_S is calculated similarly to calculate original LBP with encoding -1 instead of 0 in the generated code, while CLBP_M encodes the difference between the neighborhood pixels with the average differences in the entire block. CLBP_M catches the merely strong differences in intensity.

CLBP_C is computed by calculating the difference between the center pixel and the average of grey value of entire image. Then, the produced codes are accumulated to generate a final pattern of each pixel as shown in Figure 2.10. Two types of databases have been used to verifies the results of CLBP, Outex databases and the Columbia-Utrecht Reflection and Texture (CUReT) databases. The results showed that CLBP increased the discriminating ability than the original LBP.

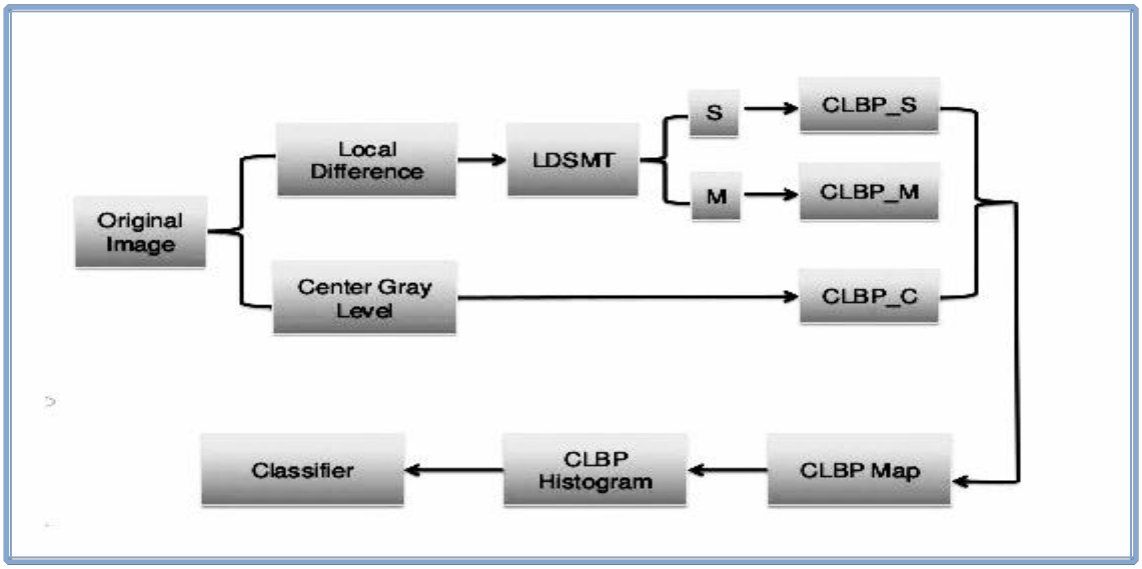


Figure 2.10: Framework of CLBP

Rassem and Khoo (2014) proposed a new approach as an extension to what so called, Local Ternary Pattern (LTP), which is called Completed Local Ternary Pattern (CLTP). In CLTP, every pixel in the image is described by three LTP values which are LTP- sign, LTP-magnitude, and LTP-center. The generated codes are aggregated to come out with the final pattern for each pixel with upper and lower patterns. The results confirmed the success of the proposed CLTP compared to the existing texture operators CLBP. The CLTP is more resistant to noise, rotation, and illumination variance, thus, achieved better classification accuracy compared to CLBP.

2.4.2 Raising LBP Performance

To enhance the performance of original LBP several significant issues were addressed, such as make the descriptor more noise-resistant and improving the robustness of the LBP extensions.

Tan and Triggs (2010) proposed a new extension of original LBP to removes the noise sensitivity, which is called Local Ternary Patterns (LTP), where preserves the important characteristic of original LBP which is computational simplicity. The Local Ternary Patterns technique expands LBP by using three values of encoding, these are: {-1,0,1}, as the output of the threshold function, as given in in Equation 2.7.

$$s(v) = \begin{cases} 1 & v \geq +t \\ 0 & -t < v < +t \\ -1 & v \leq -t \end{cases} \quad (2.7)$$

where t is a threshold value defined by the user, thus, if the difference value is between $+t$ and $-t$, the output will be 0, while if the value is greater than the threshold, then the value +1 is given and -1 values is given otherwise. The generated ternary codes are then transformed into upper and lower binary codes. Creating a ternary code and convert it into two binary code are shown in Figure 2.11 and Figure 2.12, respectively.

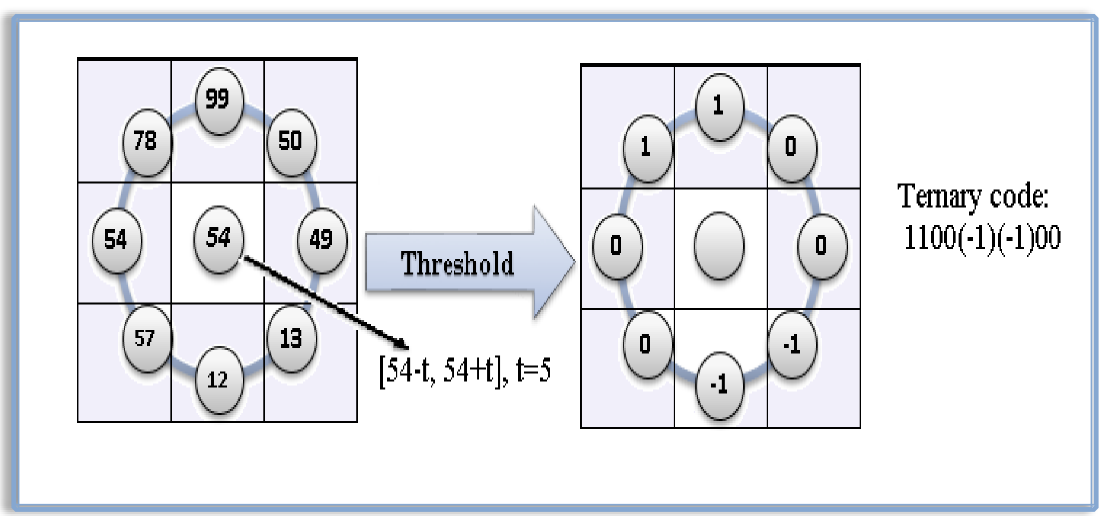


Figure 2.11: Illustration Code Generation in LTP Descriptor

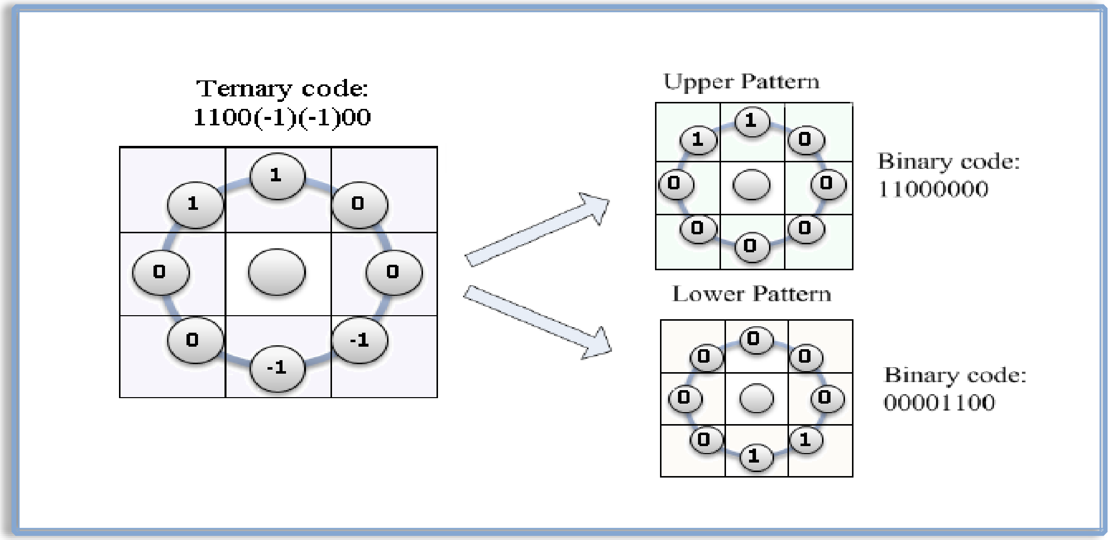


Figure 2.12: Example of Converting LTP Code to Two Binary Code

Ahonen and Pietikainen (2007) proposed a new approach to eliminating the sensitivity of the original LBP descriptor to noise by using a soft histogram. This approach is called Soft LBP (SLBP). This technique replaced the threshold function that is used in original LBP with two fuzzy membership functions as given in Equation 2.8 and Equation 2.9, respectively.

$$f_1, t(x) = \begin{cases} 0\ , & x < -t \\ 0.5 + 0.5\frac{x}{t}, & -t \leq x \leq t \\ 1\ , & x > t \end{cases} \tag{2.8}$$

$$f_0, t(x) = 1 - f_1, t(x) \tag{2.9}$$

where t is the variable factor dominates the degree of fuzzification implemented by the function. The idea of $SLBP$ revolves around that a single pixel can be shared with various bins in a histogram. Nevertheless, the total contribution of each pixel to all bins of the histogram should be always equal to one. The disadvantages of this approach are increase the sensitivity to illumination changes, increasing the complexity of the computations and rising the parameter setting problem.

Iakovidis, Keramidas and Maroulis (2008) introduced a novel approach, which is called Fuzzy LBP (FLBP), to improve the original LBP sensibility to noise. FLBP encoded the local structures in the image through extends the Local Binary Pattern (LBP) for including fuzzy logic in the representation of the patterns. This approach used two fuzzy rules and two functions (i.e.: $w0$ and $w1$) to convert the input variables to fuzzy variables, where function $w0$ determine the degree when Pi has a smaller gray value compared to Pc, as given in Equation 2.10. On the other hand, function $w1$ defines the degree when Pi has a greater gray value compared to Pc as given in Equation 2.11.

$$w0(i) = \begin{cases} 0 & If\ Pi \geq Pc + T \\ \frac{T-(pi-pc)}{2.T} & If\ \ Pc - T < Pi < Pc + T \\ 1 & If\ Pi \leq Pc - T \end{cases} \tag{2.10}$$

$$w1(i) = 1 - w0(i) \tag{2.11}$$

For both $w0$ and $w1$ the parameter T can be any value in the range $[0,255]$. Overall, the histograms of FLBP have no bin with a null value as compared to the original LBP, which usually includes various bins with null values. The results of experiments proved that the new descriptor is powerful for noisy image representations compared to the original LBP.

Nanni, Lumini and Brahnam (2010) proposed a novel approach to reduce the sensitivity of the original LBP descriptor to noise by taking into consideration the difference between neighbor pixels and the center pixel. This approach is called Local Quinary Pattern (LQP), wherein the difference between the value of the neighbor pixels and the value of the center pixel is encoded using five values and two thresholds (t1, t2), as given in Equation 2.12.

$$S(g_p, g_c, t_1, t_2) = \begin{cases} 2, & g_p \geq g_c + t_2 \\ 1, & g_c + t_1 \leq g_p < g_c + t_2 \\ 0, & g_c - t_1 \leq g_p < g_c + t_1 \\ -1, & g_c - t_2 \leq g_p < g_c - t_1 \\ -2, & otherwise \end{cases} \quad (2.12)$$

To reduce the complexity of employing a code with base 5 to encode the five cases, LQP uses four binary codes to represent the quinary code. For example, the generated four binary codes for the cases {−2, −1, 1, 2} is illustrated in Figure 2.13. These binary codes are collected in Four separate histograms and, finally, the histograms are concatenated to form the LQP feature vector.

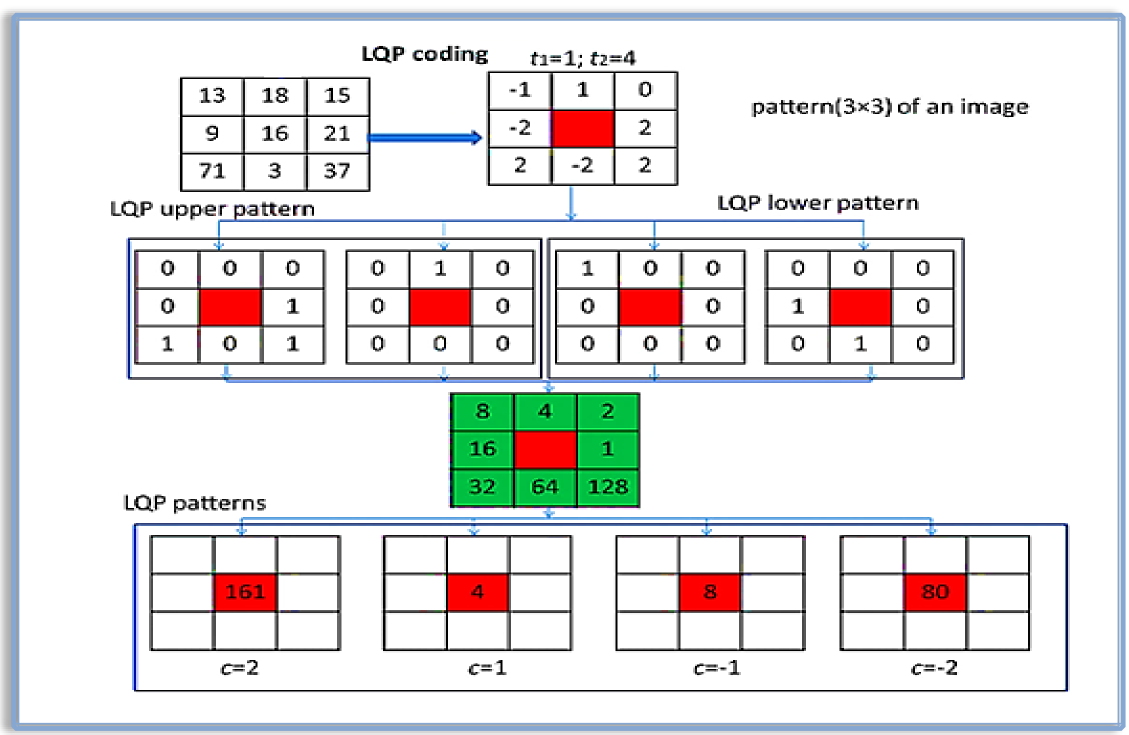


Figure 2.13: Example of Splitting a Quinary Code into Four LBP Codes

Zhao, Jia, Hu and Min (2013) introduced a new approach to solve the sensitivity of traditional LBP to noise and variant in illumination. The new approach is called Robust LBP (RLBP), which aims to modify the threshold of the original LBP to illuminate and noise sensitivity. The technique of the Robust local binary pattern (RLBP) is based on utilizing the Average Local Gray Level (ALG) as given in Equation 2.13

$$ALG = \frac{\sum_{i}^{8} g_i + g}{9} \tag{2.13}$$

where g denotes to the value of the center pixel and gi ($i = 0, \ldots, 8$) indicates to the value of the neighbor pixel. ALG represents the average gray level of local texture. Thus, RLBP uses ALG as a new threshold rather than fixed value. Accordingly, RLBP can be described as given in Equation 2.14.

$$RLBP_{P,R} = \sum_{i=0}^{i-1} s(g_p - ALG)2^i \tag{2.14}$$

ALG ignores the value of the central pixel, which is required to generate a robust descriptor. To solve this issue, Weighted Local Grey Level (WLG) is used as given in Equation 2.15

$$WLG = (\sum_{i=1}^{i=8} g_i + \alpha g)/8 + \alpha \tag{2.15}$$

The parameter α is set by the user, ALG is a specific case of WLG, while both are identical when α is equal to one. The final RLBP descriptor can be computed as given in Equation 2.16.

$$RLBP_{P,R} = \sum_{i=0}^{i-1} s\left(g_p - WLG\right)2^i \tag{2.16}$$

Heikkila and Pietikainen (2006) proposed a new approach to solve one of the original LBP limitations. The flat area limitation which occurs when the gray value of the adjacent pixel is close in value to the center pixel. According to the thresholding function of the original LBP descriptor, if both pixels are similar in value the result will be zero. On the other hand, if the values were identical, so the result will be one. This approach is called Robust Local Binary Patterns (RLBP), which aim to make the original LBP more powerful against these minor changes in pixel values. The essential concept of the RLBP depends on changing the thresholding function as given in Equation 2.17.

$$RLBP_{P,R(x,y,t)} = \sum_{i=0}^{i=p-1} s(\, P_i \, - \, (P_c - t)\,)2^i \tag{2.17}$$

where t is a threshold value defined by a user and should be an assigned as small value. This approach presents many benefits compared to other methods, such as being stable to the change in illumination in natural images and being extremely quick in computation, which is a significant feature.

2.4.3 Neighborhood Topology

Liao and Chung (2007) extended LBP with an approach that is called Elongated Local Binary Pattern (ELBP) with the aim to enhance the robustness and effectiveness of the original LBP descriptor. This descriptor modified the shape and the size of the neighborhood to fit some local structure which is important to some application such as face recognition. Neighborhood modification is implemented using an ellipse form instead of the circular form. The neighbor pixels in ELBP are distributed around the center pixel as ellipse shape with different distance from the center pixel, with x-distance and y-distance that are determined using parameters, A and B corresponding to x and y axis as illustrated in Figure 2.14. Although, ELBP has the ability to identify many significant primitives in the image such as the face parts primitives, however, it lost a rotation invariant and a lot of significant properties captured by the original LBP.

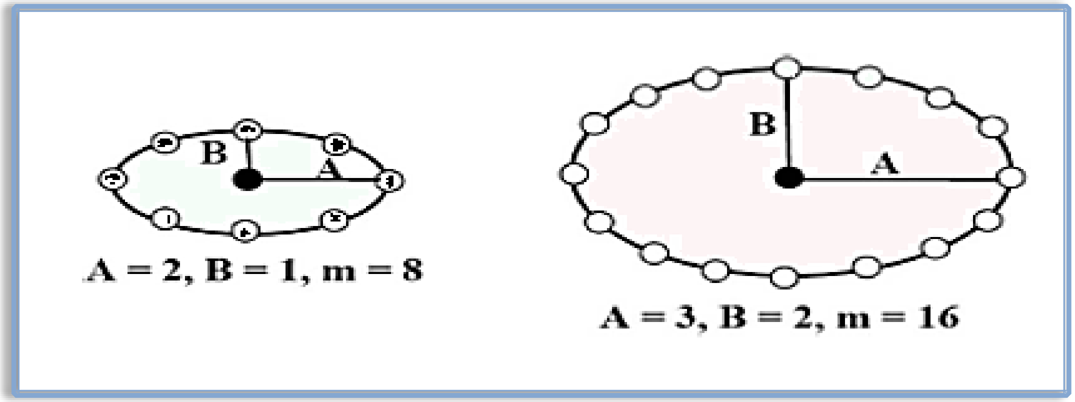


Figure 2.14: Examples of Elongate LBP

Zhang, Chu, Xiang, Liao and Li (2007) proposed a new LBP-based extension, which is called Multi-Block LBP (MB-LBP). that calculated the patterns of rectangle blocks in order to capture macrostructures instead, which give more details about local texture compared to the microstructures in the original LBP. MB-LBP encoded the rectangular areas intensities using the original LBP process. However, the essential concept of MB-LBP is calculating the differences in intensity by comparing the average intensity of the central region with the intensities of neighborhood sub-regions which can be a rectangle or a square. Subsequently, when the block size is equal to one, MB-LBP will be identical to the original LBP. Thus, original LBP is considered as a special case of the MB-LBP. Figure 2.15 shows an example of MB-LBP. The advantages of MB-LBP are: noise insensitivity and generating a smaller feature set compared to original LBP, thus, less time is required for saving, processing, and retrieving the feature set. Unfortunately, some significant primitives obtained by original LBP are lost by MB-LBP.

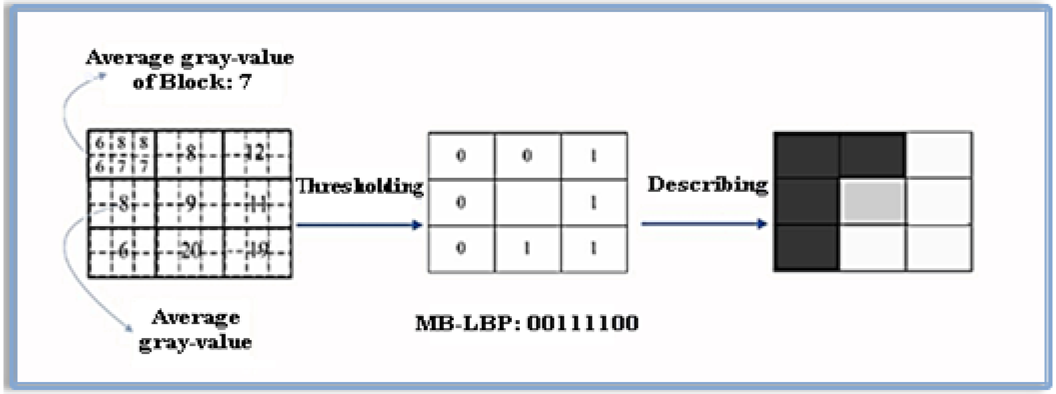


Figure 2.15: Example of Code Generation in MB-LBP

Král and Vrba (2017) proposed an LBP-extension, which is called Enhanced LBP (ELBP) to be used for face recognition. This approach enhanced the original LBP by utilizing larger center space and larger neighborhood space in code generation. ELBP depended on considering sets of neighborhood pixels, Gn, around a center set, Cn, where n is the number of neighborhood regions and Gc is a set of center pixels that are centered at the pixel Cc and r is the length between the center of Cn and center of Cc. The descriptor calculated the average of each set as $g'n = mean(Gn)$ and $g'c = mean(Gc)$. The descriptor created using this approach is like the original LBP. Thus, the proposed descriptor is expressed as $(ELBP\ x, y, r)$, where $x \in \{4,9\}, y \in \{4,9\}$ and r is the distance between the centers of Cn and Cc as shown in Figure 2.16.

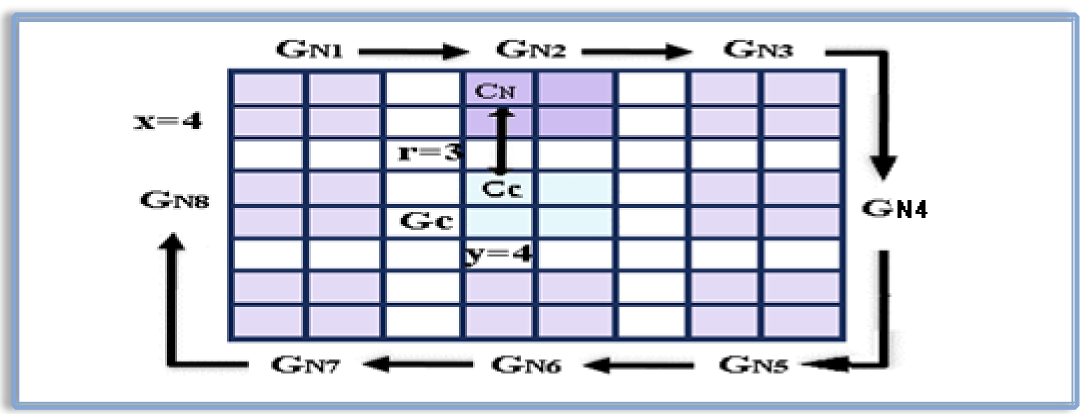


Figure 2.16: Scheme of ELBP $_{4,4,3}$ Operators

Kazak and Koc (2016) introduced a new approach that is called Spiral LBP (SLBP), which used the original LBP calculation with spiral topology rather than the circular topology that is used in the original LBP as shown in Figure 2.17. The Spiral neighborhood represents the anisotropic structural information within a superior way, which is considered significant to some applications such as face recognition.

The radius of the SLBP is expressed as $r = a + b\theta$, where a and b can be any numerical value. The variable a is used for turning the spiral and the variable b is used for keeping the distance between the turnings. Calculation SLBP codes is similar to the calculation of the original LBP. Firstly, the nearest neighbor is compared to the center pixel then the comparison continues to the farthest one. Thus, the binary code is generated depending on whether the central pixel is greater than, in grey level, less than or equal to the neighborhood. The obtained binary code is converted to a decimal value, then the histogram is created from the decimal values. The uniform spiral LBP is identical to the original LBP, while the non-uniform SLBP patterns are aggregated to only one bin in the histogram as original LBP.

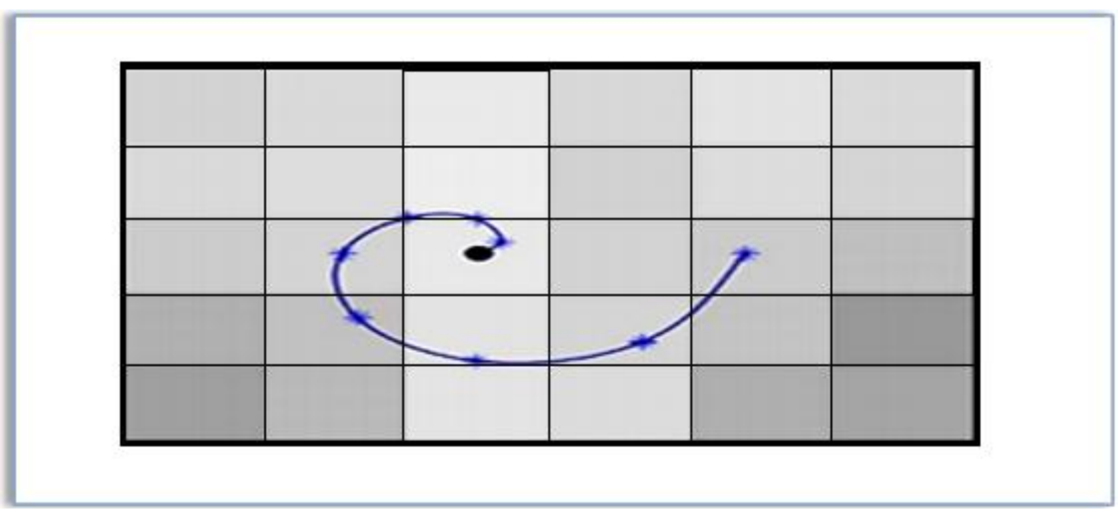


Figure 2.17: Spiral Neighborhood Topology

2.4.4 Minimization LBP Size

Heikkilä, Pietikäinen and Schmid (2006) proposed Center-Symmetric local binary pattern (CS-LBP) descriptor, which is based on reducing the size of the generated code by half by comparing the center-symmetric pairs of pixels at each time and generate a single bit for each pair. The center-symmetric pairs of pixels are located on the same diameter of the circle, as illustrates in Figure 2.18. CS-LBP reduces the size of the code, for instance, if the number of pixels of the neighborhood is equal to eight, the size of code will reduce to 4-bit. Thus, the histogram also decreased and became shorter than the histogram of the original LBP descriptor. The new descriptor has several advantages such as insensitivity to illumination changes, robustness on flat image areas, and computational simplicity.

Lahdenoja, Laiho and Paasio (2005) Introduced a novel approach, which called a symmetry-level descriptor to decrease the size of original LBP. The methodology of this approach depends on defining new notion for distinguishing between significant and non- significant patterns, which is called symmetry. The idea came from the observation that uniform and non-uniform patterns do not probably identify the significant patterns in the image.

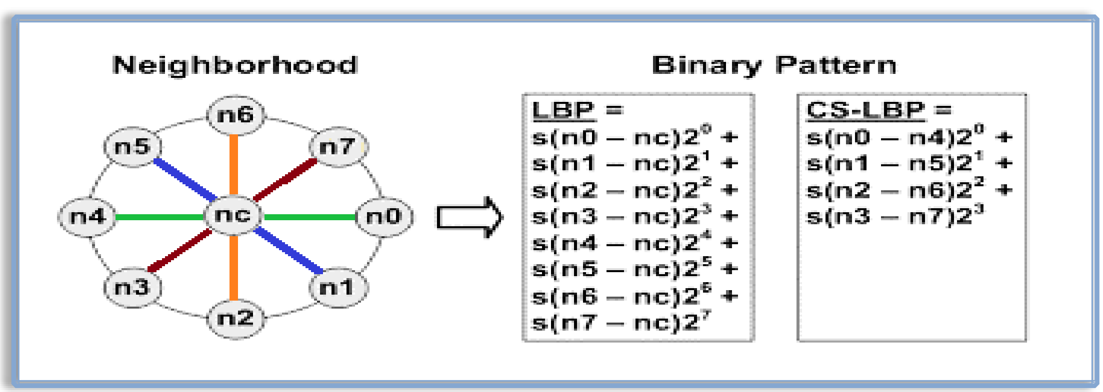


Figure 2.18: LBP and CS-LBP Features for a Neighborhood of 8-Pixels

Thus, the symmetrical patterns are considered as significant patterns in local textures. Experiments showed that symmetrical patterns increased the performance of face recognition. The degree of symmetry for the patterns is determined by computing the lower total number of ones and zeros in the pattern. For instance, the level of symmetry (Lsym) of the 11111100 and 01100000, is two. A pattern is considered as a high symmetric if it has the same number of ones and zeros, which is referred to as a symmetric edge. Whereas, patterns which include only ones or zeros considered as low symmetry level, significant patterns are those of high symmetry level.

Wong, Abu-Shareha, Pasha and Mandava (2013) proposed a new descriptor to enhance the original LBP in image classification especially in Chest X-ray (CXR) image, which is called Enhanced LBP (ELBP). The technique used in this approach is based on the idea of encoding the sets of a pixel in the same diagonal, which contains the center pixel and two pairs of the neighborhood pixels to the right and the left of the center pixel. The difference between the right pixel and the center pixel then the left pixel of the center pixel is calculated and encoded in a single bit. Subsequently, there are $P/2$ couples of neighborhood pixels. The function of the calculation is defined in Equation 2.18 and 2.19 respectively.

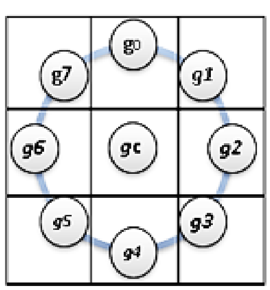
$$ELBP_{P,R}(X,Y) = \sum_{i=0}^{i=\left(\frac{p}{2}\right)-1} s[((g_r - g_c),(g_l - g_c))]2^{\mathrm{i}} \qquad (2.18)$$

where

$$s(v1,v2) = \begin{cases} 1, & if\ v1 > 0\ AND\ v2 \le 0\ OR\ v1 = 0\ AND\ v2 < 0 \\ 0, & if\ v1 < 0\ AND\ v2 \ge 0\ OR\ v1 = 0\ AND\ v2 > 0 \\ -1 & otherwise \end{cases} \qquad (2.19)$$

where gl refers to left neighborhood pixel and gr refers to the right neighborhood pixel. The function $\mathbf{s}$ $(\mathrm{v1}, \mathrm{v2})$ identifies three distinct patterns: the first one for incremented in grey value, second one for decremented and the third one for a fixed in intensity or abnormal change. The incremented and decremented almost represent an edge. To overcome the complexity, the generated ternary code is transformed into two binary code which are, upper code and lower code. For the upper code, a digit with value of -1 is turned into 0 and the rest stays the same. In lower code, the value of 0 is transformed to 1 and values of 1 and -1 are transformed to 0. The empirical outcomes have demonstrated that the new approach performs significantly better than the original LBP in addition to maintaining important edge properties.

Fu and Wei (2008) presented a novel approach called Centralized Binary Pattern (CBP) descriptor to address some problems in the original LBP and increase its discriminative power. One of these problems that the original LBP loses the local structure. In addition, original LBP does not give representation to the center pixel. The CBP descriptor considers the center pixel and assigns to it the highest value. Thus, increases the impact of the center pixel. CBP is based on comparing pairs of neighbors which are symmetrically around the center pixel. In addition, compare center pixels with the total average of the block. Figure 2.19 shows an illustrative example of calculation CBP. The results showed that CBP has many advantages such as reduce the size of the histogram, increasing discriminative power and insensitive to noise compared with the original LBP.



$$\begin{aligned} CBP(8,1) = {} & s(g_0 - g_4)2^0 + s(g_1 - g_5)2^1 \\ & + s(g_2 - g_6)2^2 + s(g_3 - g_7)2^3 \\ & + s(g_c - g_T)2^4 \end{aligned}$$

$$\downarrow$$

$$g_T = (g_0 + g_1 + \cdots + g_7 + g_c)/9$$

Figure 2.19: Example of CBP (8,1) Descriptor

2.5 LBP Variants Comparison

After reviewing previous studies that are related to LBP variations, a comparison between these variations are conducted as given in Table 2.2.

Table 2.2: Comparison Between LBP Variations

Subsection	Author & Year	Variations	Advantages	Disadvantages
Increasing of Discriminative Ability of LBP	Jin, Liu, Lu, &Tong, (**2004**)	Improved Local Binary Pattern (ILBP)	Considering the effects of the center pixel	Generating complex patterns
	Hafiane, Seetharaman, Palaniappan, & Zavidovique, (**2008**)	Median Local Binary Pattern (**MLBP**)	Considering the effects of the center pixel and reducing noise sensitivity	Generating complex patterns and losing grey-scale invariant
	Yang & Wang (**2007**)	Hamming Local Binary Pattern (**HLBP**)	Combining the non-uniform patterns into uniform pattern	Increasing the computational complexity
	Huang, Wang & Wang (**2007**)	Extended Local Binary Pattern (**ELBP**)	Increasing the discriminative power of the original LBP	Increasing the computational complexity
	Guo, Zhang, & Zhang, (**2010**)	Completed Local Binary Pattern (**CLBP**)	Enhancing LBP Performance	Increasing the computational complexity

Subsection	Author & Year	Variations	Advantages	Disadvantages
	Rassem & Khoo (**2014**)	Completed Local Ternary Pattern (**CLTP**)	Enhancing LBP Performance	Increasing the computational complexity
Raising the Performance of LBP	Tan & Triggs (**2010**)	Local Ternary Patterns (**LTP**)	Reducing noise sensitivity	Losing illumination-invariant
	Ahonen & Pietikäinen (**2007**)	Soft Local Binary Pattern (**SLBP**)	Reducing noise sensitivity	Increasing the computational complexity and losing invariant to monotonic grey scale
	Iakovidis, Keramidas, & Maroulis, (**2008**)	Fuzzy Local Binary Pattern (**FLBP**)	Reducing noise sensitivity	Increasing the computational complexity
	Nanni, Lumini & Brahnam (**2010**)	Local Quinary Pattern (**LQP**)	Reducing noise sensitivity	Lose illumination-invariant.
	Zhao, Jia, Hu & Min (**2013**)	Robust Local Binary Pattern (**RLBP**)	Reducing noise sensitivity	Computational Complexity

Subsection	Author & Year	Variations	Advantages	Disadvantages
	Heikkila & Pietikainen (**2006**)	Robust Local Binary Pattern (**RLBP**)	Improving robustness of the original LBP	Losing significant image information
Neighborhood Topology	Liao & Chung (**2007**)	Elongated Local Binary Pattern (**ELBP**)	Addressing special image primitives (anisotropic information)	Losing rotation-invariant
	Zhang, Chu, Xiang, Liao, & Li, (**2007**)	Multi-block Local Binary Pattern (**MB-LBP**)	Easing noise sensitivity problem	Losing significant image information
	Král & Vrba (**2017**)	Enhanced Local Binary Pattern (**ELBP**)	Reducing noise sensitivity	Losing significant image information
	Kazak & Koc (**2016**)	Spiral Local Binary Pattern (**SLBP**)	Addressing special image primitives (anisotropic information)	Losing rotation-invariant

Subsection	Author & Year	Variations	Advantages	Disadvantages
Minimization the Lengthening of LBP	Heikkilä, Pietikäinen, & Schmid, (**2006**)	Center-Symmetric Local Binary Pattern (**CS-LBP**)	Minimizing the length	Losing significant image information
	Lahdenoja, Laiho & Paasio (**2005**)	Level of Symmetry (**Lsym**)	Minimizing the length	Losing significant image information
	Wong, Abu-Shareha, Pasha & Mandava, (**2007**)	Enhanced Local Binary Patterns (**ELBP**)	Minimizing the length and using semi histogram	Losing significant image information
	Fu & Wei (**2008**)	Centralized Binary Pattern (**CBP**)	Increasing insensitive to noise and reducing the size	Losing significant image information

2.6 Classification Algorithms

Classification algorithm aims at building a model from a set of the training image and then uses the built model for classifying some images with unknown classes to their proper class. Numerous algorithms were proposed for classification with several models. The aim of all these algorithms is to perform classification efficiently, according to various evaluation criteria, which are accuracy, time efficiency and less space. Accordingly, there are several classification algorithms such as Decision Tree Algorithms (Murthy,1998), Bayesian Algorithms (Cheeseman, 1983), Neural Networks algorithms (Park, Lee & Kim, 2004), K-Nearest Neighbor algorithms (Larose, 2005) and Support Vector Machine (Gunn, 1998).

2.6.1 Support Vector Machines

Support Vector Machines (SVM) was proposed for binary classification in 1970 and became popular in 1992. SVM is considered as linear classifier as it separates between the involved classes using linear model. The idea of SVM is to find an upright hyper-plane that maximize the separation between two classes of data (the training data). Furthermore, to split classes nonlinearly, Kernel functions with SVM can be used. SVM has a strict logical base, also it obtains an accurate classification output compared to other classification methods, particularly for data with multiple dimensions. It is probably the best-known classifier. SVM describes the input samples as points in the feature space and separates samples that belongs to different classes by a distinct gap that is as broad as possible. Examining cases are then outlined into the same range and their labels are assigned depends on which side of the gap they fit.

2.6.2 SVM Multi-class Classification

The basic Support Vector Machine is a binary classifier, where the class labels can be one of two values 1 or -1. However, there are numerous problems that have more than two classes such as, texture image classification and natural scene classification. So, SVM can be expanded to multi-class classification using two common methods: First approach is called (One Versus the Rest), by using binary SVM and divide the problem into various binary sub- problems (1-against-all). To build M-classifiers, C1, C2, . . ., CM, each class is modelled with a hyperplane that distinguish it from M-1 classes.

These sub-problems are merged to obtain a multi-class classification according to the highest output. Second approach is called (one versus one), in this approach for M classes, M(M−1)/2 hyperplanes are required (1-against-1). Each hyperplane split each class from another one and a max-wins voting scheme is used, in which the class with most votes is the class of the input data.

2.7 Summary

In this chapter, several studies that are related to Local Binary Pattern (LBP), were reviewed, described, analyzed, and summarized. This review included the advantages, limitations, objectives, and effectiveness of the LBP variations. Ojala (1996) proposed LBP, which have several advantages, such as its low computational complexity cost, its low sensitivity to illumination and its rotation invariant. However, LBP suffers from several disadvantages, such as its inability to noise resistance and its descriptor size, especially in large applications. In general, the existing approaches aim at either increasing the performance of the original LBP or reducing its size. The LBP extensions resulted in descriptors that are comparable to the original LBP.

Each of the discussed literature in this chapter achieved a certain goal, but as noted, literature is lacking a descriptor that combines the discriminative power, increasing performance and reducing the size of vector feature. Thus, there is a need to propose a descriptor with such properties.

Chapter Three

Proposed Work

3.1 Overview

This chapter presents the proposed work, which is developing a new descriptor for image classification by extending the original LBP with the aim at reducing the size of the extracted feature and enhance its discriminating power, which is called Diagonal Intersection LBP (DILBP). The organization of this chapter is as follows: Section 3.2 introduces the chapter and gives the motivations behind building the new descriptor. Section 3.3 presents and analyzes the proposed DILBP descriptor and its implementation steps. Section 3.4 shows the flowchart and constructs an algorithm in pseudocode. Section 3.5 analyzes the DILBP patterns. Section 3.6 discusses the DILBP characteristics. Finally, Section 3.7 gives a summary of this chapter.

3.2 Introduction

Image classification is a core process for a large number of computer vision applications, which require features that are extracted from the images, in addition to an algorithm to implements the classification task. Recently, extracting local features from images, such as LBP feature, were discussed extensively in the literature. As mentioned in Chapter Two, LBP extensions are generally divided into two categories, one investigated the reduction of the extracted feature size, while, the other investigated the enhancement of the discriminating power. This chapter introduces a new approach for local feature extraction by extending the LBP with the aim at reducing the size of the extracted feature and enhance the discriminating power by encoding and discriminating different patterns in the image. Yet, the extracted patterns are also related to those extracted in the original LBP to maintain the properties of the original LBP.

3.3 Diagonal Intersection LBP Descriptor

The proposed Diagonal Intersection Local Binary Pattern (DILBP), was named according to its implemented mechanism and the utilized neighborhood structure that forms of intersected diagonals. Images that are used in the proposed descriptor are a grayscale image, for instance, given a gray image W, with **n** rows and **m** columns of pixels, the pixel W (x, y), where $x<n$ and $y < m$, is given a code using the proposed descriptor.

The proposed descriptor does not consider the pixels that fall on the image border as in the original LBP, thus, the sum of the output codes calculated for each image is equal to ($n*m$ - the number of pixels on the border) codes.

The implementation steps of the proposed DILBP can be described as follows: First, the neighborhood structure is identified by a set of diagonals, where each diagonal pass through the center pixel. Each diagonal consists of three pixels, the center pixel, and a pair of neighbors, where each pixel of the neighborhood located on a different side of the center pixel.

Second, the intersect diagonals are used for calculating the sub-patterns around the center pixel. Third, the sub-patterns are aggregated to form the pattern encoded by the center pixel.

Finally, the patterns of all pixels in the image are aggregated as a final image descriptor, which is represented by a histogram. Figure 3.1 shows the neighborhood structure of the proposed DILBP.

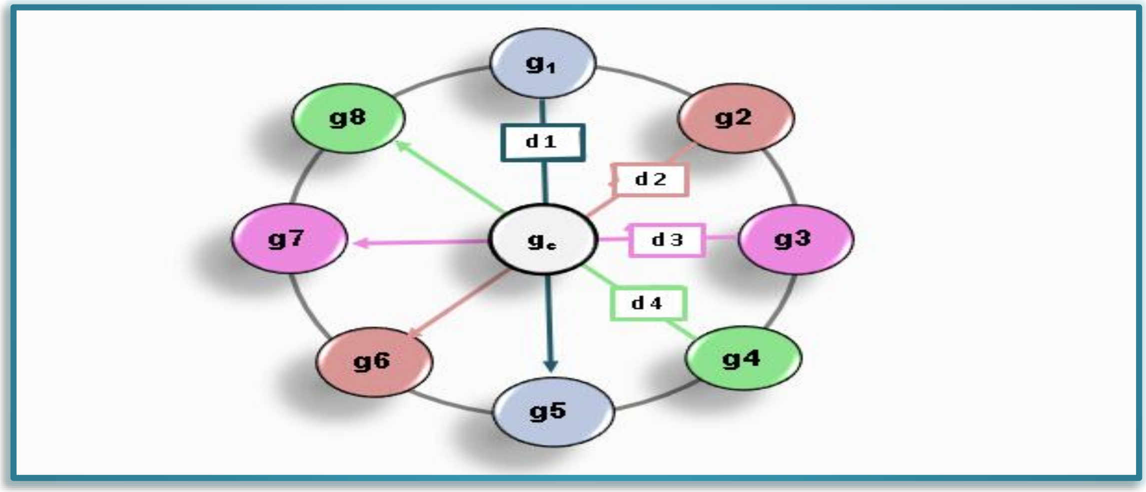


Figure 3.1: Neighborhood Structure of 8-Pixels in Proposed DILBP

3.3.1 DILBP Neighborhood Topology

The Diagonal Intersection LBP descriptor expressed using $\mathrm{DILBP}_{P,R}$ symbol, where R is referring to the radius of the neighborhood pixels around the center pixel and P is the number of sampling point surrounding the center pixel (neighborhood pixels). The neighborhood system used in the Diagonal Intersection LBP is the same as used in the original LBP, which is composed of the set $\{P1, P2, \ldots, Pn\}$. The center pixel is symbolized by the symbol c. R can be equal to one, two or more and the neighboring pixel can be two, three, etc. as shown in Figure 3.2.

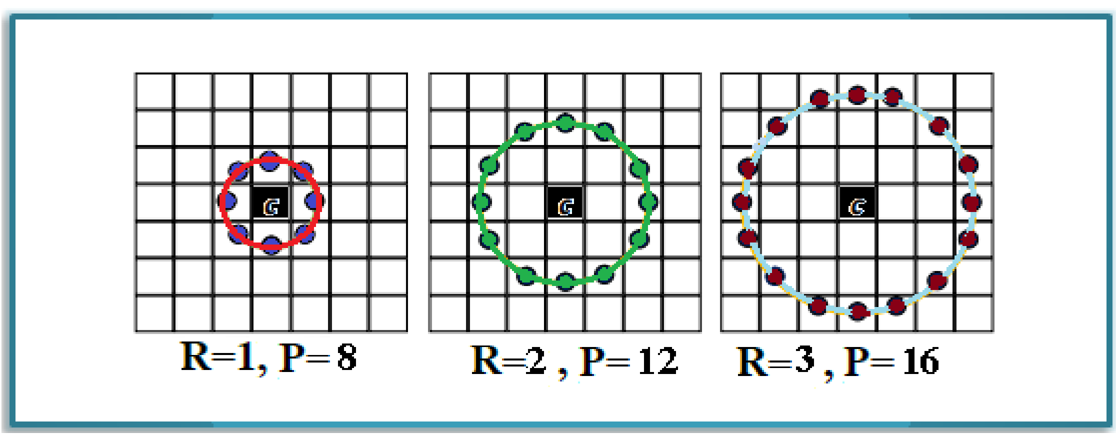


Figure 3.2: DILBP Neighborhood System

3.3.2 DILBP Codes Generation Process

The Diagonal Intersection LBP descriptor encodes the intensity variation for the neighborhood pixels around the center pixel in various orientations, by processing each diagonal independently. The pair of the pixels at each diagonal in the neighborhood is compared with the center pixel, as opposite to CS-LBP that did not compare the neighborhood pixels with the center. Each diagonal is given a unique sub-code. Consequently, compared to the original LBP, the length of the generated code in the proposed DILBP is decreased by the half, because the number of diagonals is half the number of the neighborhood pixels, which means that having P neighborhood pixels leads to generate a code of $P \,/\, 2$ in length.

As the neighborhood pixels of each center pixel is identified, the diagonals are identified and then processed. Accordingly, each diagonal that consists of three pixels, which are the center pixel and a pair of neighborhoods located on different sides of the center pixel, is processed independently. Thus, the pixel of the neighborhood on the left side is compared with the center pixel then the center pixels are compared with the neighborhood pixel on the other side. Calculating the differences and thresholding in the original LBP have been eliminated and replaced by one step implemented by **IF Statement** in the proposed DILBP, which is given in Equation 3.1.

$$\boldsymbol{s}(g_c) = \begin{cases} \mathbf{3}, & IF \quad g_i = g_c = g_{i+\frac{p}{2}} \\ \mathbf{2}, & IF \quad g_i \le g_c < g_{i+\frac{p}{2}} \; OR \; g_i < g_c \le g_{i+\frac{p}{2}} \\ \mathbf{1}, & IF \quad g_i \ge g_c > g_{i+\frac{p}{2}} \; OR \; g_i > g_c \ge g_{i+\frac{p}{2}} \\ \mathbf{0}, & Otherwise \end{cases} \tag{3.1}$$

According to Equation 3.1, the implemented function, $s(g_c)$, detects four different sub- patterns at each diagonal, which are: increasing intensity, decreasing intensity, indication of having a straight-line pass through the center pixel and insignificant change. The sub- pattern obtained in the cases of increasing and decreasing of intensity refers to an edge while the pattern obtained if the gray value for all pixels in the same diagonal is identical it represents a line. These sub-patterns, when aggregated to form a single pattern at each pixel, it encoded various significant image properties such as spots, edges, lines, flat areas, etc.

Figure 3.3 illustrates different patterns that can be captured by the proposed DILBP. In Figure 3.3(a), sub-pattern Pixels (2, 9 ,6), (3, 9, 7) and (4, 9, 8) form an increasing in intensity value, which finally forms a pattern that represents an edge. In Figure 3.3(b), sub- pattern Pixels (2, 9 ,6), (3, 9, 7) and (4, 9, 8) form a decreasing in intensity value, which finally form a pattern represent an edge. In Figure 3.3(c), sub-pattern Pixels (1, 9 ,5) form an identical intensity values, which form a pattern that represents a line. For Figure 3.3(d), referred to representation of an insignificant pattern.

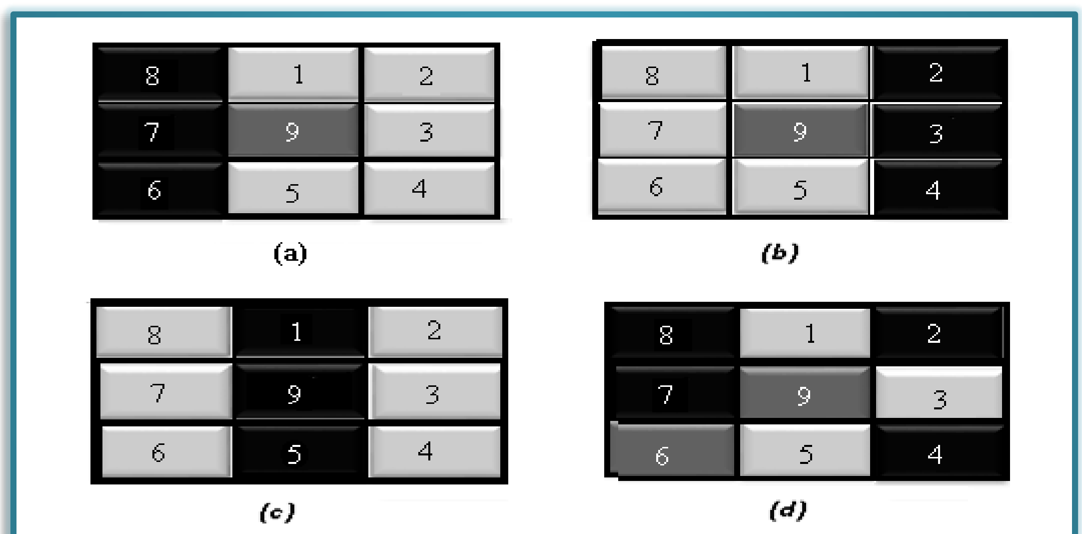


Figure 3.3. Example of DILBP Patterns: (a) Decrement Pattern (b) Increment Pattern (c) Line Pattern (d) Insignificant Pattern

3.3.3 Complexity Reduction of DILBP

According to Equation 3.1, each diagonal is given a single value among four values that are used for encoding, these values are 3,2,1 and 0. In the process of aggregating the sub- codes to form the final code for each center pixel, each diagonal value is added to the final code of the center pixel. Subsequently, at each pixel, there are huge possibilities of ($\mathbf{4^i}$), where 4, is the number of possibilities at each diagonal and **i** refer to the number of the diagonal. Thus, to decrease the complexity of the output quaternary code that is generated by the function, s(gc), these values are transformed into binary codes. Subsequently, the quaternary code is converted into two binary codes (the upper code and lower code) as the illustration in Figure 3.4 and Table 3.1, that shows conversion process to the binary code for the proposed DILBP.

Table 3.1: Converted DILBP Sub-Code to Upper and Lower Binary Code

DILBP Sub-Code	Binary Code	Upper Code	Lower Code
3	11	1	1
2	10	1	0
1	01	0	1
0	00	0	0

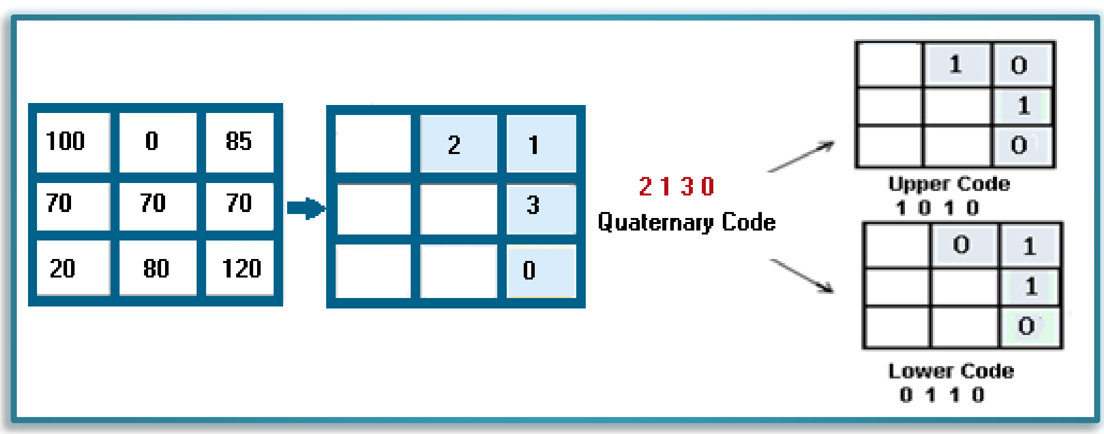


Figure 3.4: Example of Converting a Quaternary Code into Two Binary Codes

3.3.4 DILBP Weights Assignment

Given that the code at each diagonal is calculated and is converted into binary code of two bits, one bit is used to build one of the two binary codes at each center pixel. Therefore, for the diagonals {d_1, d_2, d_3, d_4}, there are series of bit values $\{s1, s2, s3, s4\}$, are aggregating to form a code. The aggregation process can be viewed as concatenation process in which each bit is placed next to the previous one. This binary code is then converted into a decimal value by multiplying each bit value with 2^i, where i is the index of the bit and ranged in between 0 to p/2-1. The process of converting binary code to decimal for the upper code and the lower code is given in Equation 3.2.

$$DILBP_{R,P}(x,y) = \sum_{i=1}^{i=\frac{p}{2}}[s\,(g_i\,,\ g_c\,, g_{i+\frac{p}{2}})]2^{i-1} \tag{3.2}$$

3.3.5 Final Descriptor Construction

The histogram construction is the last stage of feature extraction in DILBP, which result in generating the descriptor for the whole image. In the proposed DILBP, two codes (upper code and lower code) are extracted at each pixel, thus, two histograms are constructed, each of which corresponding to the two codes in the pixels.

The generated histograms consist of $2^{P/2}$ bins. For instance, if the neighborhood contains eight pixels there $2^{8/2}$, which is equal 16 bins for each histogram. Each histogram is constructed with the values acquired from the pixels in the image.

Subsequently, each pixel contributes to one bin in each histogram. As the histograms are constructed, these histograms can be combined to generate a descriptor of double size $2{*}2^{P/2}$ bins. Examples of individual and combined histograms are illustrated in Figure 3.5.

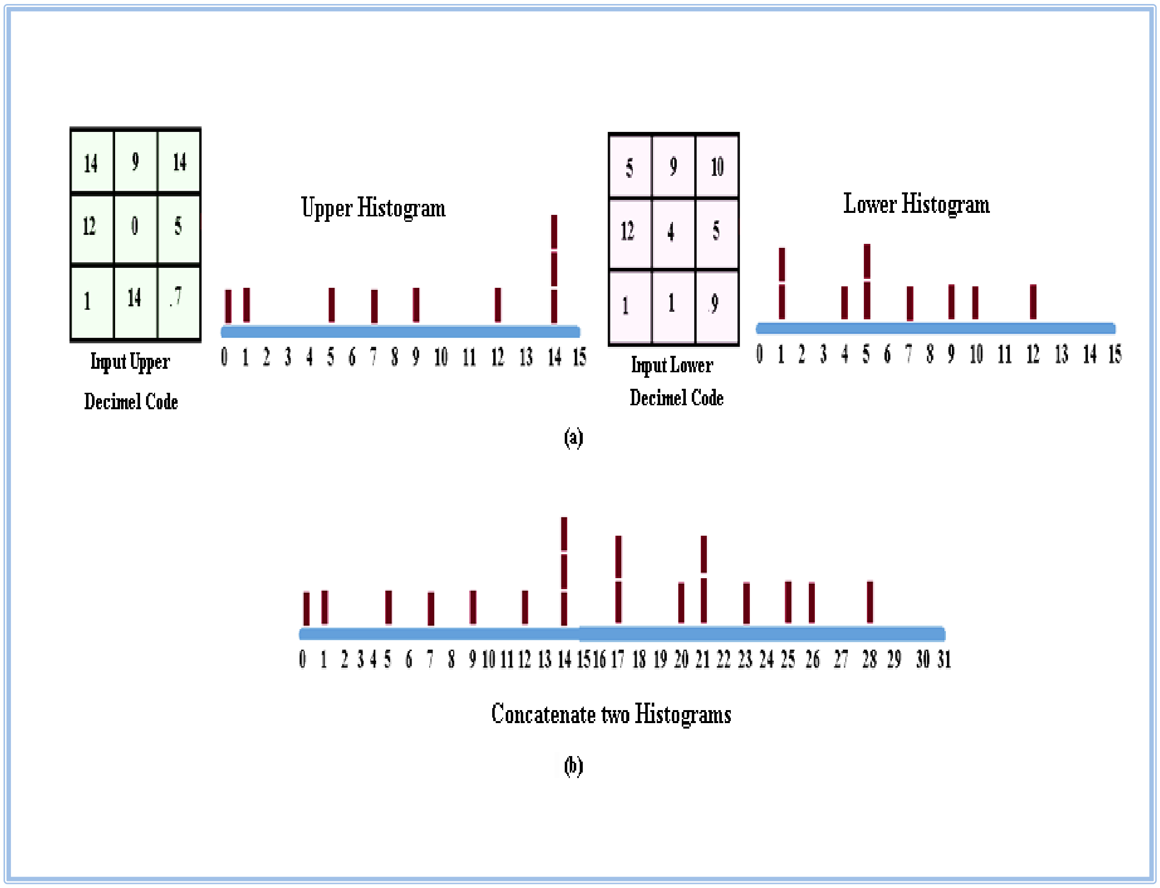


Figure 3.5: Example of Histograms (a) Histograms Construction for the Upper and Lower Pattern, (b) Two Histograms Concatenations

3.4 DILBP Pseudocode and Flowchart

The flowchart for the proposed descriptor (DILBP) is explained in detail as shown in Figure 3.6. The flowchart is starting from the top through input image and terminating in the down through getting a final descriptor.

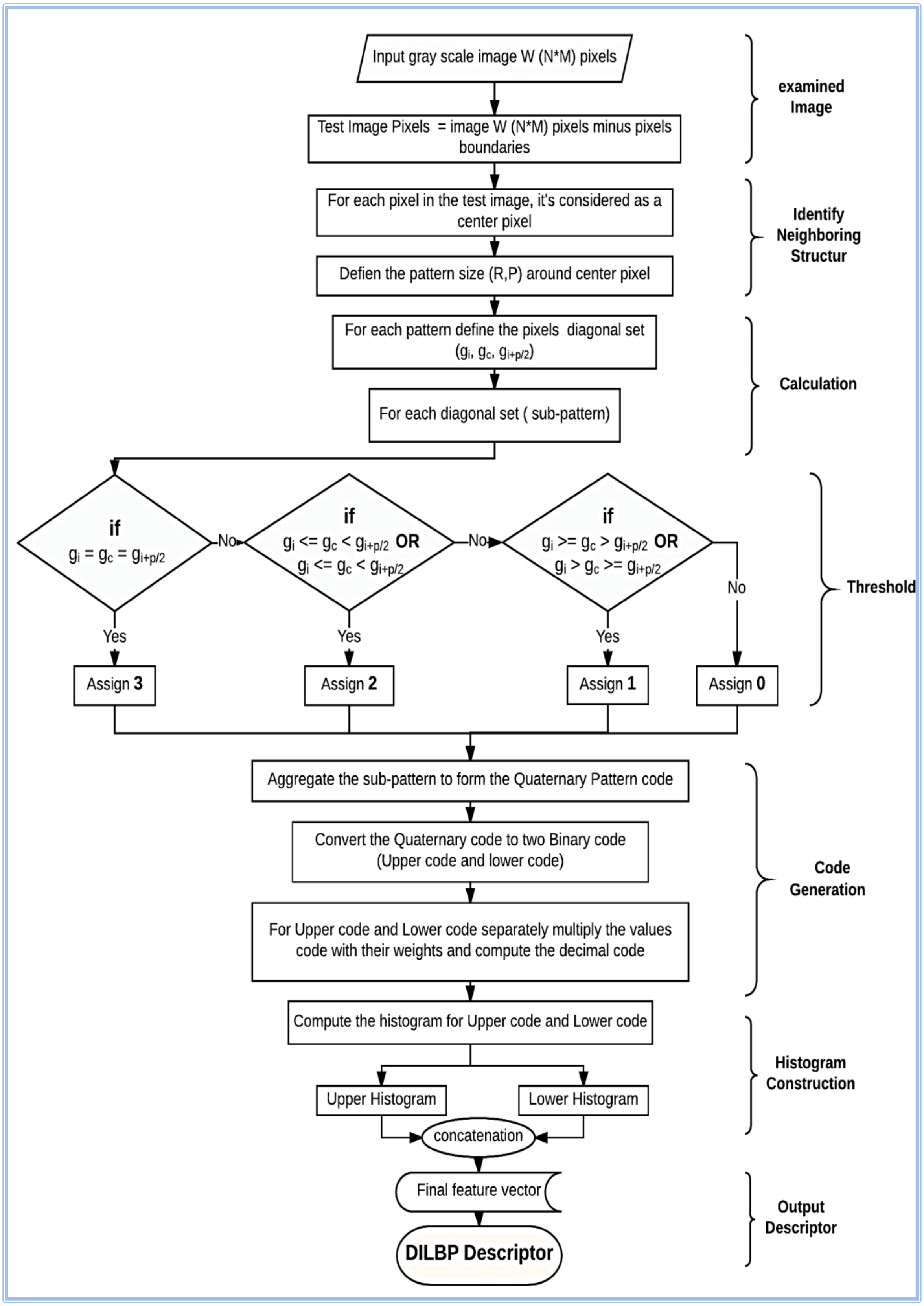


Figure 3.6: DILBP Flowchart Explain the Proposed DILBP Descriptor

The pseudocode of the DILBP algorithm, which used in the construction of the final description is given in Algorithm 1.

Algorithm 1: DILBP

Input: grayscale Image W (N*M) pixels

Output: DILBP histogram (Descriptor)

Test Image = Image W (N*M) pixels - pixels boundaries

for each Test Image **do**

define the descriptor patch size (**R, P**) and **c**

// R is the radius of patch, P is the neighborhood pixels, and c is the center pixel of patch

for every patch of Test image **do**

define the pixels diagonal set (**g i, g c, g i+p/2**)

for each diagonal set (sub-pattern) **do**

if $gi = g_c = gi_{+p/2;}$ Assign **3**

else if $g_i \leq g_c < g\ i_{+p/2}$ **OR** $gi < g_c \leq gi_{+p/2;}$ Assign **2**

else if $g_i \geq g_c > gi_{+p/2}$ **OR** $gi > g_c \geq gi_{+p/2;}$ Assign **1**

else Assign **0**

end if

Aggregate the sub-pattern to form the **Quaternary** Pattern code

end for

end for

for each Quaternary code **do**

Convert the Quaternary code to two Binary code (upper code, lower code)

```
 if sub-code of Quaternary code = = 3
     sub-code of upper code = =1; sub-code of lower code = = 1
 else if sub-code of Quaternary code = = 2
     sub-code of upper code = =1; sub-code of lower code = = 0
 else if sub- code of Quaternary code = = 1
     sub-code of upper code = =0; sub-code of lower code = = 1
 else if sub- code of Quaternary code = = 0
     sub-code of upper code = =0; sub-code of lower code = = 0
 for each Upper code do
   multiply the values code with their weights and compute the decimal
   code and compute the histogram
 for each lower code do
    multiply the values code with their weights and compute the decimal
    code and compute the histogram
   end for
 end for
end for
 Concatenate the histograms of upper histogram and lower histogram to form the
  final DILBP   histogram H
   H = Output descriptor
end for
return H
end
```

3.5 Generated Patterns using DILBP

The proposed DILBP descriptor encodes the intensity changes in the neighborhoods around each pixel in different orientations. The intensity changes that are captured by the proposed DILBP resulted in encoding significant image primitives such as (edges, corners, line- ends, lines, flat areas), which contributes to high discriminating power of the proposed descriptor. DILBP descriptor encodes various significant primitives, which can be summarized as follows:

1- **Line End**: texture primitive encoded by proposed DILBP, which represents a single spot with unique intensity that differs from the other pixels on the various diagonals, as illustrated in Figure 3.7.

2- **Corner**: primitive textures of various shapes, which represents spots with unique intensity that differs from the other pixels on the various diagonals, as illustrated in Figure 3.8.

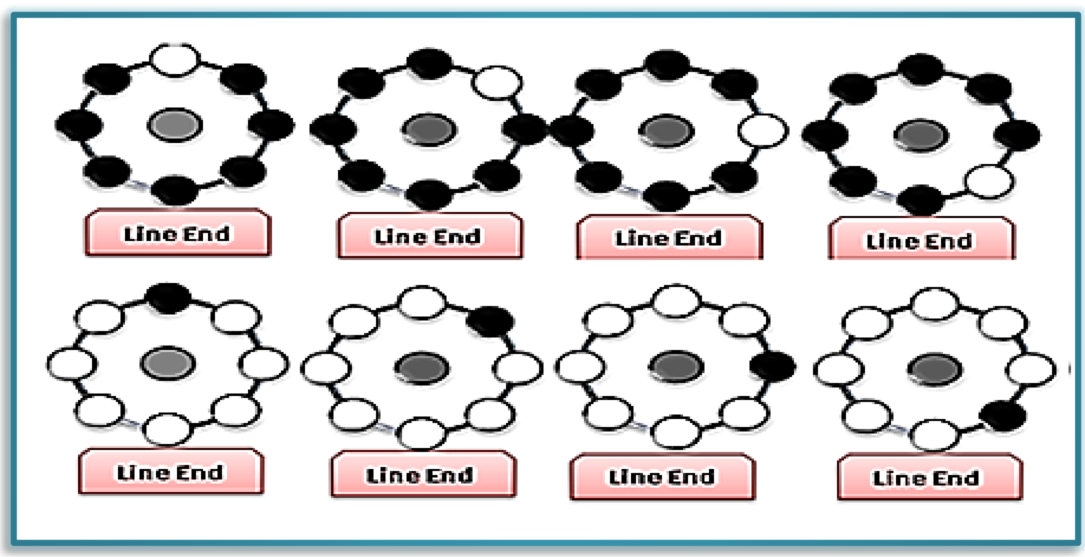


Figure 3.7: Example of Line-End Pattern That are Captured by DILBP Descriptor

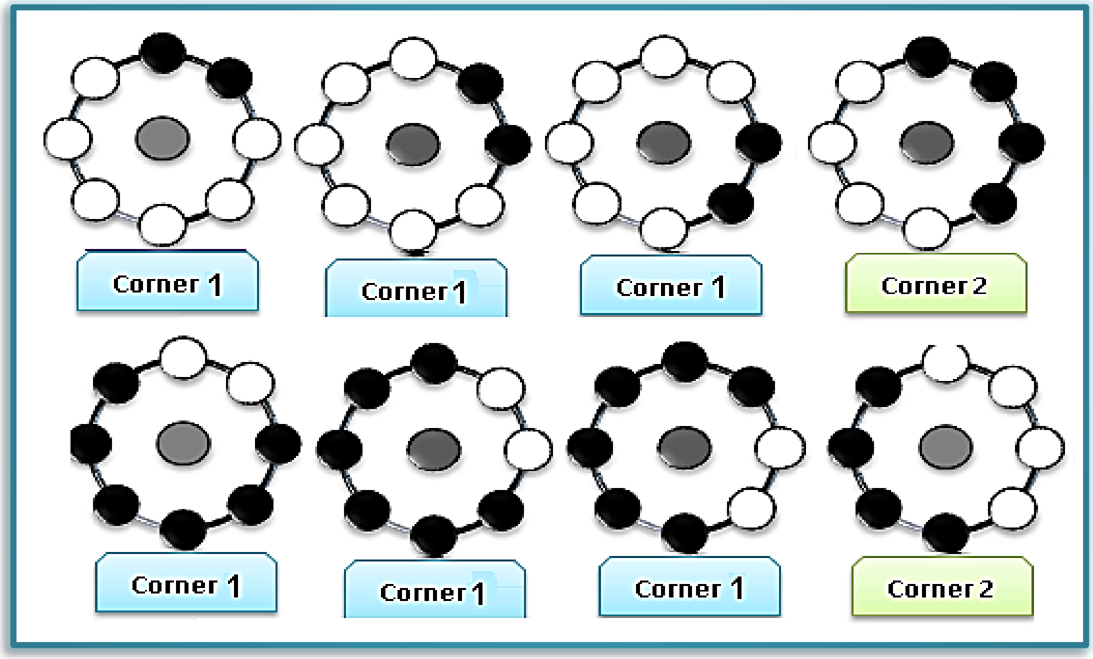


Figure 3.8: Example of Various Corner Shape Pattern That are Captured by DILBP Descriptor

3- **Edges**: edges are significant changes around a pixel with several shapes, as illustrated in Figure 3.9. In general, edges are described as collections of points in the image which has a high degree of gradient.

4- **Spot and Flat Area:** feature that is represented by white circles, which refer to a flat region, or black circles surrounding the center pixel, which refers to a spot. These encoded patterns are represented by the proposed descriptor as illustrated in Figure 3.10.

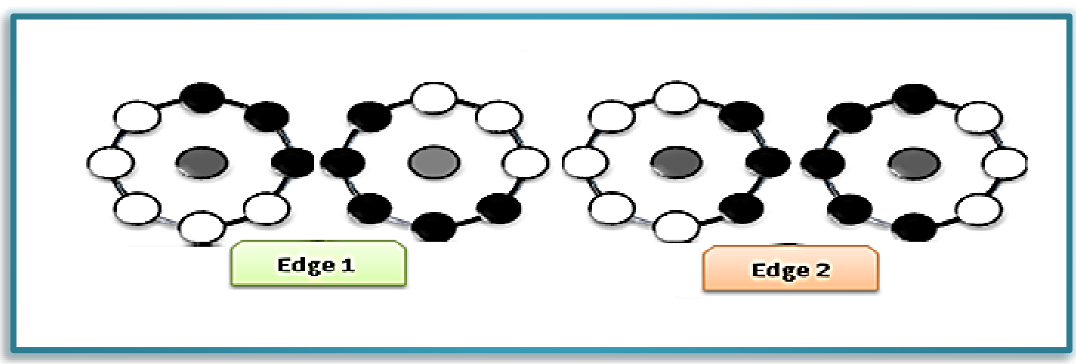


Figure 3.9: Example of Various Edges Pattern That are Captured by DILBP

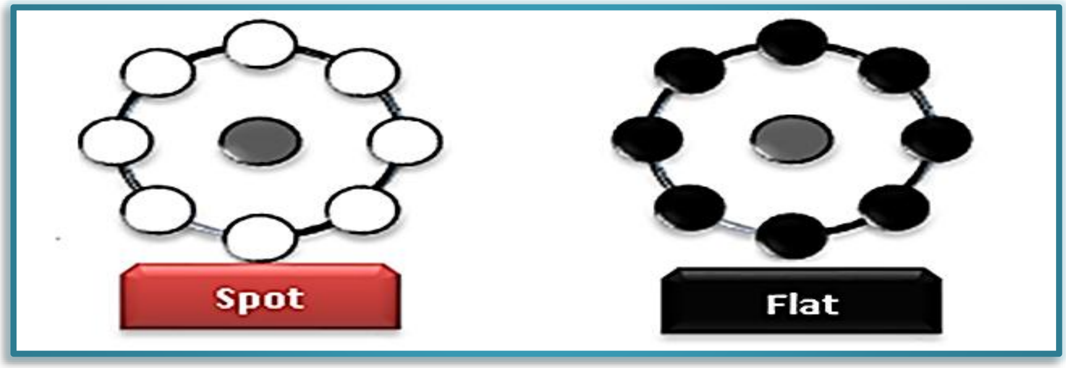


Figure 3.10: Example of Spot and Flat Pattern That are Captured by DILBP

5- Complete Line: the line that passes through the center pixel is an extra primitive, which is not explicitly captured by the original LBP, which are encoded in the proposed DILBP as illustrated in Figure 3.11.

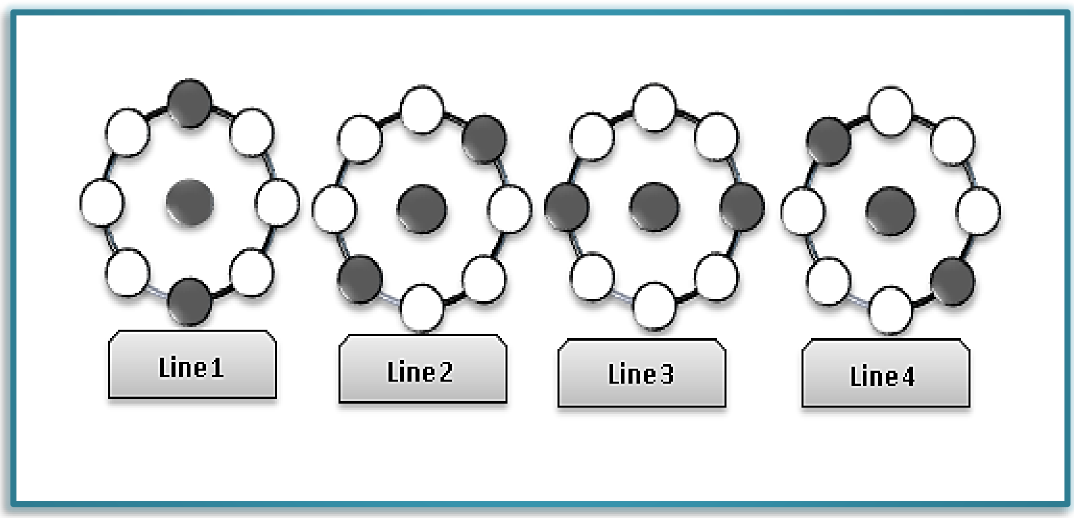


Figure 3.11: Example of Lines Pattern That are Captured by DILBP

3.6 DILBP Characteristics

The proposed DILBP descriptor have several characteristics that enable it to be considered with various image applications. These characteristics can be listed as follows:

1. The proposed DILBP descriptor is distinguished by its computational simplicity as in the original LBP descriptor.
2. The proposed DILBP descriptor is compact in size compared to the original LBP and other extensions of LBP in term of size. Subsequently, it enables low space requirement utilization.
3. The proposed DILBP descriptor encodes some important features of the image (such as edges, line-ends, corners, and complete lines).
4. The proposed DILBP descriptor is insensitivity to illumination changes, which is pointed to as invariant to the change in grayscale value. This is because the proposed DILBP descriptor does not refer to the exact grey value of a pixel, rather it considers the forms of texture features that are created by the relations of joint pixels in each diagonal. Consequently, a little modification of pixel's values does not influence as generally on the calculated pattern, as illustrated in Figure 3.12.

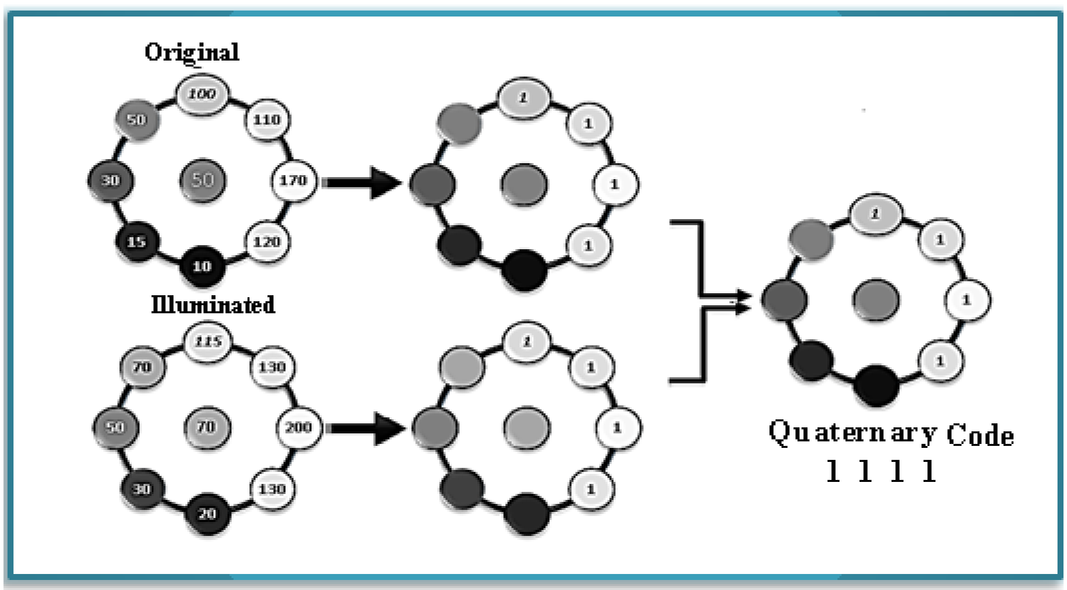


Figure 3.12: Illustration of DILBP's Illumination Insensitivity

5. The proposed DILBP descriptor has low sensitivity to noise, where the sensitivity to noise means that any slight change up or below in the value of neighbor pixels around the center pixel can cause a change in code of the pattern. The proposed DILBP descriptor works perfectly in the incrementing and decrementing grayscale cases, where any change in the pair of neighbor pixel around center pixel does not affect as generally on the calculated pattern. Figure 3.13 shows an example of DILBP low sensitivity to noise.

6. The proposed DILBP descriptor is rotation invariant. Generally, image rotation leads to rotate each neighborhood pixel g_p around the center pixel g_c. In the proposed DILBP descriptor, the pixel g_0, for example, which is located to the top of the center pixel g_c, is constantly selected to be (0, R). Each rotation will change the position of the pixel g_0 and the rest of the pixel, which logically will lead to generating different code. Patterns containing identical bit values (i.e.: zeros only or ones only) will remain as they are with whatever rotations implemented, while the rest of the codes will be altered with every rotation occurred to the image.

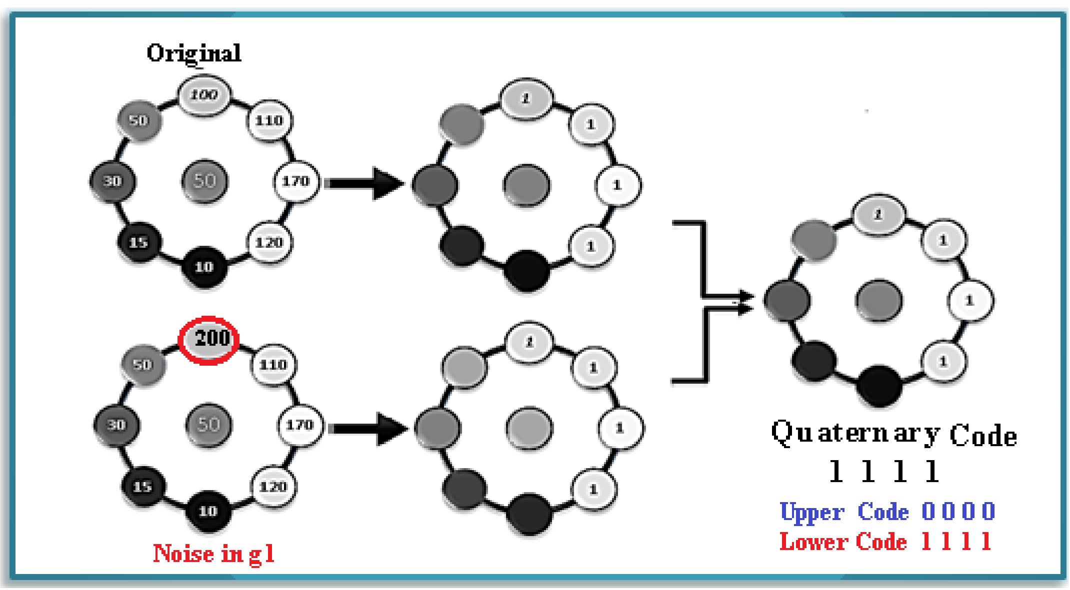


Figure 3.13: Illustration of DILBP Low Sensitivity to Noise

To make the proposed descriptor invariant to image rotation and generate identical code with any possible rotation in the image. A bit-wise shifting is implemented by shifting the bit to the right each time. Multiple bit-wise shifting is implemented for each generated code and the smallest possible value of code is selected to be the unique rotation code. As such, all similar patterns, yet rotationally different, will give an identical code by following the bit-wise shifting mechanism. The concept of rotation invariance used in the proposed DILBP descriptor is based on generating another code from the original code through converting value sub-code 2 to 1, value sub-code 1 to 2 and keeping the values sub-code 0 and 3 without change. Consequently, the two codes are concatenated together, and then applied the bit-wise shifting on the concatenation code as the original LBP to obtain the lowest value. Then the code with less value will be taken, then divided into two codes and pick out the left code, which is the unique rotation code as shown in Figure 3.14.

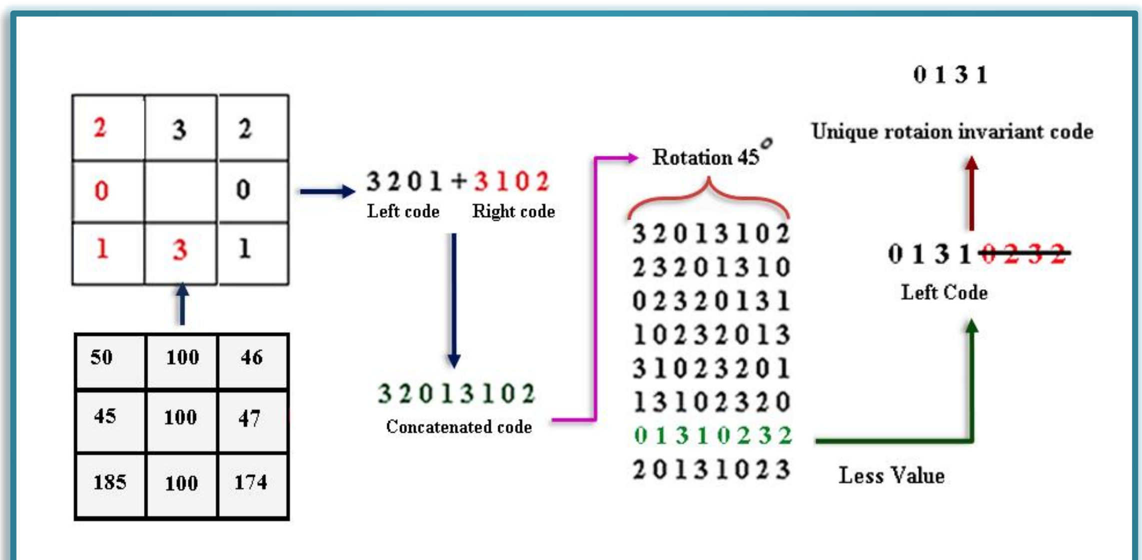


Figure 3.14: Illustration of DILBP's Rotation Invariance

7. The proposed DILBP could be merged with other LBP extensions. The proposed DILBP and the other extensions would produce various binary code for each pixel. Hence, DILBP code can be produced then concatenated with a code produced using other descriptors.

8. Some patterns that are considered as non-uniform and insignificant patterns in original LBP and subsequently ignored are considered as significant patterns and encoded discriminately in the proposed DILBP as illustrated in Figure 3.15.

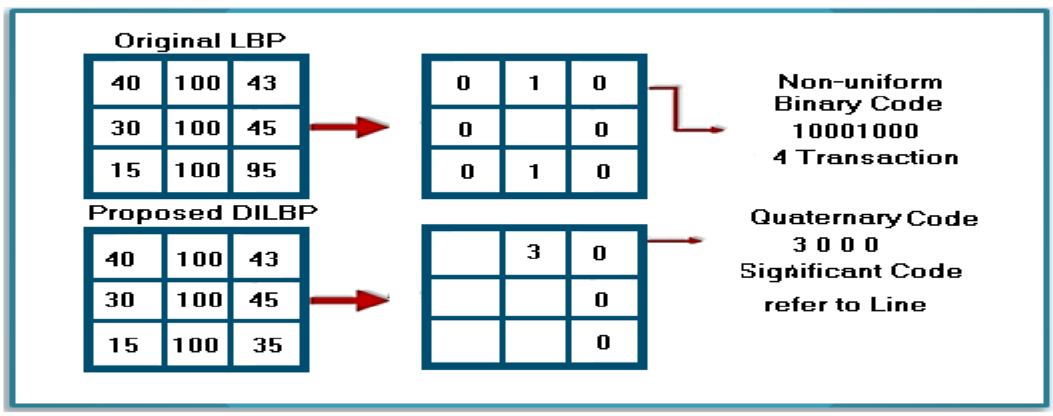


Figure 3.15: Illustration of Encoding Significant Non-Uniform Pattern Using DILBP Descriptor

3.7 Summary

In this chapter, an LBP-extension, called DILBP was proposed. DILBP captured the intensity variation for the neighborhood pixels around the center pixel in various diagonal orientations. This is implemented by processing each diagonal independently. Each pixel in the diagonal (the pair of neighbors) is compared with the center pixel and each diagonal is finally contributed to the final code with a single value. DILBP is shorter in length compared to the original LBP. Specifically, the proposed descriptor reduces the size to the half, which enhance utilization of resources and increase the speed of results. The proposed DILBP has obtained other advantages such as catch significant primitives, maintaining the image information, and insensitivity to change in gray scales (illumination).

Chapter Four

Implementation and Results

4.1 Overview

This chapter presents the experimental results of the proposed descriptor. This chapter is organized as follows: Section 4.2 gives an introduction to the experiments conducted. Section 4.3 presents the details of the datasets that are used in the implementation. Section 4.4 discusses the implementation details. Section 4.5 presents the parameter settings for the proposed and compared approaches. Section 4.6 presents the measurements that are used to evaluate the proposed descriptor. Section 4.7 presents the results of the implementation and shows the performance of the proposed descriptor. Finally, Section 4.8 gives a summary for this chapter.

4.2 Introduction

The proposed Diagonal Intersection LBP (DILBP) descriptor, which was presented in Chapter 3, is implemented using Java programming language. The implementation stages, to obtain and compare the results, are presented in this chapter. Besides the proposed DILBP, other LBP descriptors, which are the original LBP and CS-LBP which is the most like the proposed DILBP, will be evaluated and compared with the proposed descriptor. The performance evaluation of proposed and compared descriptors will be tested using the Fifteen Scene dataset (Lazebnik, Schmid, & Ponce, 2006), in addition to two common texture databases, these are: Outex datasets (Ojala et al. 2002) and the Columbia-Utrecht Reflection and Texture (CUReT) database (Dana et al. 1999). Support vector machine (SVM) multi- class classification is used for the classification task. Accuracy measure is used for performance rating and compares outcomes.

4.3 Datasets

This section describes the properties and lists some statistics about the utilized datasets.

4.3.1 Fifteen Scene Dataset

The proposed DILBP descriptor is verified, tested, and compared based on "Fifteen scene" dataset. The "Fifteen scene" dataset was progressively developed. The first eight essential categories (Coast, Forest, Highway, Inside City, Mountain, Open Country, Street, and Tall Building) were collected by (Oliva & Torralba, 2001), and the additional five categories (Bedroom, Suburb, Kitchen, Livingroom, Office) were added by (Fei-Fei & Perona, 2005), finally, another two additional categories (Store and Industrial) were submitted by (Lazebnik, Schmid, & Ponce, 2006).

Fifteen scene dataset contains 15 categories indoor and outdoor scenes. The number of images in each category is as follows: bedroom (216 images), CALsuburb (241 images), industrial (311 images), kitchen (210 images), livingroom (289 images), MITcoast (360 images), MITforest (328 images), MIThighway (260 images), MITinsidecity (308 images), MITmountain (374 images), MITopencountry (410 images), MITstreet (292 images), MITtallbuilding (356 images), PARoffice (215 images) and store (315 images).

The average size of each image in each category is around 270×250 pixels. Examples of these images are shown in Figure 4.1. For each category of Fifteen Scene dataset, the involved images are divided randomly into two sets: the training set, and testing set.

Figure 4.1: Examples of The Fifteen Scene Categories (Liu & Shah, 2007)

In the training set, half of the images are randomly selected for building classifier model, while the rest are combined for the testing. For instance, the industrial category has 311 images, 155 (311/2) for training and the rest for testing. Accordingly, Fifteen Scene dataset contains totally 4485 images, half of them used as train set (2240) and the rest (2245) used as test set.

4.3.2 Outex Dataset

The Outex dataset is composed of textural images that are captured for large collection of different materials. One of the common applications of the Outex dataset is texture classification, by which the algorithms, methods and approaches are tested for accuracy, rotation and illumination invariant. While Outex contains multiple datasets, the one with rotation and illumination variance is Outex TC 00012 (TC12).

Outex TC_00012 (TC12) dataset contains twenty-four classes of texture images, each of which contains twenty nonoverlapping images at the size of 128 x 128, which is captured at the 100-dpi spatial resolution and eight-bit intensity images. All images in each class were captured under three several illuminations (Inca, horizon, and t184) and rotated at nine angles (0°, 5°, 10°, 15°, 30°, 45°, 60°, 75°, and 90°) as shown in Figure 4.2. In Outex TC 12, there are 480 (24 x 20) images with illumination Inca and 0° angle, which are used in the training phase. While in testing phase there are 4320 (24 x 20x9) images with two illuminations: tl84 (sub- dataset 000) and horizon (sub-dataset 001), and with all rotation angles (nine angles) used as testing data.

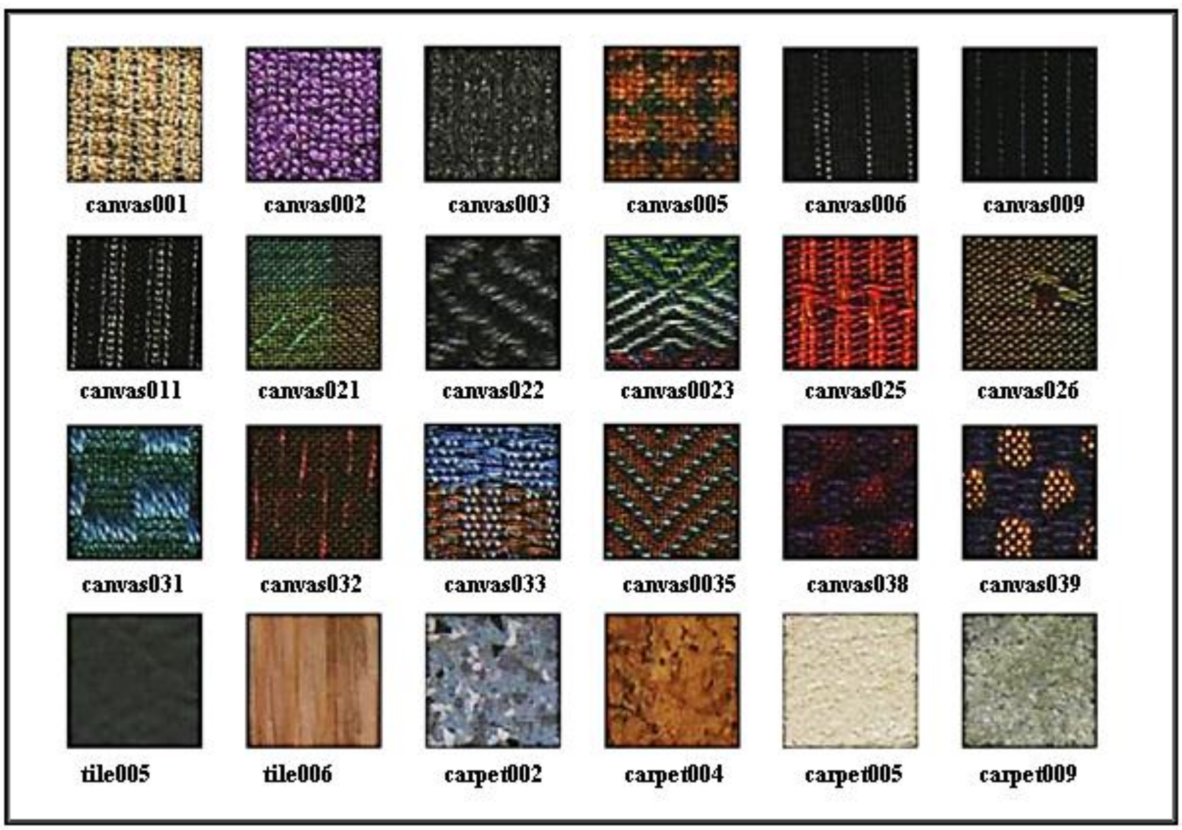


Figure 4.2: Examples of Different 24 Classes in Outex TC12 Dataset

Accordingly, there are 480 (24 × 20) training images and 4320 (24×20x9) testing images in both illumination variance (tl84 and horizon).

4.3.3 CUReT Dataset

The Columbia-Utrecht Reflectance and Texture dataset (CUReT) includes 61 various textures classes, as shown in Figure 4.3. Each class of CUReT dataset is composed of 205 original images that represent physical textures that are acquired at different angles and illumination (Dana, 1999). A number of 92 images are chosen from the original images with an angle less than 60^o and a clearly defined region while the rest is discarded. Area of 200 ×200 in size is cropped from the chosen images and the rest of the image are discarded. Finally, the resulted images are transformed into a greyscale.

Figure 4.3: The 61 Classes in CUReT Dataset (Varma & Zisserman 2009)

Accordingly, the total number of the image in the dataset is 5612 (61 x 92), where the training images (T) are randomly selected from each class whereas the remaining (92-T) images for each class is used for testing. In this thesis, half of the images are taken as a training and the rest as a testing, two folds random splits are executed separately, then the average of accuracy classification is used to avoid a bias. The main thing that distinguishes this dataset is that it has large number of classes. Hence, it possibly contains a small interclass similarity in the feature space.

4.4 Implementation

The proposed and compared descriptors, besides the classification method, which is used for comparison purpose, are implemented using Java programming language. The flowchart for the implementation process is illustrated in Figure 4.4.

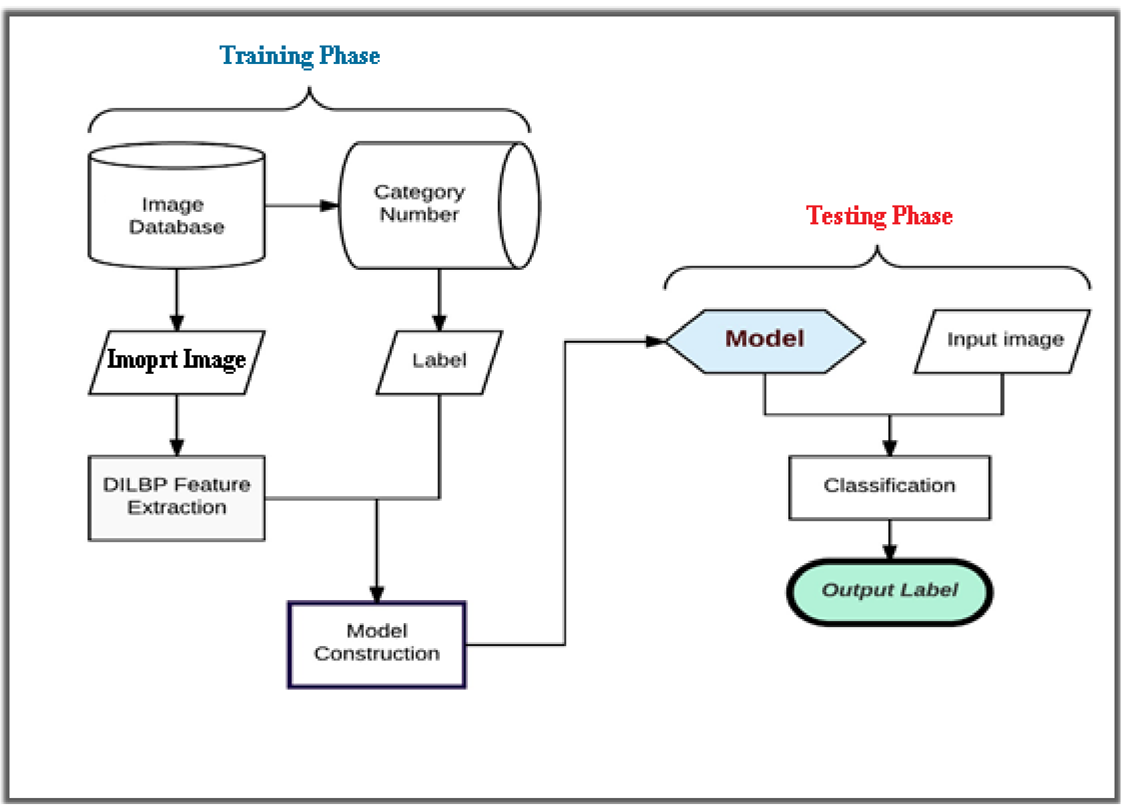


Figure 4.4: Implementation Flowchart for Proposed DILBP

4.5 Parameter Setting

The implementation needs a collection of parameters that require setting for Fifteen Scene dataset, Outex TC12 dataset and CUReT dataset. The same settings are applied to the proposed and compared descriptors, as listed in Table 4.1.

Table 4.1: Parameter Settings of Fifteen Scene, Outex TC12 and CUReT Datasets

Parameter	DATASETS		
	Fifteen Scene	Outex TC12	CUReT
Number of classes	15 classes	24 classes	61 classes
Image Sizes	270×250	128x128	200x200
Pixel format	8-bit	8-bit	8-bit
Descriptor size	P=8, R=1	P=8, R=1	P=8, R=1
Histogram Representation	Normalize [0~1]		
Classifier	SVM multi-class	SVM multi-class	SVM multi-class
Rotation Invariant	Yes/No	Yes/No	Yes/No

As noted in Table 4.1, images in Fifteen Scene dataset have average size of 270×250 with 8-bit. In Outex TC 12, images have equal size of 128x128 with 8-bit. In CUReT images have equal size of 200x200 with 8-bit. The Normalize histogram is used to convert the integer values to the range of [0~1], by dividing each histogram bin value over the total sum of all bins. Finally, each image will have equal size feature vector that will be inserted into the SVM classifier. For classification, multi-classes are used, with number of classes equal to the number of categories.

4.6 Evaluation Metrics

Classification Accuracy, which is refer to the ability of the algorithm to predict the correct class label for instances of unknown class labels (testing set), is calculated as given in Equation 4.1. Accuracy measure is used for evaluating and comparing between the underlying descriptors.

$$Classification\ Accuracy = \frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ test\ sapmles} \times 100 \tag{4.1}$$

4.7 Results

In this section, the results of the proposed DILBP and the other comparison methods (LBP and CS-LBP) are presented. The results were categorized into two classes:

1. The results of the basic descriptors without additional modification (without implementing rotation invariant), which is referred to as Basic mode.

2. The results of rotation invariance descriptors, by implementing an extra process for the proposed and compared descriptors to be rotation invariant, which is referred to as Rotation Invariant mode.

4.7.1 Results of Fifteen Scene Dataset

Figure 4.5 illustrates the classification accuracy of the compared methods (original LBP and CS-LBP) and proposed DILBP descriptors over Fifteen Scene Dataset in Basic mode and in Rotation Invariant mode.

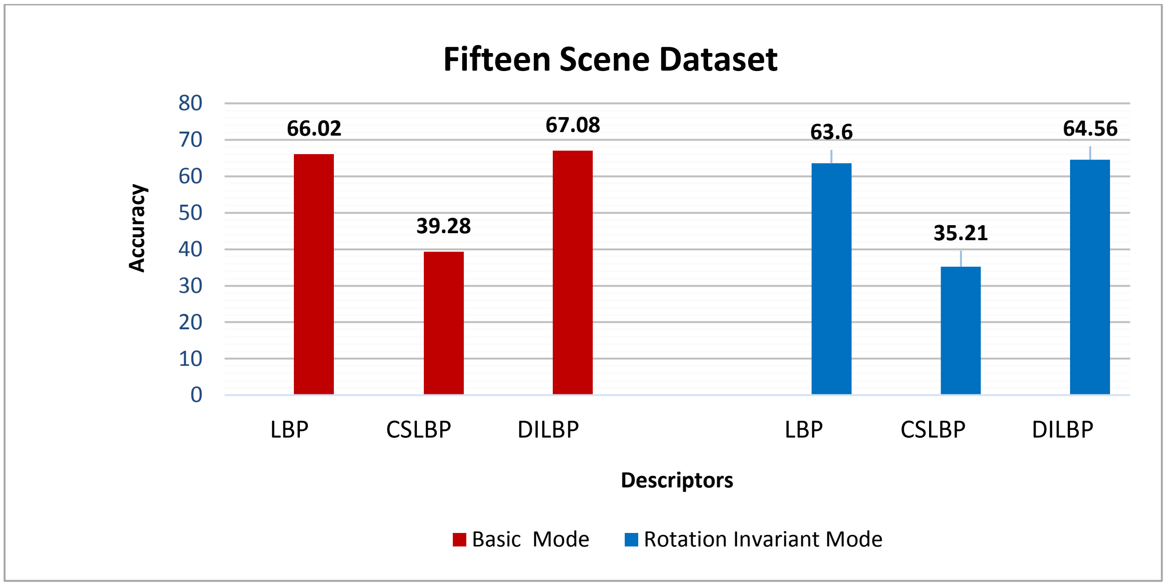


Figure 4.5: Accuracy Results of Descriptors in Basic and Rotation Invariant Modes Over Fifteen Scene Dataset

As noted in Figure 4.5, the Proposed DILBP has obtained better accuracy compared to the original LBP and CS-LBP in the Basic mode. The best result for image classification was for Proposed DILBP with an accuracy of 67.08%, the second rank was for the original LBP with accuracy of 66.02%. While, the worst result was obtained by CS-LBP with accuracy of 39.28%. The accuracy enhancement of DILBP descriptor compared to the original LBP is around 1.5% and the enhancement over the CS-LBP, which has almost similar size to proposed DILBP, is around 41.4 %. The improved results of DILBP can be justified due to the additional primitives (e.g.: lines) that DILBP addresses. These results satisfy the third objective of this thesis which aims at enhancing image classification.

In Rotation Invariant mode, the classification accuracy of the proposed DILBP, original LBP, and CS-LBP are fairly similar to the basic mode. In general, the results of the proposed descriptor in Rotation Invariant mode for natural scene images classification is satisfactory. The proposed DILBP has the best results with accuracy of 64.56%. the second rank for the original LBP with accuracy of 63.60%. While the worst result was obtained by CS-LBP with accuracy of 35.21%. The accuracy enhancement of proposed DILBP descriptor compared to the original LBP is around 1.5% and the enhancement over the CS-LBP is around 45.4%. The improved results of the proposed DILBP compared with the original LBP and CS-LBP, again can be justified with the presence of extra captured primitives. These results confirm that the proposed DILBP is rotation invariant, accordingly, satisfies the second objective of this thesis which is related to the rotation invariance and third objective, which is related to enhancing image classification.

Table 4.2 lists the values of the classification accuracy of the compared and proposed descriptors. As noted, the proposed DILBP gives the best results, whereas, the original LBP is the second best and finally, the CS-LBP is the worst in both Basic mode and Rotation invariant mode. Accordingly, using the proposed DILBP is considered the best for classifying natural images.

Table 4.2: Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes of Fifteen Scene Dataset

Descriptor Mode	Descriptor	Descriptor Size	Accuracy
Basic	LBP	256	66.02
	CS-LBP	16	39.28
	Proposed DILBP	32(16*2)	**67.08**
Rotation Invariant	LBP	36	63.6
	CS-LBP	6	35.21
	Proposed DILBP	12(6*2)	**64.56**

4.7.2 Results of Outex TC 12 Dataset

Figure 4.6 illustrates the accuracy results of the proposed DILBP, original LBP, and CS-LBP over Outex TC 12 dataset, in Basic mode and Rotation Invariant mode.

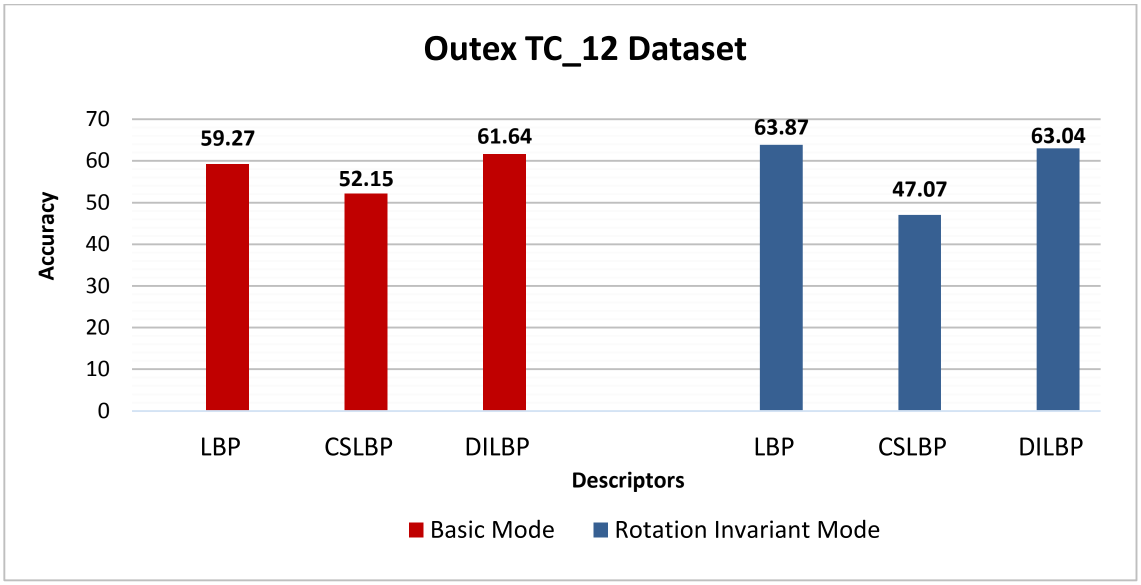


Figure 4.6: Accuracy Results of the Compared Descriptors in Basic and Rotation Invariant Modes Over Outex Tc12 Dataset

As noted in Figure 4.6, the Proposed DILBP has obtained better accuracy compared to the original LBP and CS-LBP in the Basic mode. The best result for image classification was for proposed DILBP with accuracy of 61.64%, the second rank for the original LBP with accuracy of 59.27%. While the worst result was obtained by CS-LBP with accuracy of 52.15 %. The accuracy enhancement of DILBP descriptor compared to the original LBP is around 3.8% and the enhancement over the CS-LBP is around 15.3%. The improved results of DILBP are attributed to the additional primitive (e.g.: line) and resistance to change in illumination. This confirms that proposed DILBP is illumination invariant, which satisfies the second objective, which is related to the illumination invariance and third objective, which is related to enhancing image classification. In Rotation Invariant mode, the original LBP has the best results with accuracy of 63.87%. the second rank for the proposed DILBP with accuracy of 63.04%. While, the worst result was obtained by CS-LBP with accuracy of 47.07%. The accuracy enhancement of proposed DILBP descriptor compared to the CS-LBP is around 25.3%. On the other hand, DILBP accuracy is lower than the original LBP at a decreased rate of 1.2%. The results of proposed DILBP and original LBP are very close even though the proposed DILBP uses 32 feature vectors compared to 256 feature vectors to original LBP. This result satisfies the first objective, which is related to reducing size of feature without degrading the discriminatory ability.

Table 4.3 lists the values of the classification accuracy of the compared and proposed descriptors. As noted, the proposed DILBP gives the best results, whereas, the original LBP is the second best and finally, the CS-LBP is the worst in Basic mode. While the original LBP gives the best results in Rotation Invariant mode, the proposed DILBP is the second best and finally, the CS-LBP is the worst.

Table 4.3: Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes of Outex Tc12 Dataset

Descriptor Mode	Descriptor	Descriptor Size	Accuracy
Basic	LBP	256	59.27
	CS-LBP	16	52.15
	Proposed DILBP	32(16*2)	**61.64**
Rotation Invariant	**LBP**	36	**63.87**
	CS-LBP	6	47.07
	Proposed DILBP	12(6*2)	63.04

4.7.3 Results of CUReT Dataset

Figure 4.7 illustrates the accuracy results of the proposed DILBP, original LBP, and CSLBP over CUReT dataset in Basic mode and Rotation Invariant mode.

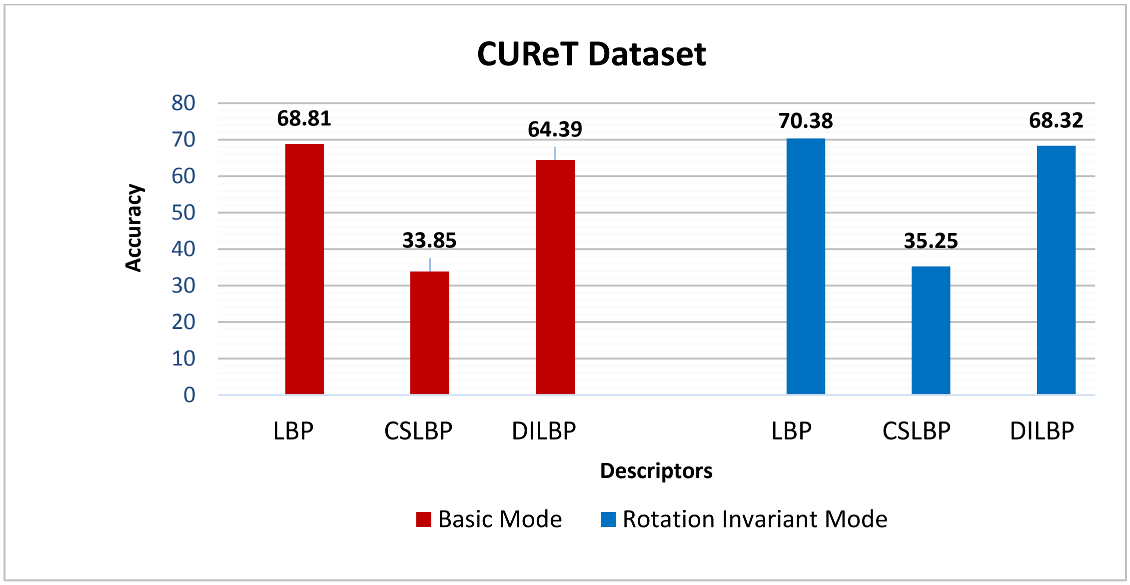


Figure 4.7: Accuracy Results of the Compared Descriptors in Basic and Rotation Invariant Modes Over CUReT Dataset

As noted in Figure 4.7, the original LBP obtained better accuracy compared to the proposed DILBP and CS-LBP in the Basic mode. The best result for image classification was for original LBP with accuracy of 68.81%, the second rank for the proposed DILBP with accuracy of 64.39%. While the worst result was obtained by CS-LBP with accuracy of 33.85%. The accuracy enhancement of DILBP descriptor over the CS-LBP, which has almost similar size to proposed DILBP, is around 47.4%. On the other hand, proposed DILBP accuracy is lower than the original LBP at a decreased rate of 6.4%.

In Rotation Invariant mode, the classification accuracy of the original LBP has the best results with accuracy of 70.38%. the second rank for the proposed DILBP with accuracy of 68.32%. While the worst result was obtained by CS-LBP with accuracy of 35.25%.

The accuracy enhancement of proposed DILBP descriptor compared to the CS-LBP is around 48.4%. On the other hand, DILBP accuracy is lower than the LBP at a decreased rate of 2.9%. The improved results of the proposed DILBP compared with the CS-LBP can be referred to the additional primitive that is captured in the proposed DILBP, while less accuracy results compared with original LBP is due to the long code that is generated from original LBP.

Table 4.4 lists the values of the classification accuracy of the compared and proposed descriptors. As noted, the original LBP gives the best results, whereas, the proposed DILBP is the second best and finally, the CS-LBP is the worst in Basic mode and Rotation Invariant mode.

Table 4.4: Accuracy Results of the Proposed and Compared Descriptors in Basic mode and Rotation Invariant Modes

Descriptor Mode	Descriptor	Descriptor Size	Accuracy
Basic	**LBP**	256	**68.81**
	CS-LBP	16	33.85
	Proposed DILBP	32(16*2)	64.39
Rotation Invariant	**LBP**	36	**70.38**
	CS-LBP	6	35.25
	Proposed DILBP	12(6*2)	68.32

4.8 Summary

This chapter presents the implementation of the proposed DILBP using Java programming Language over several datasets to evaluate the classification accuracy of the proposed DILBP. Two different LBP descriptors, which are LBP and CS-LBP are used for comparison purposes. The empirical results of the proposed DILBP demonstrate that its accuracy is satisfying in natural scene image classification.

In Fifteen Scene dataset, the proposed DILBP achieved the best accuracy results, in Basic mode, with increased accuracy rate of 1.5 % compared to the original LBP, and with CS-LBP with fairly similar size at an increased rate of 41.4 %. In Rotation Invariant mode, the proposed DILBP achieved the best results at an increased rate of 1.5% compared to original LBP and at an increased rate of 45.4% compared to CS-LBP.

In Outex TC12 dataset, the proposed DILBP achieved the best results at an increased rate of 3.8% compared to original LBP and at an increased rate of 15.3% compared to CS-LBP in Basic mode, while in Rotation Invariant mode, the result of the proposed DILBP is lower than the original LBP at a decreased rate of 1.2%, but at an increased rate of 25.3% compared to CS-LBP.

In CUReT dataset, the result of the proposed DILBP is lower than the original LBP at a decreased rate of 6.4%, but at an increased rate of 47.4% compared to CS-LBP in Basic mode. In Rotation Invariant mode, the result of the proposed DILBP is lower than original LBP at a decreased rate of 2.9%, but at an increased rate of 48.4% compared to CS-LBP. This is because the length of a feature vector has been reduced, which leads to losing some accuracy which is considered as a trivial. The proposed DILBP could be utilized in many large dataset applications and real-time applications, where the small feature size considered as a significant issue such as human- computer interaction and face authentication.

Chapter Five

Conclusion and Future Work

5.1 Conclusion

In this thesis, a new LBP-based descriptor that is used for image classification is proposed, which is aimed at reducing the size of the extracted feature and enhance the discriminating power, the new descriptor is called Diagonal Intersection LBP (DILBP). DILBP descriptor is applied to every pixel independently, like the original LBP, to generate a pattern for each pixel in the image and then aggregate these patterns to form the final image descriptor. DILBP is developed in a set of processes, as described in the following: First, the neighborhood structure identified by a set of diagonals, where each diagonal pass through the center pixel. Each diagonal consists of three pixels, the center pixel and a pair of neighbors, where each pixel of the pair of neighbors located on a different side of the center pixel. Second, the intersect diagonals are used for calculating the sub-patterns around the center pixel. Third, the sub-patterns are aggregated to form a single pattern. Finally, the patterns of all pixels in the image are aggregated in a histogram, which represents the final image descriptor.

The contributions of this thesis are as follows:

1. Reduce the size of the proposed descriptor by comparing and analyzing a set of pixels rather than an individual pixel to produce the sub-patterns. Decreasing the length of the feature vector is done through reducing the size of a pattern from 8-bits code to 4-bits code, which leads to reducing the size of the histogram bins from 256 to 16 for each upper code and lower code, which results in decreasing the storage requirements.

2. The proposed DILBP descriptor achieved the rotation and illumination invariance, where invariance in rotation and illumination is a fundamental aspect to enhance the accuracy of image applications such as image classification.

3. Use the proposed DILBP to enhance image classification, where the proposed DILBP descriptor is efficient for extraction primitive image properties, such as edges, corners, flat areas, and lines-ends, and these primitives are essential in image classification. In addition, the proposed DILBP captures extra primitives (e.g. Lines). Natural image scene (Fifteen Scene dataset) and texture images (Outex dataset and CUReT dataset) have been used with multi-class SVM classifier in the experiments to prove the ability of DILBP in encoding these primitives.

4. Measuring the discriminative ability of the proposed DILBP and comparing its classification accuracy with accuracy of original LBP and CS-LBP by using public image datasets. The experimental results over the Fifteen Scene dataset showed that the accuracy enhancement of proposed DILBP descriptor compared to the original LBP is around 1.5% and the enhancement over the CS-LBP is around 45.4%. In Outex TC 12 dataset, the accuracy enhancement of proposed DILBP descriptor compared to the original LBP is around 3.8% and the enhancement over the CS-LBP is around 15.3%. while in CUReT dataset, the accuracy enhancement of proposed DILBP descriptor compared to the CS-LBP is around 48.4% but is lower than the original LBP at a decreased rate of 2.9%, although proposed DILBP has the size of 16 (4-bits) for upper and lower code compared to the original LBP descriptor, which has the size of 256 (8-bits).

5.2 Future Work

The future and suggested works for the proposed descriptor in this thesis are as follows:

- The proposed DILBP descriptor can be expanded to classify different image modalities or more complex images, such as medical images and remote sensing images.
- The proposed DILBP descriptor implemented on greyscale images only. Therefore, converting the color image to a grayscale image may resulted in some loss of information. So, the proposed LBP descriptor can experiment with color images datasets, through handling three channels (RGB) independently.
- Implement the proposed DILBP descriptor with different multi-class classifiers to investigate enhancing the classification accuracy.
- The proposed DILBP descriptor can be affected by noise. A potential enhancement to the proposed descriptor is to be combined with other LBP extension, which is robust to noise such as Local Ternary Pattern (LTP).

References

Ahonen, T., & Pietikäinen, M. (2007). Soft histograms for local binary patterns. In *Proceedings of the Finnish signal processing symposium, FINSIG* (Vol. 5, No. 9, p. 1).

Ahonen, T., Hadid, A., & Pietikäinen, M. (2004). Face recognition with local binary patterns. Computer vision-eccv 2004, 469-481.

Aid, P. F. (n.d.). Photo Restoration, Audio & Video Transfer Services. Retrieved June 07, 2017, from http://photofirstaid.co.uk/.

Caputo, B. (2014). Medical image annotation. In Multimodal Interactive Systems Management (pp. 205-217). EPFL Press.

Cheeseman, P. (1983). A method of computing generalized Bayesian probability values for expert systems. In *Proceedings of the Eighth international joint conference on Artificial Intelligence-Volume 1* (pp. 198-202). Morgan Kaufmann Publishers Inc.

Crnojević, V., Panić, M., Brkljač, B., Ćulibrk, D., Ačanski, J., & Vujić, A. (2014). Image processing method for automatic discrimination of hoverfly species. *Mathematical Problems in Engineering*, *2014*.

Dana, K. J., Van Ginneken, B., Nayar, S. K., & Koenderink, J. J. (1999). Reflectance and texture of real-world surfaces. *ACM Transactions on Graphics (TOG)*, *18*(1), 1-34.

Despotović, I., Goossens, B., & Philips, W. (2015). MRI segmentation of the human brain: challenges, methods, and applications. *Computational and mathematical methods in medicine*, *2015*.

Doshi, N. P. (2014). *Multi-dimensional local binary pattern texture descriptors and their application for medical image analysis* (Doctoral dissertation, © Niraj P. Doshi).

Fan, J., Yau, D., Elmagarmid, A., & Aref, W. (2001). Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE Transactions on Image Processing, 10(10), 1454-1466. doi:10.1109/83.951532.

Fei-Fei, L., & Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 2, pp. 524-531). IEEE.

Fu, X., & Wei, W. (2008). Centralized binary patterns embedded with image Euclidean distance for facial expression recognition. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on* (Vol. 4, pp. 115-119). IEEE.

Ghatge, M. S., & Dixit, V. V. (2013). Face Recognition under varying illumination with Local binary pattern. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 2(2).

Gunn, S. R. (1998). Support vector machines for classification and regression. *ISIS technical report*, *14*, 85-86.

Guo, Z., Zhang, L., & Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, *19*(6), 1657-1663.

Hafiane, A., Seetharaman, G., Palaniappan, K., & Zavidovique, B. (2008). Rotationally invariant hashing of median binary patterns for texture classification. *Image Analysis and Recognition*, 619-629.

Heikkila, M., & Pietikainen, M. (2006). A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, *28*(4), 657-662.

Heikkilä, M., Pietikäinen, M., & Schmid, C. (2006). Description of interest regions with center-symmetric local binary patterns. In *Computer vision, graphics and image processing* (pp. 58-69). Springer Berlin Heidelberg.

Heikkilä, M., Pietikäinen, M., & Schmid, C. (2009). Description of interest regions with local binary patterns. Pattern recognition, 42(3), 425-436.

Huang, D., Shan, C., Ardabilian, M., Wang, Y., & Chen, L. (2011). Local binary patterns and its application to facial image analysis: a survey. IEEE *Transactions on Systems, Man, and Cybernetics*, *Part C (Applications and Reviews)*, 41(6), 765-781.

Huang, D., Wang, Y., & Wang, Y. (2007). A robust method for near infrared face recognition based on extended local binary pattern. *Advances in Visual Computing*, 437-446.

Iakovidis, D., Keramidas, E., & Maroulis, D. (2008). Fuzzy local binary patterns for ultrasound texture characterization. *Image analysis and recognition*, 750-759.

Jackson, W. (2015). Digital Image Compositing Fundamentals. Berkeley, CA: Apress.

Jayaraman, S., Esakkirajan, S., & Veerakumar, T. (2009). Digital image processing. New Delhi: Tata McGraw-Hill Education.

Jin, H., Liu, Q., Lu, H., & Tong, X. (2004). Face detection using improved LBP under Bayesian framework. In *Image and Graphics (ICIG'04), Third International Conference on* (pp. 306-309). IEEE.

Junding, S., Shisong, Z., & Xiaosheng, W. (2010). Image retrieval based on an improved CS-LBP descriptor. In *Information Management and Engineering (ICIME), 2010 The 2nd IEEE International Conference on* (pp. 115-117). IEEE.

Král, P., & Vrba, A. (2017). Enhanced Local Binary Patterns for Automatic Face Recognition. *arXiv preprint arXiv:1702.03349*.

Lahdenoja, O., Laiho, M., & Paasio, A. (2005). Reducing the feature vector length in local binary pattern-based face recognition. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (Vol. 2, pp. II-914). IEEE.

Larose, D. T. (2005). K- nearest neighbor algorithm. *Discovering Knowledge in Data: An Introduction to Data Mining*, 90-106.

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A maximum entropy framework for part-based texture and object recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International* Conference on (Vol. 1, pp. 832-838). IEEE.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on* (Vol. 2, pp. 2169-2178). IEEE.

Lee, C., Heidbreder, G., & Berggreen, A. (1973). Digital Image Enhancement Techniques Applied to Acoustic Images. 1973 Ultrasonic Symposium. doi:10.1109/ultsym.1973.196141.

Liao, S., & Chung, A. C. (2007). Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *Asian conference on computer vision* (pp. 672-679). Springer Berlin Heidelberg.

Liao, S., Zhu, X., Lei, Z., Zhang, L., & Li, S. (2007). Learning multi-scale block local binary patterns for face recognition. *Advances in biometrics*, 828-837.

Lindahl, T. (2007). Study of local binary patterns.

Lisin, D. A., Mattar, M. A., Blaschko, M. B., Learned-Miller, E. G., & Benfield, M. C. (2005). Combining local and global image features for object class recognition. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on* (pp. 47-47). IEEE.

Liu, J., & Shah, M. (2007). Scene modeling using co-clustering. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-7). IEEE.

Malathi, S., & Meena, C. (2010). An efficient method for partial fingerprint recognition based on Local Binary Pattern. In Communication Control and Computing Technologies (ICCCCT), 2010 IEEE International Conference on (pp. 569-572). IEEE.

Mayoraz, E., & Alpaydin, E. (1999). Support vector machines for multi-class classification. Engineering Applications of Bio-Inspired Artificial Neural Networks, 833-842.

Murphy, K., Torralba, A., Eaton, D., & Freeman, W. (2006). Object detection and localization using local and global features. In Toward Category-Level Object Recognition (pp. 382-400). Springer Berlin Heidelberg.

Murthy, S. K. (1998). Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, *2*(4), 345-389.

Nanni, L., Lumini, A., & Brahnam, S. (2010). Local binary patterns variants as texture descriptors for medical image analysis. *Artificial intelligence in medicine*, *49*(2), 117-125.

O'Connor, B., & Roy, K. (2013). Facial Recognition using Modified Local Binary Pattern and Random Forest. *International Journal of Artificial Intelligence & Applications*, *4*(6), 25.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, *29*(1), 51-59.

Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 24(7), 971-987.

Pietikäinen M. (2005) Image Analysis with Local Binary Patterns. In: Kalviainen H., Parkkinen J., Kaarna A. (eds) Image Analysis. SCIA 2005. Lecture Notes in Computer Science, vol 3540. Springer, Berlin, Heidelberg

Oliva, A., & Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, *42*(3), 145-175.

Oliver, A., Lladó, X., Freixenet, J., & Martí, J. (2007). False positive reduction in mammographic mass detection using local binary patterns. Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007, 286-293.

Park, S. B., Lee, J. W., & Kim, S. K. (2004). Content-based image classification using a neural network. *Pattern Recognition Letters*, *25*(3), 287-300.

Pietikäinen, M., Hadid, A., Zhao, G., & Ahonen, T. (2011). *Local binary patterns for still images* (pp. 13-47). Springer London.

Priyadarsini, V., Sivakumar, S., & Student, A. P. (2016). Segmentation Based on Clustering and Feature Extraction for Brain MRI. International Journal of Engineering Science, 5463.

Rassem, T. H., & Khoo, B. E. (2014). Completed local ternary pattern for rotation invariant texture classification. *The Scientific World Journal*, *2014*.

Said, A., & Pearlman, W. A. (1996). An image multiresolution representation for lossless and lossy compression. *IEEE Transactions on image processing*, *5*(9), 1303-1310.

Saleem, abdullah. S. (2014). Survey on Color Image Enhancement Techniques using Spatial Filtering. International Journal of Computer Applications, 0975.

Saxena, S., Sharma, S., & Sharma, N. (2016). Parallel Image Processing Techniques, Benefits and Limitations. *Research Journal of Applied Sciences, Engineering and Technology*, *12*(2), 223-238.

Schettini, R., & Corchs, S. (2010). Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, *2010*(1), 746052.

Scheuermann, B., & Rosenhahn, B. (2009). Analysis of numerical methods for level set based image segmentation. *Advances in Visual Computing*, 196-207.

Sheha, M. A., Mabrouk, M. S., & Sharawy, A. (2012). Automatic detection of melanoma skin cancer using texture analysis. *International Journal of Computer Applications*, *42*(20), 22-26.

Tan, X., & Triggs, B. (2007). Enhanced local texture feature sets for face recognition under difficult lighting conditions. In *International Workshop on Analysis and Modeling of Faces and Gestures* (pp. 168-182). Springer Berlin Heidelberg.

Vadivambal, R., & Jayas, D. S. (2015). *Bio-imaging: principles, techniques, and applications*. CRC Press.

Varma, M., & Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, *62*(1), 61-81.

Varma, M., & Zisserman, A. (2009). A statistical approach to material classification using image patch exemplars. *IEEE transactions on pattern analysis and machine intelligence*, *31*(11), 2032-2047.

Wong, W., Abu-Shareha, A. A., Pasha, M. F., & Mandava, R. (2013). Enhanced local binary pattern for chest X-ray classification. In *Image Information Processing (ICIIP), 2013 IEEE Second International Conference on* (pp. 271-275). IEEE.

Yang, H., & Wang, Y. (2007). A LBP-based face recognition method with Hamming distance constraint. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on* (pp. 645-649). IEEE.

Young, I. T., Gerbrands, J. J., & Van Vliet, L. J. (1998). *Fundamentals of image processing*. Delft: Delft University of Technology.

Zhang, L., Chu, R., Xiang, S., Liao, S., & Li, S. (2007). Face detection based on multi-block lbp representation. *Advances in biometrics*, 11-18.

Zhang, W., Shan, S., Chen, X., & Gao, W. (2007). Local Gabor binary patterns based on Kullback–Leibler divergence for partially occluded face recognition. IEEE Signal Processing Letters, 14(11), 875-878.

Zhao, Y., Jia, W., Hu, R. X., & Min, H. (2013). Completed robust local binary pattern for texture classification. *Neurocomputing*, *106*, 68-76.