

# Improvements of Local Directional Pattern for Texture Classification



**Abuobayda Mohammed Mosa Shabat**

211538106

A thesis submitted in fulfilment of the  
academic requirements for the degree of  
*Doctor of Philosophy in Computer Engineering*  
*in the*  
*school of Engineering*  
*University of KwaZulu-Natal*  
*Durban, South Africa*

Examiner's copy

September 2017

**UNIVERSITY OF KWAZULU-NATAL,  
COLLEGE OF AGRICULTURE,  
ENGINEERING AND SCIENCE  
DECLARATION**

The research described in this thesis was performed at the University of KwaZulu-Natal under the supervision of Professor Jules-Raymond Tapamo. I hereby declare that all materials incorporated in this thesis are my own original work except where acknowledgement is made by name or in the form of reference. The work contained herein has not been submitted in part or whole for a degree at any other university.

Signed \_\_\_\_\_

Abuobayda Shabat

Date:

As the candidate's supervisor, I have approved/disapproved this dissertation for submission.

Signed \_\_\_\_\_

Prof. Jules-Raymond Tapamo

Date:

**UNIVERSITY OF KWAZULU-NATAL,  
COLLEGE OF AGRICULTURE,  
ENGINEERING AND SCIENCE  
DECLARATION 1 - PLAGIARISM**

I, Abuobayda Mohammed Mosa Shabat, declare that:

1. The research reported in this thesis, except where otherwise indicated, is my original work.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other person's data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This dissertation/thesis does not contain other person's writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
  - their words have been re-written, but the general information attributed to them has been referenced;
  - where their exact words have been used, their writing has been placed inside quotation marks, and referenced.
5. Where I have reproduced a publication of which I am an author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.
6. This dissertation/thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the dissertation/thesis

Signed \_\_\_\_\_

**UNIVERSITY OF KWAZULU-NATAL,  
COLLEGE OF AGRICULTURE,  
ENGINEERING AND SCIENCE  
DECLARATION 2 - PUBLICATIONS**

I, Abuobayda Mohammed Mosa Shabat, declare that the following publications resulted from this dissertation.

1. A.M. Shabat, J. R. Tapamo, "A comparative study of the use of local directional pattern for texture-based informal settlement classification", Journal of Applied Research and Technology, vol 15, pp. 250-258, 2017
2. A. M. Shabat and J.R. Tapamo, "Directional Local Binary Pattern for Texture Analysis", Lecture Notes in Computer Science, vol. 9730, pp. 226-233, June 2016.
3. A. M. Shabat and J.R. Tapamo, "An Improved scheme of local Directional Pattern for texture analysis", Lecture Notes in Computer Science, vol. 10425, pp. 1–13, 2017.
4. A. M. Shabat and J.R. Tapamo. "Angled Local Directional Pattern for Texture Analysis with an Application to Facial Expressions", IET Computer Vision. (under revision)
5. A. M. Shabat and J.R. Tapamo. "Circular Local Directional Pattern for Texture Analysis", National Academy Science Letters. (under review)

Signed \_\_\_\_\_

Say, "Indeed, my prayer, my rites of sacrifice, my living and my dying are for Allah , Lord of the worlds".



## **Acknowledgements**

I would like to show my gratefulness to my supervisor, Professor Jules-Raymond Tapamo for his guidance, encouragement and support during the course of my PhD programme. You have been a wonderful supervisor. Special thanks also goes to my father (Abuobayda Mohammed Shabat), my Mother (Hanan Shabat) and siblings; they offered prayers constantly towards the successful completion of this PhD programme. The support of my friends at the University of KwaZulu-Natal is also appreciated. I wish to say a very special thank you to my loving and caring wife, Bayan Shabat, who stood by me through all the difficult times of my programme. Lastly my son, Mohammed Shabat, is deeply appreciated.



## Abstract

The Local Directional Pattern (LDP) method has established its effectiveness and performance compared to the popular Local Binary Pattern (LBP) method in different applications. In this thesis, several extensions and modification of LDP are proposed with an objective to increase its robustness and discriminative power. Local Directional Pattern (LDP) is dependent on the empirical choice of three for the number of significant bits used to code the responses of the Kirsch Mask operation.

In a first study, we applied LDP on informal settlements using various values for the number of significant bits  $k$ . It was observed that the change of the value of the number of significant bits led to a change in the performance, depending on the application.

Local Directional Pattern (LDP) is based on the computation Kirsch Mask application response values in eight directions. But this method ignores the gray value of the center pixel, which may lead to loss of significant information. Centered Local Directional Pattern (CLDP) is introduced to solve this issue, using the value of the center pixel based on its relations with neighboring pixels. Local Directional Pattern (LDP) also generates a code based on the absolute value of the edge response value; however, the sign of the original value indicates two different trends (positive or negative) of the gradient. To capture the gradient trend, Signed Local Directional Pattern (SLDP) and Centered-SLDP (C-SLDP) are proposed, which compute the eight edge responses based on the two different directions (positive or negative) of the gradients.

The Directional Local Binary pattern (DLBP) is introduced, which adopts directional information to represent texture images. This method is more stable than both LDP and LBP because it utilizes the center pixel as a threshold for the edge response of a pixel in eight directions, instead of employing the center pixel as the threshold for pixel intensity of the neighbors, as in the LBP method. Angled Local directional pattern (ALDP) is also presented, with an objective to resolve two problems in the LDP method. These are the value of the number of significant bits  $k$ , and to taking into account the center pixel value. It computes the angle values for the edge response of a pixel in eight directions for each angle ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ). Each angle vector contains three values. The central value in each vector is chosen as a threshold for the other two neighboring pixels. Circular Local Directional Pattern (CILDP) is

also presented, with an objective of a better analysis, especially with textures with a different scale. The method is built around the circular shape to compute the directional edge vector using different radiuses.

The performances of LDP, LBP, CLDP, SLDP, C-SLDP, DLBP, ALDP and CILDP are evaluated using five classifiers (K-nearest neighbour algorithm (k-NN), Support Vector Machine (SVM), Perceptron, Naive-Bayes (NB), and Decision Tree (DT)) applied to two different texture datasets: Kylberg dataset and KTH-TIPS2-b dataset. The experimental results demonstrated that the proposed methods outperform both LDP and LBP.

# Table of contents

<b>List of figures</b>	<b>xiii</b>
<b>List of tables</b>	<b>xvii</b>
<b>List of Abbreviations</b>	<b>xix</b>
<b>1 General Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	3
1.2 The contribution of the thesis . . . . .	5
1.3 The outline of the thesis . . . . .	6
<b>2 Texture Analysis</b>	<b>9</b>
2.1 Introduction . . . . .	9
2.2 Methods related to LDP . . . . .	12
2.2.1 The Gray-Level Co-occurrence Matrix (GLCM) . . . . .	12
2.2.2 Local Binary Pattern . . . . .	14
2.2.3 Local Directional Pattern (LDP) . . . . .	15
2.3 Classification . . . . .	16
<b>3 A Comparative Study of the Use of Local Directional Pattern for Texture Based Informal Settlements Classification</b>	<b>19</b>
<b>4 Directional Local Binary Pattern for Texture Analysis</b>	<b>29</b>
<b>5 An Improved Scheme of Local Directional Pattern for Texture Analysis</b>	<b>39</b>
<b>6 Angled Local Directional Pattern for Texture Analysis with an Application to Facial Expressions</b>	<b>53</b>
<b>7 Circular Local Directional Pattern for Texture Analysis</b>	<b>69</b>

<b>8 Discussion</b>	<b>77</b>
8.1 Classification evaluation . . . . .	77
8.2 Applying the enhancement methods to KTH-TIPS2b dataset . . . . .	79
8.2.1 Dataset . . . . .	79
8.2.2 Experimental Results . . . . .	79
8.3 Facial expression application . . . . .	85
8.3.1 Dataset . . . . .	85
8.3.2 Experimental Results . . . . .	86
<b>9 Conclusions</b>	<b>93</b>
<b>References</b>	<b>95</b>

# List of figures

1.1	Natural Texture . . . . .	1
1.2	Man-made texture . . . . .	2
1.3	This example has four specific textures: leopard texture, jungle texture, snow texture and grass texture. . . . .	3
1.4	The results of vegetation extraction (Chen et al., 2014) . . . . .	4
1.5	Road extraction (Li and Zhang, 2015) . . . . .	4
1.6	Examples of periodic patterns . . . . .	5
2.1	Examples of detection of a desired element (Leung and Malik, 1996) . . . . .	10
2.2	Example of detection of deformed lattice wallpaper image (Park et al., 2009). . . . .	11
2.3	"Example results for the PSU dataset. Each column shows the results for a particular image. The first row shows the original images. The second row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The third row shows the scale estimates"(Yalniz and Aksoy, 2010) . . . . .	11
2.4	Calculation of the GLCM matrix in different directions . . . . .	12
2.5	GLCLL structure . . . . .	13
2.6	GLCHS structure . . . . .	13
2.7	LBP Example: binary code is read clockwise starting from the top left neighbor . . . . .	14
2.8	Circular neighbor sets for different number of pixel (P) and radius (R) . . . . .	15
2.9	CSLBP structure . . . . .	15
2.10	Applying LDP operator on a $3 \times 3$ block . . . . .	16
2.11	"3D local directional kernels. Three unique cases are presented, showing the kernel direction and the corresponding weight values. The other 23 kernels can be obtained by applying rotational transforms to them."(Sivapalan et al., 2013) . . . . .	17

8.1	"Blue and gray circles indicate cases known to be positive (TP + FN) and negative (FP + TN), respectively, and blue and gray backgrounds/squares depict cases predicted as positive (TP + FP) and negative (FN + TN), respectively. Equations for calculating each metric are encoded graphically in terms of the quantities in the confusion matrix. FDR, false discovery rate."Lever et al. (2016) . . . . .	78
8.2	KTH-TIPS2b samples . . . . .	79
8.3	The histogram applied on LDP image . . . . .	80
8.4	The histogram applied on LDP image . . . . .	80
8.5	The Learning Curve of LDP . . . . .	80
8.6	The Learning Curve of ALDP . . . . .	80
8.7	The Learning Curve of DLBP . . . . .	81
8.8	The Learning Curve of CLDP (UP) . . . . .	81
8.9	The Learning Curve of CLDP(DOWN) . . . . .	81
8.10	The Learning Curve of LDP(Positive) . . . . .	81
8.11	The Learning Curve of LDP(Negative) . . . . .	81
8.12	The Learning Curve of CILDP(2) . . . . .	81
8.13	The Learning Curve of CILDP(3) . . . . .	82
8.14	The Learning Curve of CILDP(4) . . . . .	82
8.15	The Learning Curve of CILDP(5) . . . . .	82
8.16	The Classification Report of LDP . . . . .	83
8.17	The Classification Report of ALDP . . . . .	83
8.18	The Classification Report of DLBP . . . . .	83
8.19	The Classification Report of CLDP (UP) . . . . .	83
8.20	The Classification Report of CLDP(DOWN) . . . . .	83
8.21	The Classification Report of LDP(Positive) . . . . .	83
8.22	The Classification Report of LDP(Negative) . . . . .	84
8.23	The Classification Report of CILDP(2) . . . . .	84
8.24	The Classification Report of CILDP(3) . . . . .	84
8.25	The Classification Report of CILDP(4) . . . . .	84
8.26	The Classification Report of CILDP(5) . . . . .	84
8.27	Samples of the CK+ dataset . . . . .	85
8.28	The Learning Curve of LDP . . . . .	86
8.29	The Learning Curve of ALDP . . . . .	86
8.30	The Learning Curve of DLBP . . . . .	87
8.31	The Learning Curve of CLDP (UP) . . . . .	87

---

8.32	The Learning Curve of CLDP(DOWN)	87
8.33	The Learning Curve of LDP(Positive)	87
8.34	The Learning Curve of LDP(Negative)	87
8.35	The Learning Curve of CILDP(2)	87
8.36	The Learning Curve of CILDP(3)	88
8.37	The Learning Curve of CILDP(4)	88
8.38	The Learning Curve of CILDP(5)	88
8.39	The Classification Report of LDP	89
8.40	The Classification Report of ALDP	89
8.41	The Classification Report of DLBP	89
8.42	The Classification Report of CLDP (UP)	89
8.43	The Classification Report of CLDP(DOWN)	90
8.44	The Classification Report of LDP(Positive)	90
8.45	The Classification Report of LDP(Negative)	90
8.46	The Classification Report of CILDP(2)	90
8.47	The classification Report of CILDP(3)	90
8.48	The classification Report of CILDP(4)	90
8.49	The Classification report of CILDP(5)	91



# List of tables

8.1	Interpretation of Kappa( $\kappa$ ) . . . . .	79
8.2	Average kappa scores for all the methods applied on the KTH-TIPS2b dataset	85
8.3	The average kappa scores for all the methods applied on the Cohn-Kanade Dataset . . . . .	91



# List of Abbreviations

## Acronyms / Abbreviations

ALDP Angled Local Directional Pattern

CILDLP Circular Local Directional Pattern

CLBP Completed Local Binary Pattern

CLDP Centered Local Directional Pattern

CSLBP Center-Symmetric Local Binary Pattern

CT Computed Tomography

DLBP Directional Local Binary Pattern

DLBP Dominant Local Binary Pattern

DOG Difference of Gaussian

FN False Negative

FP False Positive

GLCHS Gray Level Co-occurrence Hybrid Structure

GLCLL Gray Level Co-occurrence Linked List

GLCM Gray-Level Co-occurrence Matrix

GMRF Gaussian Markov Random Fields

HOG Histogram of Oriented Gradients

HWLD Histogram of Weight Local Directional

- k-NN k-Nearest Neighbor Algorithm
- LBP Local Binary Pattern
- LDP Local Directional Pattern
- LDPv Local Directional Pattern Variance
- LDT Local Directional Ternary
- LEBC Locally Enhanced Binary Coding
- LLDP Local Line Directional Pattern
- LOG Laplacian of Gaussian
- MCMCM Modified Color Motif Co-occurrence Matrix
- MLP Multilayer Perceptron
- MRF Markov Random Field
- NB Naive-bayes
- PCA Principal Component Analysis
- SLDP Signed Local Directional Pattern
- SVM Support Vector Machine
- TN True Negative
- TP True Positive

# Chapter 1

## General Introduction

Texture is the visual aspect of the surface of a material, which originates from the three-dimensional structure of a physical object. It indicates the surface properties of materials and these characteristics are recognized by humans at first by visual acuity and is then verified by touch. In a general sense, the surface texture appears as a result of the interaction between light and clauses of the surface (roughness, softness, the degree of refinement, the reflected light from material surfaces and shape). A texture is usually described as rough or smooth, hard or soft, glossy or matt, etc. (Flexner, 1987). Texture can be categorized in two ways: in terms of the degree of smoothness, roughness, regular and irregular, or as natural or man-made. Figure 1.1 presents a few natural textures and Figure 1.2 shows a few man-made textures.



Fig. 1.1 Natural Texture

In computer vision, there is no exact definition of the notion of texture. The main reason is that the characterizing texture in digital image can be sensed via variations of color or the intensity of capturing. Many researchers have tried to define texture:



Fig. 1.2 Man-made texture

We may regard texture as what constitutes a macroscopic region. Its structure is simply attributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule. (Tamura et al., 1978)

A region in an image has a constant texture if a set of local statistics or other local properties of the picture function are constant, slowly varying, or approximately periodic. (Sklansky, 1978)

Texture is defined for our purposes as an attribute of a field having no components that appear enumerable. The phase relations between the components are thus not apparent. Nor should the field contain an obvious gradient. The intent of this definition is to direct attention of the observer to the global properties of the display — i.e., its overall “coarseness”, “bumpiness”, or “finess”. Physically, non enumerable (aperiodic) patterns are generated by stochastic as opposed to deterministic processes. Perceptually, however, the set of all patterns without obvious enumerable components will include many deterministic (and even periodic) textures. (Richards and Polit, 1974)

Image texture we consider is non figurative and cellular... An image texture is described by the number and types of its (tonal) primitives and the spatial organization or layout of its (tonal) primitives... A fundamental characteristic of texture: it cannot be analyzed without a frame of reference of tonal primitive being stated or implied. For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture. Then as resolution increases, it takes on a fine texture and then a coarse texture. (Haralick, 1979)

Texture is a feature that divides images into a different areas. Each area holds a common characteristic. The picture in Figure1.3 has four specific textures: leopard texture, jungle texture, snow texture and grass texture. Theses features can be employed to recognize different object classes.



Fig. 1.3 This example has four specific textures: leopard texture, jungle texture, snow texture and grass texture.

## 1.1 Motivation

Texture analysis is used to segment and recognize images based on the spatial shape of edges or color. It demonstrates its strength, especially in varied illumination conditions, for example, in the outdoor conditions. Texture analysis is an important aspect in computer vision and it has been used in many applications. For instance, it has been used in the medical field for early detection of breast cancer for women through the recognition of microcalcifications in the breast (Megalooikonomou et al., 2007); distinguishing the pain state in the patient through a facial image, characterization of pap cancer by detecting the abnormal smear cells Nanni et al. (2010), and early predicting of hepatitis C disease from liver computed tomography (CT) images for survivors of the colorectal cancer (Miles et al., 2009).

In other fields, remote sensing has been used to extract areas of vegetation using satellite images that contain natural forests and plantations and to identify hydro-thermal uranium ore areas (Pan et al., 2013), for instance, in Heyuan city- Guangdong province in south China (Figure 1.4) (Chen et al., 2014). Remote sensing is also used to extract residential areas, which can be useful in disasters or in urban extensions (Zhang et al., 2016) or for road networks where roads are extracted through very high-resolution satellite images, as shown in Figure 1.5 (Li and Zhang, 2015).

In agriculture, identification of plant diseases can be carried out by extracting the infected area of the plant, captured using a digital camera (Phadikar and Goswami, 2016). In the forensic field texture analysis is applied to identify victims of massive disasters through dental images (Joseph and Santhi, 2016). Textures have a high contrast in visual appearance, as they may be oriented or captured in various sizes and illumination conditions. Most of the features

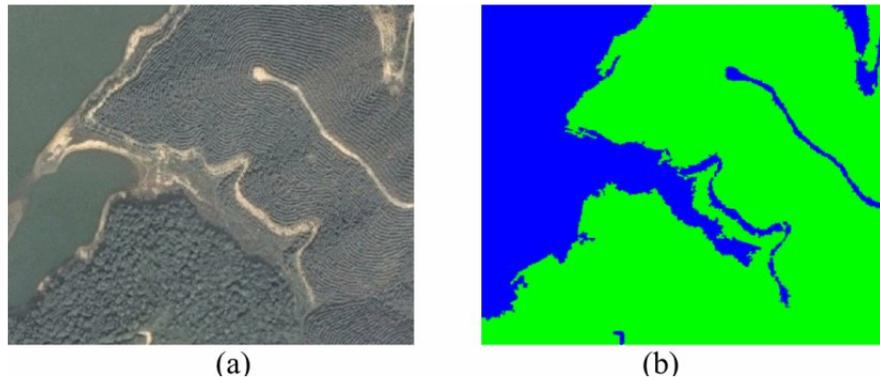


Fig. 1.4 The results of vegetation extraction (Chen et al., 2014)

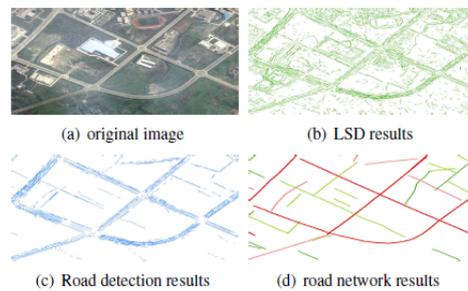


Fig. 1.5 Road extraction (Li and Zhang, 2015)

are computationally too complex to meet the real-time requirements of many computer vision applications. For this reason, only few texture feature methods are capable of performing well enough for real-world applications, despite the large amount of texture classification research done since the 1960s. In order to employ a feature in a real world application, it must be robust against rotations, scaling, illumination changes and be computationally inexpensive.

Grounded on the mathematical theory of Fedorov groups Liu et al. (2004) present a computational model for periodic pattern perception (wallpaper group and frieze group, as shown in Figure 1.6).

In the mid 90s (Ojala et al., 1996) presented an efficient method for texture analysis, which is a Local binary pattern (LBP). It shows a high demonstrated superiority in many comparative studies for speed and performance and has been employed successfully in several real world computer vision applications. The success of LBP has encouraged many scholars to further research, by presenting improvements that modernize and build powerful extensions to LBP such as dominant local binary pattern (DLBP) (Liao et al., 2009), completed local binary pattern (CLBP) (Guo et al., 2010) and center-symmetric local binary pattern (CSLBP) (Heikkilä et al., 2006), among others.

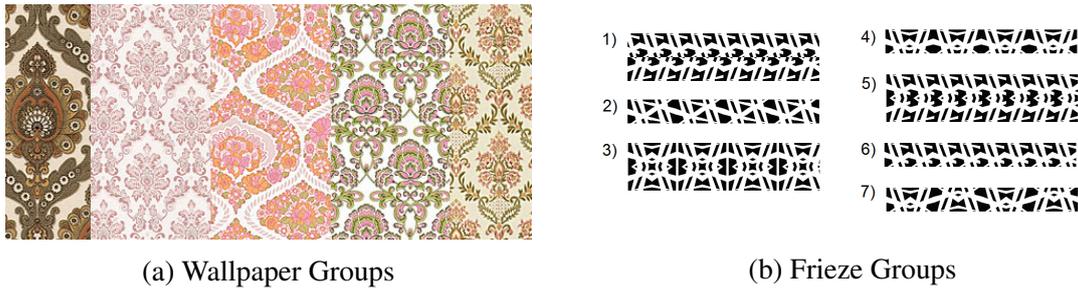


Fig. 1.6 Examples of periodic patterns

Research in recent years has started to focus on directional information instead of intensity information, the reason for this being that the directional information encoded is more stable than the pixel intensity (Luo et al., 2016). Some examples are Jabid et al. (2010), who presented a low-level feature, Local Directional Pattern (LDP), which uses the edge responses of eight different directions for each pixel; Luo et al. (2016), who presented the local line directional pattern (LLDP), using the line direction response instead of the gradient response and Shabat and Tapamo (2016), who presented the directional local binary pattern (DLBP) using the central pixel as a threshold for the eight directional response value of the neighborhood. In this thesis we present several improvement of LDP.

## 1.2 The contribution of the thesis

The Local Directional Pattern (LDP) method has established its effectiveness and performance compared to the popular Local Binary Pattern (LBP), in different applications. However, in its native form, the LDP suffers from various shortcomings.

Three of these are, overlooking the central value in the local neighborhood, randomly selecting the number of the most significant edges, which vary depending on the applications and LDP operator cannot properly detect large-scale texture structures. This is because of LDP's structure that targets a small local neighborhood. All these reasons tend to decrease the performance of LDP.

In this thesis, several extensions and modifications of LDP are proposed, with the objective of increasing its robustness and discriminative power. Its use of the value of the central pixel, based on its relations with the local neighboring pixels or as a threshold for the edge response of a pixel in eight directions, aims for a better analysis. Four orientations ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ) are employed to compute the most directional edges in eight directions, rather than the manual way, to find the appropriate number of edges, which vary based on the application. To be able to capture the gradient trend, two directions (positive and negative) of the gradient

are employed to calculate the eight edge responses. To be able to measure the texture on multiple scale through the use of the circular shape to compute the directional edge vector, different radiuses are used.

The performance of LDP and the proposed extensions are shown in an extensive comparative study.

### 1.3 The outline of the thesis

The remaining part of thesis is organized into seven chapters and the description of each is reported below.

In Chapter 2, we discuss the analysis of the texture and the most popular and well-known texture methods, such as GLCM, GLCLL, GLCHS, LBP, CLBP, CS-LBP, LDP and LDPv. Literary revisions are also made to some of the issues concerning the knowledge of computer vision and texture analysis. The grounds for the use of texture feature to analyze the texture and recognize facial expressions are also given in this chapter.

In Chapter 3, paper I entitled, “A Comparative Study of the use of Local Directional Pattern for Texture Based Informal Settlements Classification” is presented. Local Directional Pattern (LDP) is applied on the informal settlements using various values for the number of significant bits  $k$ . It is observed that the change of the value of the number of significant bits leads to a change in the performance, depending on the application.

In Chapter 4, paper II entitled, “Directional Local Binary Pattern for Texture Analysis” introduces the new features method, Directional Local Binary Pattern (DLBP), that builds on the best attributes of both the LBP and LDP.

In Chapter 5, paper III presents three extensions that improve LDP. The first improvements involves the value of the central pixel in the calculation of the LDP code to obtain the new method of Centered Local Directional Pattern (CLDP). The second improvement is made by using the gradient sign (positive and negative). The third improvement integrates both the value of the central pixel and both gradient signs.

In Chapter 6, paper IV entitled “Angled Local Directional Pattern for Texture Analysis with an Application to Facial Expressions” is presented. In this paper we solve the problem of the search for the optimal value for the most significant edges, which change from one application to another. The angles values for the edge response of a pixel in eight directions for each angle are computed ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). Each angle vector contains three values. The central value in each vector is chosen as a threshold for the other two neighboring pixels.

In Chapter 7, paper V entitled “Circular Local Directional Pattern for Texture Analysis” is presented. Circular Local Directional Pattern (CLDP) is built around the circular shape to

compute the directional edge vector using different radiuses with the objective of identifying textures with a large scale.

In Chapter 8, an extensive comparative study between all the presented methods and LDP were discussed. In this chapter, two different datasets are used to evaluate the performance of each method.

In Chapter 9, an overview of the proposed methods is given, and the improvements achieved, the limitations and future work are discussed.



# Chapter 2

## Texture Analysis

### 2.1 Introduction

Texture can be identified as an aspect that presents the spatial distribution of the gray levels of the pixels in a region of a digital image. Texture analysis is the process of extracting useful information from the surfaces of entities that appear in an image, where each entity may appear in a small area of the image or the entire image. Many of the texture analysis algorithms involve extracting texture features of each entity. These algorithms may vary in the manner in which texture features are extracted and how they are delivered and identified.

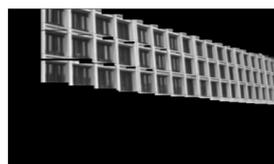
Traditionally, there are two types of texture analysis: the statistical or stochastic approach and the structural approach. Tuceryan and Jain (1990) had a wider perception of the texture analysis problems and they presented four categories of texture analysis instead of two: statistical, geometrical, model-based and signal processing. In the statistical approach, features are derived by considering the image signals, which reflect an inevitable feature of the spatial distribution of the signals into images. This approach can characterize some properties such as smoothness and roughness. Using a statistical approach, texture is measured based on the number of pixels which can be further classified into first order, second order and a higher order statistics (Ojala and Pietikäinen, 2004). For statistical properties of texture it is easiest to compute the first order like computing the variance or the mean of the of individual pixel values. For the second order, it computes the number of occurrences of two pixels separated by distance and direction. Entropy, energy and many more texture features are derived from calculating the second order.

The most popular statistical texture measures are co-occurrence matrix (Gotlieb and Kreyszig, 1990; Haralick, 1979), grey level run length (Galloway, 1975), gray level difference (Weszka et al., 1976), auto-correlation (Kaizer, 1995), Markov random field (MRF) (Besag, 1986), among others. The quantity of information arising from the statistical type of analysis

is usually not significant and it is easy to argue the utility of it in real-world applications or its validity in solving real problems.

The structural approach sees the texture image as a group of primitive (tonal), which are placed in the texture in a different pattern, as regular or frequent. Primitives are the key components for recognizing the external structure of the complex texture. Edge detection has been used to extract texture primitives by using different filters, for example, Difference of Gaussian (DOG) (Voorhees and Poggio, 1988) or Laplacian of Gaussian (LOG) (Marr, 1982) filters. Structural analysis is usually considered more stable in terms of the change in the illumination and sensitivity to noise than the statistical.

Leung and Malik (1996) present a structural texture algorithm that involves finding a desired element in the image and corresponding elements with their neighbors, and grouping the elements, as shown in Figure 2.1.



(a) architectural image



(b) A textile image

Fig. 2.1 Examples of detection of a desired element (Leung and Malik, 1996)

The mean-shift belief propagation method is applied to detect deformed lattice wallpaper in real-world images, which involves detecting points of interest, grouping these points (clustering) and iteration in finding the corresponding lattice structure, as shown in Figure 2.2 (Park et al., 2009).

Yalniz and Aksoy (2010) present a method to detect the natural structure of the texture, using multiple scale and multiple direction estimation in order to analyze their regularity through the texture primitive detected, using the LOG filter. Figure 2.3 shows an example of detection of texture regularity.

Model-based methods are based on the usage of a set of parameters generated by the variation in the pixel element in the texture to determine the image model. This can be employed to describe the texture. The random field model (Cross and Jain, 1983) is the most popular model-based method. For instance, Gaussian Markov random fields (GMRF) (Chellappa et al., 1993) are applied to capture the textural intensity. Another model-based

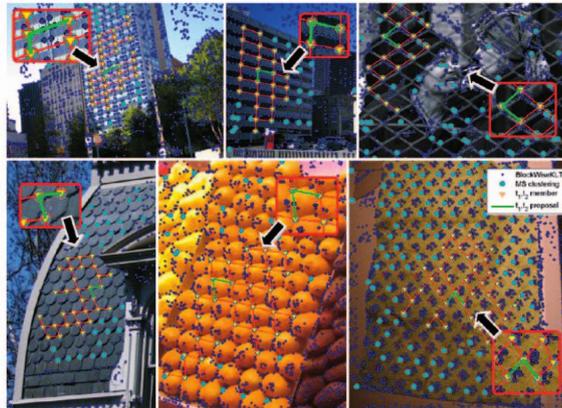


Fig. 2.2 Example of detection of deformed lattice wallpaper image (Park et al., 2009).

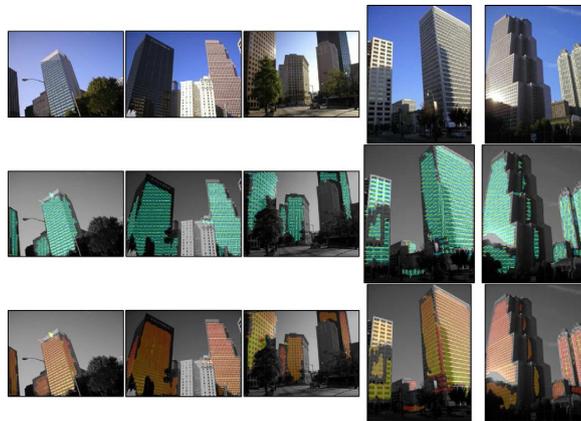


Fig. 2.3 "Example results for the PSU dataset. Each column shows the results for a particular image. The first row shows the original images. The second row shows the areas detected by thresholding the regularity index as green, and the associated orientation estimates as yellow line segments. The third row shows the scale estimates"(Yalniz and Aksoy, 2010)

method, named autoregressive has been employed to extract the texture features (Mao and Jain, 1992).

In texture analysis, signal processing methods are based on the idea of applying a particular filter on an image and analyzing the filter response frequency to create texture features.

Three domains are used to filter the image: spatial domain, frequency domain, and joint spatial/spatial-frequency. In the spatial domain, for instance, Sobel & Robert (Rosenfeld and Kak, 1976), Laplacian (Burt and Adelson, 1983) and Laws filters (Laws, 1980) have been regularly used to extract the edges and boundaries and to measure the edge density. In the frequency domain, Fourier transform is applied to an image, and the global frequency is extracted to create texture features (Campbell and Robson, 1968). In the joint

spatial/spatial-frequency domain, DOG (Voorhees and Poggio, 1988) and pseudo-Wigner distribution (Jacobson and Wechsler, 1982) uses a blank filter with a specific orientation and frequency to filter the image, and texture features are then extracted from the transformed image.

There are some methods which integrate both approaches, statistical and structural (Tuceryan and Jain, 1990), such as local binary pattern (LBP) (Ojala et al., 1996) and local directional pattern (LDP) (Jabid et al., 2010).

A combined statistical and structural approach for unsupervised texture classification is also found in literature (Umarani et al., 2008).

## 2.2 Methods related to LDP

### 2.2.1 The Gray-Level Co-occurrence Matrix (GLCM)

In the early seventies, (Haralick et al., 1973) presented a GLCM method to classify terrain in aerial photographs. This method is one of the most commonly used in statistical texture analysis. (Tomita and Tsuji, 1990) called GLCM a second order statistical methods, because it gathers information from a pair of pixels instead of a single pixel. The GLCM method is based on counting the number of occurrences of a pair of pixels with a certain value and specific spatial relationship in the image. The spatial relationship is the displacement between a particular pixel and its neighbors in any of the four directions ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ). All of these values for pixel pairs are grouped in a two-dimensional matrix of size  $(G \times G)$ , where  $G$  is the number of gray levels. Figure 2.4 shows an example of the computation of the GLCM matrix in  $0^\circ$  and  $45^\circ$  directions. A number of statistical features can be extracted from the

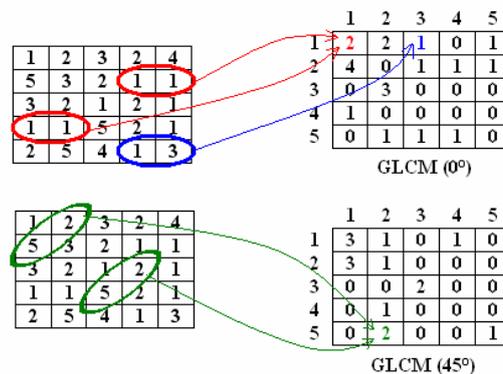


Fig. 2.4 Calculation of the GLCM matrix in different directions

GLCM matrix. From amongst these features, five features were selected as being the most

efficient (entropy, energy, homogeneity, correlation and inertia) (Connors and Harlow, 1980). Over the past years there have been numerous studies aiming to improve the performance and get more information from the GLCM. In one study by (Gelzinis et al., 2007), different values for the displacement factor between a pair of pixels are used. The result establishes that the alteration in the parameter value corresponds to change in performance, especially for a the texture of various sizes. Another version of GLCM that deals with color, Modified Color Motif Co-occurrence Matrix (MCMCM), is presented by (Subrahmanyam et al., 2013). It measures the inter-correlation of the three colors in the RGB (red, green, blue) color image. It also makes some improvements that reduce the amount of calculations and accelerate the GLCM computation. The gray level co-occurrence linked list (GLCLL) method was presented by (Clausi and Jernigan, 1998) to speed up the matrix calculation through the use of the linked list data structure and stores only the non zero probabilities, which led to a dramatic decrease in the number computations and memory storage. Figure 2.5 shows the GLCLL structure.

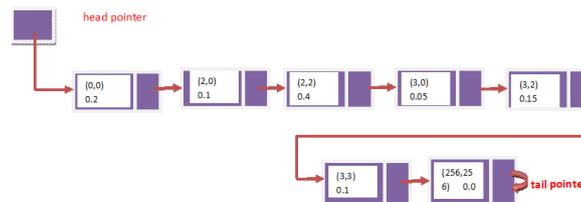


Fig. 2.5 GLCLL structure

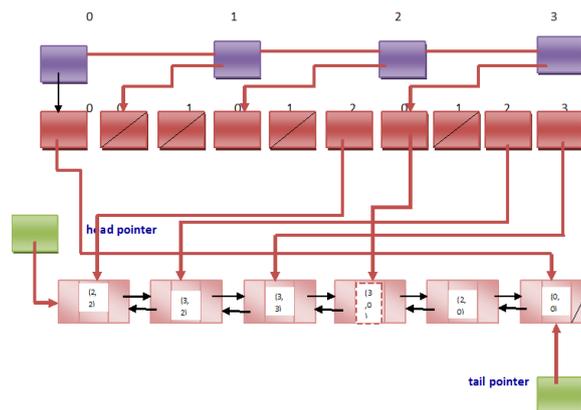


Fig. 2.6 GLCHS structure

The grey level co-occurrence hybrid structure (GLCHS) is another improvements that uses the hash table combined with the linked list data structure. It showed a high speed

performance compared to both the GLCM and GLCLL methods, because it does not require a sorted list (Clausi and Zhao, 2002). Figure 2.6 shows the GLCHS structure.

### 2.2.2 Local Binary Pattern

In recent years, LBP has gained popularity because of its simplicity and effectiveness in solving real-world problems. The feature method LBP was introduced by Ojala et al. (1996). It is based on the integration of texture analysis, statistical and structural approaches. Local Binary Pattern (LBP) relabels each pixel in an image based on a threshold of its surrounding pixels. The basic version of LBP works with a  $(3 \times 3)$  window. The pixel located at the central of the window acts as a threshold of its surrounding pixels. In fact, if the value of the adjacent pixel is greater than the value of the central pixel, it generates one bit value, otherwise it generates zero bit value. Numbers resulting from this process produce a binary string, which will replace the central pixel after converting it to a decimal number. Figure 2.7 shows an example of applying LBP operator on a  $3 \times 3$  window.

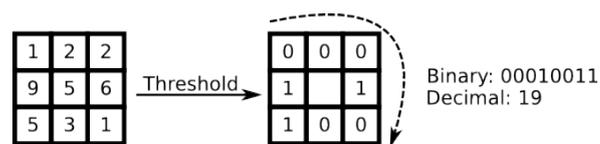


Fig. 2.7 LBP Example: binary code is read clockwise starting from the top left neighbor

Afterwards, the LBP operator will proceed to the next window until the image is completely transformed. After this stage, Histogram is extracted in order to gather the number of different binary frequency types from the transformed image, which represents the description of the texture. Every bin in the Histogram may stand for different types of edges and flat regions and spots. Because the original LBP is based on the thought of eight neighboring pixels, the number of the bin is equal to  $2^8 = 256$ . Since its origin, there have been many additions and improvements appended to the LBP with the aim of improving the performance. Ojala et al. (2002) introduced a novel feature which gives the LBP the ability to analyze a different scale and rotation invariant texture, applying the concept of circular shape in the selection the number of surrounding pixels and the size of the neighborhood by manipulating a diameter of this circle, as shown in Figure 2.8. One of the disadvantages of this method is that if the number of adjacent pixels is chosen to be 16, it leads to the increase of the size of the Histogram which becomes too large and impractical at  $2^{16} = 65536$  bins. To resolve this issue, Ojala et al. (2002) have proposed something called a "uniform pattern", which computes the upper limit value of the change in the bit status from 0 to 1 and vice versa. For instance, 0 transitions in both 11111111 and 00000000, while 00001100 and 01111000

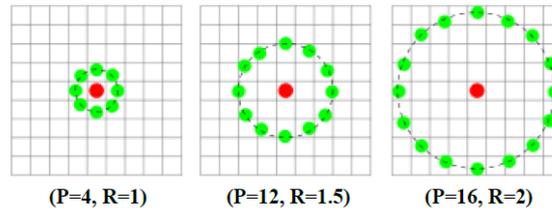


Fig. 2.8 Circular neighbor sets for different number of pixel (P) and radius (R)

contain 2 transitions and so on. It was also observed that there are nine patterns and their circular rotated version presents more than 90% of local patterns in the image. The center symmetric LBP (CS-LBP) is based on the idea of comparing each pixel with the parallels of the opposite pixel, with the objective of getting a shorter histogram (Heikkilä et al., 2009), as shown in Figure 2.9. In this method the number of bins is only  $2^4 = 16$  compared to the LBP, where the number of bins is  $2^8 = 256$ , which reduces the LBP labels that are used in region descriptors.

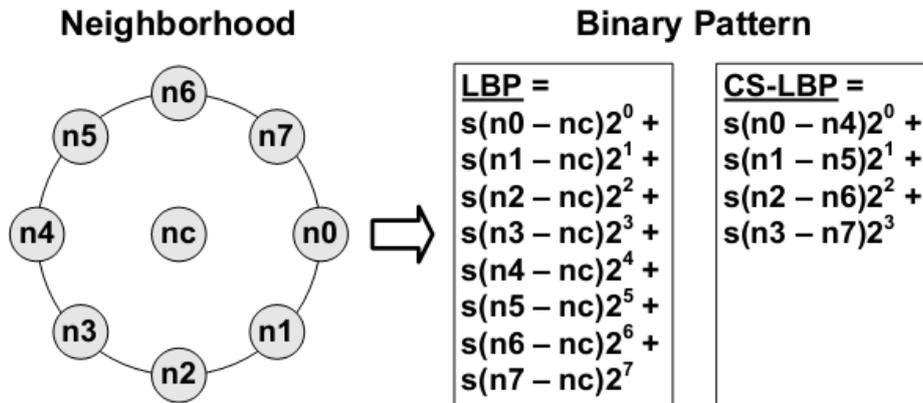


Fig. 2.9 CSLBP structure

### 2.2.3 Local Directional Pattern (LDP)

Despite the strength and spread of the LBP, it suffers from excessive noise sensitivity and is non-monotonic in illumination conditions due to its structure, which is based on computing the pixel intensity (Jabid et al., 2010; Zhou et al., 2008). For this reason, Jabid et al. (2010) present a more stable method which is less impressed by the random noise method called Local Directional Pattern (LDP). This method is based on computing the edge response values of the surrounding pixels in a  $3 \times 3$  window. Edge response values are computed by

applying the Kirsch Mask in eight directions at each pixel of the image. Figure 2.10 shows an example of applying LDP to the eight Kirsch Masks.

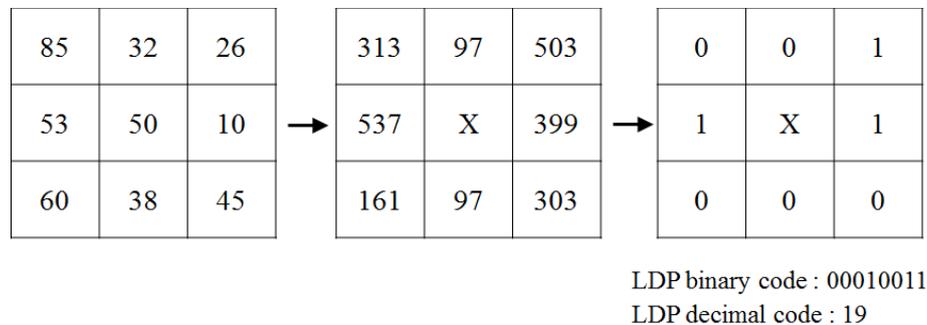


Fig. 2.10 Applying LDP operator on a  $3 \times 3$  block

The LDP variance (LDPv) method works on the classification of both the contrast and the spatial structure in order to bring more stability against varying illumination and more accuracy in facial identification (Kabir et al., 2010). In LDPv, each LDP code is integrated with the corresponding variance weight. Principal Component Analysis (PCA) is applied to reduce the feature dimension by selecting the most distinguishable features.

Most researchers in a local descriptor use histogram to characterize an object. In both LDP and LBP histogram is extracted as a feature vector. However Kim et al. (2013) use the LDP image as an input for PCA and Gabor-wavelet to improve the performance and reduce the affect of change in the light conditions.

Other improvements by Sivapalan et al. (2013) integrate the idea of both the LDP and the histogram of oriented gradients (HOG) method and in the Histogram of weight local directional (HWLD) method. The eight edge response values are computed as in the LDP concept, adding a specific weight based on the military posture of each direction in each pixel. Figure2.11 shows an example of extending HWLD to a 3-D application.

Different methods were proposed with an aim to improve the LDP such as Local Directional Ternary (LDT) (Ryu et al., 2017) and Improved Local Directional pattern for low resolution images (Garg and Kaur, 2016).

## 2.3 Classification

Humans usually commit many errors in analytical processing of data and are unable to visualize relationships between multi-dimension features. This makes it difficult to find solutions to many problems. Classification can be applied successfully to these problems to improve performance of systems. Classification is one of the most significant problems in

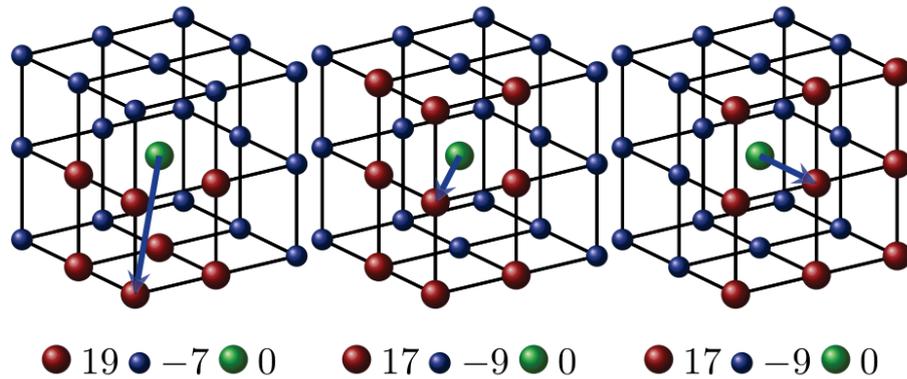


Fig. 2.11 "3D local directional kernels. Three unique cases are presented, showing the kernel direction and the corresponding weight values. The other 23 kernels can be obtained by applying rotational transforms to them."(Sivapalan et al., 2013)

machine learning and the machine vision. The objective of the classification is to recognize the unknown object and to select the most appropriate class from known classes or search for the probability of its occurrence in one of the known classes. In texture classification, texture is classified by identifying the unknown image to one of the known classes.

There are two fundamental types of image classification, the supervised and the unsupervised. In the supervised approach it is essential to know the features and the characteristics of each sample in the training sample corresponding to defined classes before determining the decision to identify the unknown sample (Duda et al., 1973). Hence the training sample needs to be merged with its own class label. In the unsupervised classification, the algorithm defines classes without any prior knowledge. It categorizes a group of samples into different classes based on their features and pattern. Texture classification has been used to evaluate many texture features methods.

Song et al. (2013) present a more robust features method called, Locally Enhanced Binary Coding (LEBC). In this setup, two classifiers VZ\_MR8 (Varma and Zisserman, 2005) and VZ\_Joint (Varma and Zisserman, 2009)) are used to compare the performance of LEBC, Local Ternary Pattern (LTP)(Tan and Triggs, 2010), CLBP(Guo et al., 2010) and the formal LBP. Four different classifiers (Naive-bayes(NB); Multilayer Perceptron(MLP); Support Vector Machine (SVM); and k-nearest Neighbor Algorithm(k-NN)) are used to evaluate the performance of both methods LDP and GLCM in four different orientations ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). Results show that LDP outperforms GLCM when applied to the classification of informal settlements (Shabat and Tapamo, 2014). Shabat and Tapamo (2016) make a comparison between directional local binary pattern (DLBP), the known LBP method and local directional pattern (LDP) using four classifiers Naive-bayes(NB), Multilayer

Perceptron(MLP), Support Vector Machine (SVM), k-nearest Neighbor Algorithm(k-NN).  
Results show that DLBP has a superior performance.

## **Chapter 3**

# **A Comparative Study of the Use of Local Directional Pattern for Texture Based Informal Settlements Classification**



Original

# A comparative study of the use of local directional pattern for texture-based informal settlement classification

Abuobayda M. Shabat, Jules-Raymond Tapamo\*

University of KwaZulu-Natal Durban, South Africa

Received 7 August 2015; accepted 9 December 2016

Available online 25 May 2017

## Abstract

In developing and emerging countries progression of informal settlements has been a fast growing phenomenon since the mid-1990s. Half of the world's population is housed in urban settlements. For instance, the growth of informal settlements in South Africa has amplified after the end of apartheid. In order to transform informal settlements to improve the living conditions in these areas, a lot of spatial information is required. There are many traditional methods used to collect these data, such as statistical analysis and fieldwork; but these methods are limited to capture urban processes, particularly informal settlements are very dynamic in nature with respect to time and space. Remote sensing has been proven to provide more efficient techniques to study and monitor spatial patterns of settlements structures with high spatial resolution. Recently, a new feature method, local directional pattern (LDP), based on kirsch masks, has been proposed and widely used in biometrics feature extraction. In this study, we investigate the use of LDP for the classification of informal settlements. Performance of LDP in characterizing informal settlements is then evaluated and compared to the popular gray level co-occurrence matrix (GLCM) using four classifiers (Naive-Bayes, Multilayer perceptron, Support Vector Machines, *k*-nearest Neighbor). The experimental results show that LDP outperforms GLCM in classifying informal settlements. © 2017 Universidad Nacional Autónoma de México, Centro de Ciencias Aplicadas y Desarrollo Tecnológico. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

**Keywords:** Informal settlements; Texture features; Gray level co-occurrence matrix; Local directional pattern; Classification

## 1. Introduction

In developing countries, informal settlements have become a phenomenon which grow very fast specially in the 21st Century. Half of the world's population is housed in urban settlements. The reason for this phenomenon is the immigration of people from the rural areas to the cities. Many previous studies aimed at extracting houses outline to quantify shape-based features of informal settlements. Object-based image analysis (OBIA) method estimates the size, spacing and shape of the houses by extracting the houses footprint (Blaschke & Lang, 2006). OBIA partitions remote sensing (RS) imagery into meaningful image-objects and assesses their characteristics through spatial, spectral and temporal scale (Hay & Castilla, 2008). Previous studies on geospatial methods have been used to estimate populations and

to distinguish the human settlements. For example, a study by Aminipouri, Sliuzas, and Kuffer (2009) estimates the population by creating an accurate inventory of buildings. The computer vision community is facing a very complex and challenging task extracting the spatial data from informal settlements. Constructions in informal settlements are built using various materials and are very close to each other and have no suitable organization. It makes the classification of informal settlements images an uphill task (McLaren, Coleman, & Mayunga, 2005). A number of researchers have tried to develop tools and techniques to characterize the informal settlements areas from remotely sensed data. Mayunga, Coleman, and Zhang (2007) present a new semi-automatic approach to extract buildings from informal settlements images obtained using Quick Bird. Snakes and radial casting algorithm were used to map the informal settlements images. The main limitation in this study is the difficulty of characterizing small houses. Khumalo, Tapamo, and Van Den Bergh (2011) applied two feature methods, Gabor filters and GLCM to distinguish different textural regions in Soweto area (Johannesburg, South Africa). They found Gabor filters more

\* Corresponding author.

E-mail address: [tapamoj@ukzn.ac.za](mailto:tapamoj@ukzn.ac.za) (J.-R. Tapamo).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

accurate than GLCM in classifying informal settlements. A study carried out by

Ella, Van Den Bergh, Van Wyk, and Van Wyk (2008) compared gray level co-occurrence (GLCM) and local binary pattern (LBP) in their ability to classify urban settlement. It is shown that both methods performed very well with a superior performance for LBP. Van Den Bergh (2011) investigates the powers of two features methods, GLCM and LBP, to classify Soweto (Johannesburg, South Africa) areas. It is established that the performance of the gray level co-occurrence matrix is superior to the local binary pattern on a combined spatial and temporal generalization problem, but the LBP features perform better on spatial-only generalization problems. In Owen and Wong (2013) an analysis is conducted on the shape, texture, terrain geomorphology and road networks to characterize the informal settlements and formal neighborhoods in Latin America. The results achieved were promising when finite data were used to recognize informal settlements. Asmat and Zamzami (2012) introduced an automated house detection technique to extract legal and illegal settlements in Pulau Gaya, Saba. The result shows that the edge to edge features can separate between houses that are less than 2 m away from each other. In Graesser et al. (2012), an investigation of nine statistics methods (GLCM Pan- Tex, Histogram of Oriented Gradients, Lacunarity, Line Support Regions, Linear Feature Distribution, Psuedo NDVI, Red-blue NDVI, Scale Invariant Feature Transform, and TEXTONS) is presented with different direction, structure size and shape and tested in four different cities. The GLCM PanTex, LSR, HoG and TEXTON features were found to be the best in characterizing the informal settlements and formal areas. A new feature method, local directional pattern (LDP), based on the known Kirsch kernels was recently proposed by Jabid, Kabir, and Chae (2010a). LDP has mainly been applied in biometrics: face recognition (Jabid, Kabir, & Chae, 2010b), signature verification (Ferrer, Vargas, Travieso, & Alonso, 2010) and facial expression recognition (Jabid, Kabir, & Chae, 2010c). In Shabat and Tapamo (2014) the powers of GLCM and LDP to characterize texture images are compared; the result shows that LDP outperforms GLCM. In this paper, GLCM and LDP are investigated using different numbers of significant bits; the final goal is to identify the most effective amongst them. The computation of the local directional pattern is based on the number of significant bits, and in this work four alternative values are considered: 2, 3, 4, 5 instead of 3 as in the classic LDP.

## 2. Materials and methods

In the following sections the different feature methods used the in the paper are presented

### 2.1. Gray level co-occurrence matrix (GLCM)

In the early 1970s Haralick, Shanmugam, and Dinstein (1973) proposed the extraction of fourteen features, from the GLCM of a gray level, to characterize the image texture. The computation of GLCM depends on two parameters: the orientation  $\theta$  formed by the line-segment connecting the two considered

pixels, and the distance ( $d$ )[number of pixels] between them. The direction  $\theta$  is usually quantized in 4 directions (horizontal –  $0^\circ$ , diagonal -  $45^\circ$ , vertical –  $90^\circ$ , anti-diagonal –  $135^\circ$ ).

To compute the gray-level co-occurrence matrix of a window in an image, the following parameters are considered:

- The window size,  $N_x \times N_y$ , where  $N_x$  is the number of rows and  $N_y$  the number of columns.
- Distance ( $d$ ) and directions  $\theta$ .
- And the range of gray values to consider in calculations  $0, \dots, G - 1$ .

We adopt the formulation used in Bastos, Liatsis, and Conci, 2008 and Eleyan and Demirel (2011) to present the calculation of GLCM. The GLCM is defined as the probability of occurrence of two gray levels at a given offset (with respect to given distance and orientation). Given the image  $I$ , of size  $N_x \times N_y$ , the value of the co-occurrence for the gray values  $i$  and  $j$ , at the distance ( $d$ ) and direction  $\theta$ ,  $P_{d,\theta}(i, j)$  can be defined as

$$P_{d,\theta}(i, j) = \sum_{x=0}^{N_x-1} \sum_{y=0}^{N_y-1} \delta_{d,\theta,i,j}(x, y) \tag{1}$$

Where

$$\delta_{d,\theta,i,j}(x, y) = \begin{cases} 1 & \text{if } I(x, y) = i \text{ and } I(x + \pi_x(d, \theta), y + \pi_y(d, \theta)) = j \\ 0 & \text{otherwise} \end{cases}$$

The offset  $(\pi_x(d, \theta), \pi_y(d, \theta))$  is used to compute the position of  $(x, y)$  with respect to its neighbor at the distance ( $d$ ) and direction  $\theta$ . For the 4 directions ( $0^\circ, 45^\circ, 90^\circ$ , and  $135^\circ$ ) and the offsets are given in Table 1.

#### 2.1.1. Haralick's features

Given an image  $I$  with  $G$  gray levels, an angle  $\theta$  and a distance ( $d$ ), after the gray level co-occurrence matrix,  $(P_{d,\theta}(i, j))_{0 \leq i, j \leq G-1}$ , number of features can be extracted, amongst which the most popular are the 14 Haralick features (energy or angular second moment (ENR), contrast (CON), correlation (COV), variance (VAR), inverse different moment (IDM), sum average (SAV), sum variance (SVA), sum entropy (SEN), entropy (ENT), difference variance (DIV), difference entropy (DEN), information measures of correlation (IMC1, IMC2), maximum correlation coefficient (MCC)). The computation of Haralick features is done using a normalized GLCM. The  $(i, j)$ th normalized entry,

Table 1  
Definition of different offsets.

$\theta$	$0^\circ$	$45^\circ$	$90^\circ$	$135^\circ$
$\pi_x(d, \theta)$	0	$-d$	$-d$	$-d$
$\pi_y(d, \theta)$	$d$	$d$	0	$-d$

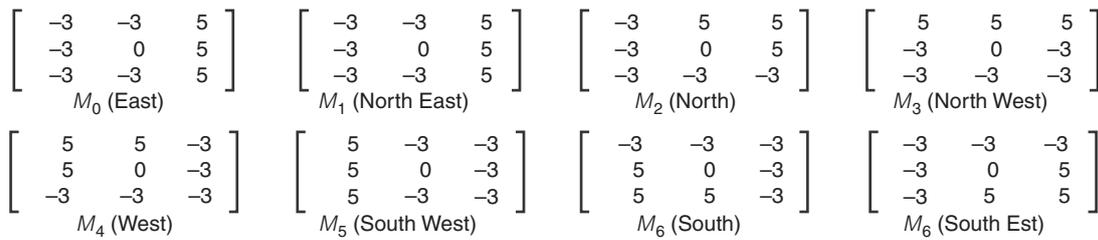


Fig. 1. The 8 Kirsch masks.

$P_{d,\theta}(i, j)$ , of  $P_{d,\theta}(i, j)$  is defined as

$$P_{d,\theta}(i, j) = \frac{P_{d,\theta}(i, j)}{\|P_{d,\theta}\|} \quad (2)$$

where  $\|P_{d,\theta}\| = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} P_{d,\theta}(i, j)$ . Details on the calculation of all these features can be found in Haralick et al. (1973). For each texture,  $T$ , a chosen distance ( $d$ ) and a direction  $\theta$ , 14 Haralick features can be extracted.

$$(glcm_{T,d,\theta}^i)_{i=1,\dots,14} \quad (3)$$

### 2.2. Local directional pattern (LDP)

The local binary pattern (LBP) operator depends on the change of the intensity around the pixel to encode the micro-level information of spot, edges and other local features in the image (Jabid et al., 2010a). The gradient is known to be more stable than the gray level; that is why some researches have replaced the intensity value at a pixel position with its gradient magnitude and calculated the LBP (Ferrer et al., 2010). The local directional pattern (LDP) was proposed by Jabid et al. (2010b) to resolve the problem with LBP, mentioned earlier. Since the LBP depends on the neighboring pixels' intensity which makes it unstable. Instead, LDP considers the edge response value in different direction. LDP features are based on eight bit binary

codes assigned to each pixel of an input image. It is composed of three steps (Kabir, Jabid, & Chae, 2010):

- **Calculation of eight directional responses** of particular pixels using the Kirsch compass edge detector in eight orientations ( $M_7, \dots, M_0$ ) centered on its own position as shown in Figure 1. Given a pixel  $(x, y)$  of an image,  $I$ , for each direction  $i$ , and using the corresponding mask  $M_i$  the  $i$ th directional response  $m_i$  can be computed as

$$m_i = \sum_{k=-1}^1 \sum_{l=-1}^1 M_i(k+1, l+1) \times I(x+k, y+l) \quad (4)$$

For the 8 directions a vector  $(m_7, \dots, m_0)$  is obtained. Figure 2 shows the Kirsch directional responses of a pixel  $(x, y)$ .

- **LDP code generation of the directional responses** obtained in the previous step. It is based on the selection of  $k$  most significant responses and set the corresponding bit to 1 leaving other  $(8 - k)$  bits to 0. Finally, the LDP code,  $LDP_{x,y}(m_0, \dots, m_7)$ , of the pixel  $(x, y)$  with directional response  $(m_0, \dots, m_7)$ , is derived using Eq. (5).

$$LDP_{x,y}(m_0, \dots, m_7) = \sum_{i=0}^7 s(m_i - m_k) \times 2^i \quad (5)$$

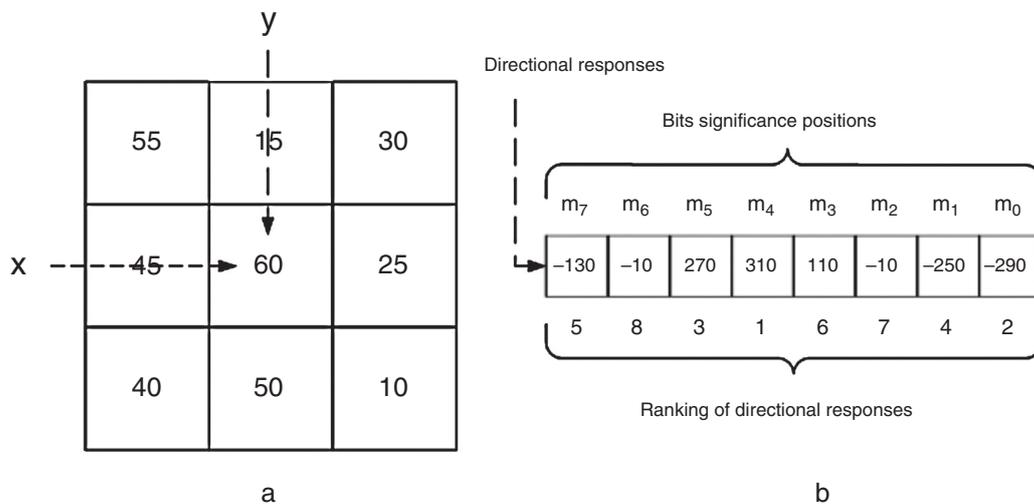


Fig. 2. Kirsch directional response. (a) This figure shows a pixel  $(x,y)$  that has a gray level 60. (b) Directional responses, together with the ranking of those responses, and the associated bit significance, with  $m_0$  being at the less significant position and  $m_7$  at most significant position. Note that the ranking of responses is done on absolute values.

where,  $m_k$  is the  $k$ th most significant response and  $s(x)$  is defined as

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (6)$$

Given the directional responses generated by the Kirsch convolution on pixel  $(x, y)$  presented in Figure 2, the LDP code for  $k=3$  is computed as follows:

- o  $m_5 = 270$  is the 3rd most significant directional response.
- o The code of the LDP code of the pixel  $(x, y)$  is then

$$\begin{aligned} LDP_{x,y}(m_0, \dots, m_7) &= \sum_{i=0}^7 s(m_i - m_5) \times 2^i \\ &= 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 \\ &\quad + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 \\ &\quad + 0 \times 2^1 + 1 \times 2^0 \end{aligned}$$

- o The LDP code,  $LDP_{x,y}$ , of the pixel  $(x, y)$  is then 49.

- **Construction of LDP descriptor** which is carried out after the calculation of the LDP code for each pixel  $(x, y)$ . The input image  $I$  of size  $M \times N$  is then represented by a LDP histogram using Eq. (7), that is also called LDP descriptor. In this case  $k=3$ , is used; It means,  ${}^8C_3 = 56$  distinct values are generated and used to encode the image. The histogram  $H$  obtained from the transformation has 56 bins and can be defined as

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(LDP_{(x,y)}, C_i) \quad (7)$$

where  $C_i$  is the  $i$ th LDP pattern value,  $i = 1, \dots, {}^8C_3$  and the definition of  $p$  is given in Eq. (9).

$$p(x, a) = \begin{cases} 1, & x = a \\ 0, & x \neq a \end{cases} \quad (8)$$

Given a texture,  $T$ , and the number of significant bits  $k$ , a feature vector  $LDP_{k,T}$  can be extracted and represented as

$$LDP_{k,T} = (H_1, \dots, H_{56}) \quad (9)$$

### 2.3. Classifiers

Four classifiers are used to evaluate the power of LDP to characterize textures and compared Haralick features extracted from GLCM.

#### 2.3.1. Support vector machines

SVM is a learning technique for pattern classification and regression (Cortes & Vapnik, 1995; Vapnik, 2013). It was originally designed as two-class classifier, but many versions have been proposed to perform multi-class classification (Crammer & Singer, 2002; Hsu & Lin, 2002). The principle is, given a labeled set of  $M$  training samples  $(x_i, y_i)$ , where  $x_i \in R$  and  $y_i$  is the associated label ( $y_i \in \{-1, 1\}$ ),  $i = 1, \dots, M$ . A SVM classifier finds the optimal hyperplane that correctly separates the

largest fraction of data points while maximizing the distance of either class from the hyperplane. The discriminant hyperplane is defined by the level set function

$$f(x) = \sum_{i=1}^M y_i \alpha_i k(x, x_i) + b \quad (10)$$

where  $k(\cdot, \cdot)$  a kernel function and the sign of  $f(x)$  indicates the membership of  $x$ . Constructing an optimal hyperplane is equivalent to finding all nonzero  $\alpha_i$ . Kernel function  $K(x_i, x_j)$  is the inner product of the features space  $K(x_i, x_j) \leq \langle x_i, x_i \rangle \langle x_j, x_j \rangle$ . The three following kernel functions are often used:

- Polynomial kernel

$$K(x_1, x_2) = (\langle x_1, x_2 \rangle + d)^d \quad (11)$$

where  $c$  is a positive constant and  $d$  is the dimension of feature space in question.

- Linear kernel

$$K(x_1, x_2) = \langle x_1, x_2 \rangle \quad (12)$$

- Radial basic function kernel (RBF)

$$K(x_1, x_2) = \exp\left(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}\right) \quad (13)$$

where  $\|x_1 - x_2\|$  is the distance between the vectors  $x_1$  and  $x_2$ ,  $\sigma \in R$  is the bandwidth of a gaussian curve.

In our case, the  $M$  feature vectors ( $x_i = 1, \dots, M$ ) are extracted, from texture images that need to be classified, using the local directional pattern, or GLCM.

#### 2.3.2. Naïve Bayes classifier

One of the most popular and most simplest classification models is the naive Bayes classifier (Friedman, Geiger, & Goldszmidt, 1997). The principle of Naive bayes is: given the training data  $T$  which contain a set of samples, each sample  $X = (x_1, \dots, x_n)$  and there are  $k$  classes  $C_1, \dots, C_k$ . Each sample is labeled by one of these classes. Naive Bayes predicts a given sample  $X$  belongs to the class that has the highest posterior probability conditioned on  $X$ . Therefore, sample  $X$  is predicted to belong to class  $C_i$  if and only if  $P(C_i|X) > P(C_j|X)$ , for all  $j$  such that  $0 < j < (m - 1)$  and  $j \neq i$ . By Bayes' theorem

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (14)$$

If the data set has many attributes, it would be expensive to compute  $P(X|C_i)$ . To solve this problem, naive assumption assumes that the value of the attributes are conditionally independent of one another. This means that

$$P(X|C_i) \approx \prod_{k=1}^n P(x_k|C_i) \quad (15)$$



Fig. 3. Samples FT1 and FT2 are extracts of the formal township, where buildings are placed in a planned manner. This type contains stable structure. The different between T.1 and T.2 is on the houses.

### 2.3.3. *k*-Nearest neighbor (*k*-NN)

*k*-NN is a non-parametric classification method introduced in the early 1970s by Fix and Hodges (1951). The process is done by computing the similarity between the sample and the different classes. Let  $C_1, \dots, C_k$  be the classes of our samples. Given a new sample  $X = (x_1, \dots, x_n)$ , to find to which class it belongs, the distance  $d(x, C_j)$  between  $x$  and  $C_j$ , for  $j = 1, \dots, k$ , is calculated. Sample  $X$  is assigned to class  $C_{i_0}$  to which it is closest. Index  $i_0$  is calculated as presented in the following equation:

$$i_0 = \operatorname{argmin}_{i=1, \dots, k} d(x, C_i) \quad (16)$$

### 2.3.4. Multilayer perceptron (MLP)

MLP is well known and widely used in different detection and estimation applications (Burrascano, Fiori, & Mongiardo, 1999; Gati, Wong, Alquie, & Fouad Hanna, 2000; Kasabov, 1996). The principle of MLP is that the input layer in MLP is considered as layer 0. Assume that the total number of the hidden layers is  $L$ . In the hidden layer  $l$  the number of node is  $N_l$ ,  $l = 1, \dots, L$ . Let  $w_{ij}$  be the weight of the connection between the  $j$ th nodes of  $(l-1)$ th hidden layer and  $i$ th nodes of the  $l$ th hidden layer, and let  $x_i$  be the  $i$ th input factor to the MLP. Let  $y_i^l$  represent the output of the  $i$ th node of

the  $l$ th hidden layer, which can be calculated by the following equation:

$$y_i^l = f \left( \sum_{j=1}^{N_{l-1}} w_{ij}^l \cdot y_j^{l-1} + \theta_i^l \right), \quad i = 1, \dots, N_l \quad (17)$$

where  $\theta_i^l$  represent the bias factor of the  $i$ th node of the  $l$ th hidden layer,  $y_i^0 = x_i$ ,  $i = 1, \dots, N_0$  and  $f(\cdot)$  is the active function. Let  $v_{ki}$  be the weight of the connection between the  $k$ th node of the output layer and the  $i$ th node of the  $L$ th hidden layer. The MLP output can be calculated as

$$y_k = \sum_{i=1}^{N_L} v_{ki} \cdot y_i^L + \beta_k, \quad k = 1, \dots, N_y \quad (18)$$

where the  $\beta_k$  is the bias factor of the output layer. The MLP algorithm compares between the network output with the desire output which measures the error in the network. To correct the output layer, this algorithm updates the weight until the output of network gets closer to the desire output.

## 3. Experimental results and discussion

### 3.1. Data set

Settlements image categories, shown in Figures 3–8, were identified in Soweto (Gauteng province, South Africa) to work



Fig. 4. Informal squatters (IS type1): the structure of the informal squatters is not stable. The dwellings of this category are shack type (made out of cardboard, wood, tin, etc.). Typically characterized by high building densities.



Fig. 5. Formal township + informal squatter (FTIS Type1, FTIS Type2, and FTIS Type3): In this category we can find any type of any density, of residential unit, but buildings appear in pairs a larger building will be accompanied by a backyard shack.



Fig. 6. Informal Township: the structure of the informal township is recognized as constant or semi-constant structure. The dwelling of this category is shack type and located on serviced and un-serviced sites. The dwelling densities vary from low to high.



Fig. 7. Formal: the structure of the formal residential is a constant structure, located near well-established buildings areas.

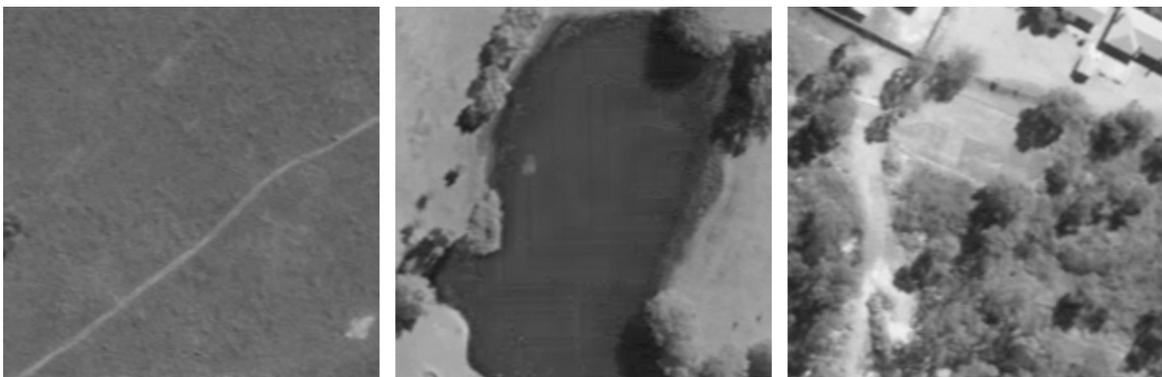


Fig. 8. Non-urban: shows samples of this type. All of them were gray level images with 8 bit per pixel.

Table 2  
Number of the sample in each category.

Samples type	Number
Formal	413
FT1	367
FT2	637
FTIS1	62
FTIS2	611
FTIS3	100
Informal	358
IS1	312
Non-urban	479

with as target classes. Data used are gray-level images with 8 bit per pixel, and a size of (200 × 200) pixels. Table 2 shows the number of samples in each category.

- Formal township (FT1, FT2): The structure of the formal township building is placed in a planned manner. The difference between FT1 and FT2 relates to the size of the houses.
- Informal squatters (IS): The structure of the informal squatters is not stable. The dwellings of this category are shack type (made out of cardboard, wood, tin, etc.). This is typically characterized by high building density.
- Formal township + informal squatter (FTIS1, FTIS2, and FTIS3): In this category we can establish any type of any density, of residential unit, but buildings appear in pairs and a larger building is accompanied by a backyard shack.
- Informal township: The structure of the informal township is recognized as a constant or semi-constant structure. The dwelling of this category is shack type and located on serviced and un-serviced sites. The dwelling density varies from low to high.
- Formal: The structure of the formal residential is a constant structure, located near well establish buildings area.
- Non-urban area: The areas outside a town or a city. Open swath of land that has few homes or other buildings.

### 3.1.1. Choice of parameters

The computation of GLCM is done using the following parameters:

- Number of gray levels: 256
- Directions: 0°, 45°, 90°, 135°
- Distance: 1

The computation of the local directional pattern is based on the number of significant bits, and in this work four alternative values are considered: 2, 3, 4, 5.

### 3.2. Result and discussion

Performance of the various classifiers using various size of training samples, with various values for  $k$  are considered.  $k$ -NN and MLP achieve reasonably good results, even when only a small proportion of the data is used for training. The superiority of the  $k$ -NN and MLP over both SVM and NB

is easily perceptible. Moreover, it can be noticed that when the number of significant bits ( $k$ ) changes from 2 to 4 the accuracy improves and declines when  $k=5$ . Classifiers have the best performance for  $k$  equal 4 and the worst performance is registered for  $k$  equal to 2. The performance of LDP is evaluated for different values of  $k$ : The average accuracy of LDP with different  $k$  values shows that for LDP when  $k$  equal 4 is the highest performance (85.6%) compared to the rest. For LDP, the value 2 for  $k$  is the lowest performance (77.7%). The best achievement was obtained by  $k$ -NN and MLP 98.9%, 98.19% in LDP (4) and LDP (3) when the data proportion is 80% as training 20% testing. NB performances are remarkably low ranging between 55% and 70%. Another observation is the fact that the accuracy of SVM is not very high either. With the gray-level co-occurrence matrix, different combinations of the 14 original Haralick features (entropy, (entropy + IDM), (energy + contrast + correlation), (entropy + energy + contrast), (energy + correlation + entropy + IDM), (entropy + energy + contrast + correlation + IDM), and all 14 features are used to characterize textures. The presentation of different feature combinations to characterize informal settlements reveals that the best performance is achieved by  $k$ -NN with value (86.04%) using the 14 original Haralick features in 45° direction. With NB, the performance is remarkably low ranging between 29% and 53%. Another observation is that both SVM and MLP are not very effective classifying informal settlements, with accuracies ranging from 30% to 77%. Using  $k$ -NN, the best achievement was achieved with 45° direction at 91%. With NB, the performances are remarkably low ranging between 50% and 55%. The accuracy of both SVM and MLP are not very high range between (48%–60%) and (64%–80%). The best performance was achieved in 45° direction with value (80%) using MLP and (60.81%) using SVM. The training set percentage: the average accuracy of the best direction changes from 62% when the training set is 10%–71.17% when the training set is 80%. With the knowledge that the best performance is achieved with GLCM and this when the 14 original Haralick features at 45° direction are chosen, it can be compared to the performance of LDP when  $k$  equals 4. The average performances of each classifier in both feature methods are such that, with NB and SVM, the performance is remarkably low in both LDP (4) and GLCM (45°), the best achievement is (83.06%) by SVM using LDP (4). With MLP, LDP (4) achieved 20% more with value (92.77%) compared to GLCM (45°) with value (77.48%). With  $k$ -NN, The best result was achieved by LDP (4) with value (92.7%), compared to GLCM (45°) with value (86.03%). For GLCM (45), we find the performance from 20% to 80% increase by approximately 1% in each step. The best achievement obtained when the training set is 80% with a value (71.17%). The worst achievement obtained when the training set is 10% with a value (62.6%). For LDP (4), we find the performance almost constant from 50% to 80% with value (86.81%). The best achievement obtained when the training set is 80% with a value (86.81%). Figures 9–10 below summarize the comparison between these two methods (LDP, GLCM). In conclusion, the best combination used in our approach is using the local directional pattern with  $k$  equals four applying on

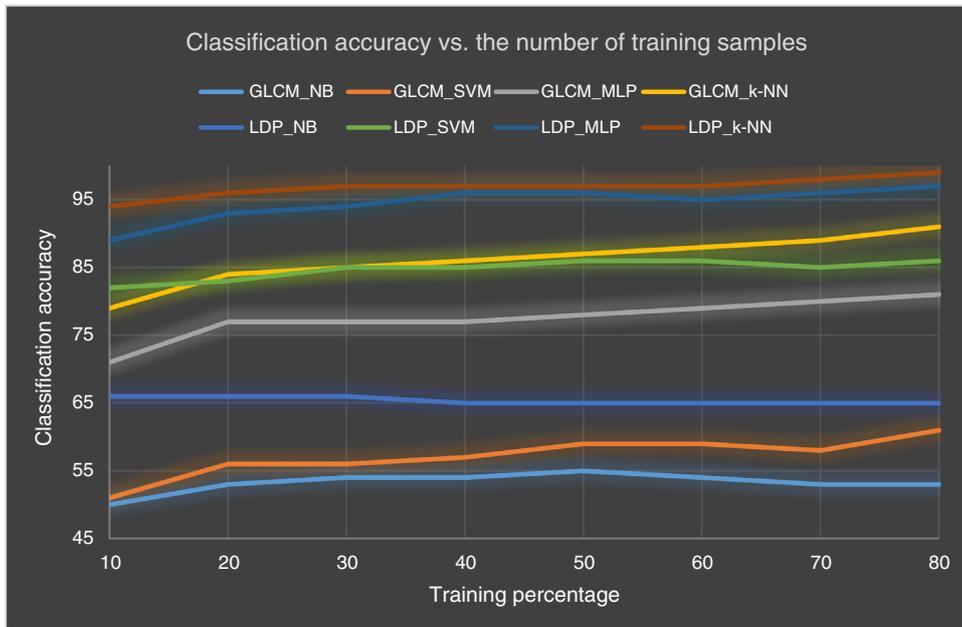


Fig. 9. Classification accuracy vs. the number of training samples using both features method LDP when  $k=4$  and GLCM with distance 1 and  $45^\circ$  direction.

nine categories of image, and the  $k$ -nearest neighbor being the preferred classifier.

#### 4. Running Times for Features Extraction

For an image of size  $n \times m$ , GLCM running time will be  $G(m, n) = O(mn)$ . However, the running time for LDP is  $L(m, n) = R(m, n) + H(m, n)$  where  $R(m, n) = O(mn)$  is the running time to compute the responses and  $H(m, n) = O(mn)$  is the running time of the computation of the histogram, then

$L(m, n) = O(mn)$ . Table 3 shows running times for both GLCM and LDP applied to nine different categories using a computer with a processor Intel Core i5, a CPU of 2.3 GHz and 4G of RAM. With GLCM 14 features were computed and for  $k$  equals 4 for LDP; the running time of the GLCM algorithm is remarkably low (less than 13 ms to process an image). Compared to LDP, which takes a very long time to compute, ranging from 4.7 s in average to process an image in FTIS1 to 9 s in FT2. It is worth mentioning that the size of each image is  $(200 \times 200)$ .

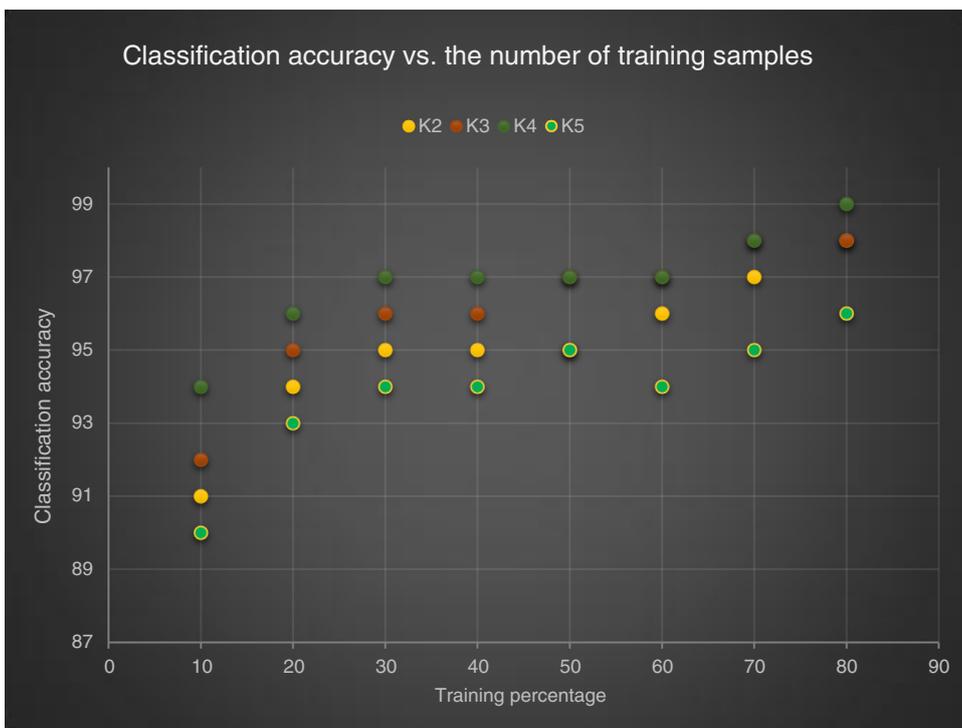


Fig. 10. Classification accuracy vs. the number of training samples using LDP with different number of significant bits ( $k$ ).

Table 3  
Average running time for both GLCM and LDP applied to nine category images.

Image category	GLCM runtime (s)	LDP runtime (s)
Formal	0.014	7.22
FT1	0.013	8
FT2	0.014	9
FTIS1	0.016	4.7
FTIS2	0.013	7.4
FTIS3	0.01	7.93
Informal	0.013	8
IS	0.012	8.7
Non-urban	0.012	6.3

## 5. Conclusion

Two feature methods, local directional pattern (LDP) and gray-level co-occurrence matrix (GLCM), have been compared using four different classifiers (Naive-Bayes, multilayer perceptron, support vector machines, and  $k$ -nearest neighbor). This work has investigated the impact of the number of significant bits considered to code the Kirsch masks application responses. Experiments have shown that the choice of 4 significant bits achieves the best accuracy for texture characterization using LDP. It has also been established that the best texture feature is the local directional pattern when  $k$ -NN is used as a classifier. On the other hand, it has been demonstrated that the local directional pattern is superior in characterizing informal settlement images. However, the running time for LDP is two orders of magnitude higher than that of the GLCM.

## Conflict of interest

The authors have no conflicts of interest to declare.

## References

- Aminipouri, M., Sliuzas, R., & Kuffer, M. (2009). Object-oriented analysis of very high resolution orthophotos for estimating the population of slum areas, case of Dar-Es-Salaam, Tanzania. In *Paper presented at the Proc. ISPRS XXXVIII Conf.*
- Asmat, A., & Zamzami, S. Z. (2012). Automated house detection and delineation using optical remote sensing technology for informal human settlement. *Procedia-Social and Behavioral Sciences*, 36, 650–658.
- Bastos, L. D. O., Liatsis, P., & Conci, A. (2008). Automatic texture segmentation based on  $k$ -means clustering and efficient calculation of co-occurrence features. In *15th International Conference on Systems, Signals and Image Processing, 2008. IWSSIP 2008* (pp. 141–144). IEEE.
- Blaschke, T., & Lang, S. (2006). Object based image analysis for automated information extraction – A synthesis. In *Paper presented at the Measuring the Earth II ASPRS Fall Conference.*
- Burrascano, P., Fiori, S., & Mongiardo, M. (1999). A review of artificial neural networks applications in microwave computer-aided design (invited article). *International Journal of RF and Microwave Computer-Aided Engineering*, 9(3), 158–174.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Crammer, K., & Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2, 265–292.
- Eleyan, A., & Demirel, H. (2011). Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering & Computer Sciences*, 19(1), 97–107.
- Ella, L. A., Van Den Bergh, F., Van Wyk, B. J., & Van Wyk, M. A. (2008). A comparison of texture feature algorithms for urban settlement classification. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International, Vol. 3, no. 1. IEEE*, pp. III-1308.
- Ferrer, M. A., Vargas, F., Travieso, C. M., & Alonso, J. B. (2010). Signature verification using local directional pattern (LDP). In *IEEE International Carnahan Conference on Security Technology (ICCST), 2010* (pp. 336–340). IEEE.
- Fix, E., & Hodges, J. L., Jr. (1951). *Discriminatory analysis-nonparametric discrimination: Consistency properties*. DTIC Document.
- Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29(2–3), 131–163.
- Gati, A., Wong, M., Alquie, G., & Fouad Hanna, V. (2000). Neural networks modeling and parameterization applied to coplanar waveguide components. *International Journal of RF and Microwave Computer-Aided Engineering*, 10(5), 296–307.
- Graesser, J., Cheriadat, A., Vatsavai, R. R., Chandola, V., Long, J., & Bright, E. (2012). Image based characterization of formal and informal neighborhoods in an urban landscape. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(4), 1164–1176.
- Haralick, R. M., Shanmugam, K., & Dinstein, I. H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6), 610–621.
- Hay, G. J., & Castilla, G. (2008). Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline. *Object-based Image Analysis*, 75–89.
- Hsu, C.-W., & Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2), 415–425.
- Jabid, T., Kabir, M. H., & Chae, O. (2010a). Gender classification using local directional pattern (LDP). In *20th International Conference on Pattern Recognition (ICPR), 2010* (pp. 2162–2165). IEEE.
- Jabid, T., Kabir, M. H., & Chae, O. (2010b). Local directional pattern (LDP) for face recognition. In *2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE)* (pp. 329–330). IEEE.
- Jabid, T., Kabir, M. H., & Chae, O. (2010c). Robust facial expression recognition based on local directional pattern. *ETRI Journal*, 32(5), 784–794.
- Kabir, M. H., Jabid, T., & Chae, O. (2010). A local directional pattern variance (LDPv) based face descriptor for human facial expression recognition. In *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 526–532). IEEE.
- Kasabov, N. K. (1996). *Foundation of neural networks, fuzzy systems, and knowledge engineering*. Cambridge, MA: MIT.
- Khumalo, P. P., Tapamo, J. R., & Van Den Bergh, F. (2011). Rotation invariant texture feature algorithms for urban settlement classification. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International* (pp. 511–514).
- Mayunga, S., Coleman, D., & Zhang, Y. (2007). A semi-automated approach for extracting buildings from QuickBird imagery applied to informal settlement mapping. *International Journal of Remote Sensing*, 28(10), 2343–2357.
- Owen, K. K., & Wong, D. W. (2013). An approach to differentiate informal settlements using spectral, texture, geomorphology and road accessibility metrics. *Applied Geography*, 38(1), 107–118.
- McLaren, R., Coleman, D., & Mayunga, S. (2005). Sustainable management of mega growth in megacities. In *Paper presented at the International Federation of Surveyors, FIG.*
- Shabat, A. M., & Tapamo, J. R. (2014). A comparative study of local directional pattern for texture classification. In *World Symposium on Computer Applications and Research (WSCAR), 2014.*
- Van Den Bergh, F. (2011). The effects of viewing-and illumination geometry on settlement type classification of quickbird images. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International* (pp. 1425–1428).
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

## **Chapter 4**

# **Directional Local Binary Pattern for Texture Analysis**

# Directional Local Binary Pattern for Texture Analysis

Abuobayda M. Shabat and Jules-Raymond Tapamo<sup>(✉)</sup>

School of Engineering, Howard College Campus, University of KwaZulu-Natal,  
Durban 4041, South Africa  
abshabat@gmail.com, tapamoj@ukzn.ac.za

**Abstract.** In this paper, a new features method, the Directional Local Binary Pattern (DLBP), is presented, with an objective to improve Local Directional Pattern (LDP) for texture analysis. The idea of Directional DLBP is inspired by the stability of the Kirsch mask directional responses and the LBP neighboring concept. The result shows that Directional Local Binary Pattern outperforms LDP and LBP.

**Keywords:** Texture features · Local Directional Pattern · Directional Local Binary Pattern · Local Binary Pattern · Classification

## 1 Introduction

The main goal of texture analysis is to quantify the different qualities of an image, such as smoothness, roughness, and bumpiness. This is modeled as a spatial variation in pixel gray values. Texture represents a basic level of spatial properties of a digital image, and can be defined as relationship between gray levels in neighboring pixels [2]. Texture analysis has been applied in several areas, including medical image analysis, biometrics, and security.

Gray Level Co-occurrence Matrix (GLCM) is one of the commonly used textures based features extraction techniques. Haralick et al. [3] proposed it in the early 1970s. It has since been used in many applications.

A study by Song et al. [12] used LBP operator to analyze textures of multi-spectral images. The technique used achieved more than 4% gain in the performance, compared to the popular GLCM method. Musci et al. [7] investigated the use of Local phase quantization (LPQ) and LBP to characterize land-cover and land-use. The result establishes that both LBP and LPQ outperform GLCM.

The successful application of the LBP inspired many scholars for further research. Several adjustments of LBP have been proposed [1, 8, 10]. However, LBP suffers from random noise, because it depends on neighboring pixels intensity. A more stable technique, based on Kirsch masks, Local Directional Pattern was recently presented by Jabid et al. [4]. LDP considers the edge response values in eight directions around the pixels obtained from the Kirsch gradient operator rather than the raw pixel intensities like LBP. LDP has been applied

in many areas, including texture classification [11], and facial expression interpretation [4]. One of the drawbacks of LDP is the number of significant bits,  $k$ , considered after the generation of Kirsch mask responses. The choice  $k = 3$ , as established in the literature is empirical. A careful investigation revealed that the change in the value of  $k$  affects the performance of LDP. DLBP takes the best of both LDP and Local Binary Pattern (LBP), by first computing the gradient directional responses since it is more stable than the local neighboring following LDP concept. And generates the code following the LBP concept.

## 2 Features Methods for Texture Analysis

Features extraction is one of the key processes in texture analysis. In this paper, Local Binary Pattern (LBP), Local Directional Pattern (LDP) together with the proposed Directional Local Binary Pattern (DLBP) are presented.

### 2.1 Local Binary Pattern

Local binary pattern introduced by Ojala et al. [8] is inspired by the general definition of texture in the local neighbourhood. Given an image of size  $R \times C$ , for each pixel  $p = (x, y)$ , where  $0 \leq x \leq R$  and  $0 \leq y \leq C$ , the LBP code of  $p$  is computed as

$$LBP_N(x, y) = \sum_{i=0}^{N-1} S(g_i - g_p)2^i \quad (1)$$

where  $g_p$  and  $g_i$  are the gray levels of pixel  $p$  and its  $i^{th}$  neighbor, respectively; and the function  $S(x)$  is defined as

$$S(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

If for each pixel,  $N$  neighbors are considered, we can have  $2^N$  distinct values for the LBP code. It means a gray-scale image representing a texture can be characterized using a  $2^N$ -bin discrete distribution.

### 2.2 Local Directional Pattern

Local Directional Pattern (LDP) was introduced by Jabid et al. [4]. It has mostly been used in face based biometrics and has received little attention from other areas. LDP descriptors of an image are calculated using eight bit binary codes generated from Kirsch masks application on each pixel of this image. Detailed description of the three steps used to calculate LDP descriptors of an image  $I$  is given below [5]:

1. **Computation of response values using Kirsch mask application:** for each pixel  $(x, y)$ , Kirsch mask convolution response values,  $(K_{M_0}(x, y), K_{M_1}(x, y) \dots, K_{M_7}(x, y))$ , are generated using Eq. 3 as

$$K_{M_q}(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 M_q(i, j) \times I(x + i, y + j) \quad (3)$$

where  $K_{M_q}(x, y)$  represents the response value at direction  $M_q$  (see Fig. 1), for  $q = 0, 1, \dots, 7$ .  $K_{M_q}$ , for  $q = 0, 1, \dots, 7$  are then allocated ranks based on their absolute values as shown in Fig. 2(b). In the rest of the text  $K_{M_q} = m_q$

2. **Generation of LDP code:** assuming that  $k$  significant bits will be considered, from Kirsch mask convolution responses generated in the previous step set to 1 the corresponding bit positions of the  $k$  most significant responses, and leave other  $(8 - k)$  bits to 0. This process is implemented by the function  $S(x)$  defined in Eq. 2. The resulting LDP code of the pixel  $(x, y)$ ,  $LDP_{x,y}(m_0, m_1, \dots, m_7)$ , can be derived as

$$LDP_{x,y}(m_0, m_1, \dots, m_7) = \sum_{i=0}^7 S(m_i - ms_k) \times 2^i \quad (4)$$

where  $ms_k$  is the  $k^{th}$  most significant response and  $S(x)$  is defined in Eq. 2. Considering the Kirsch mask application on pixel  $(x, y)$  shown in Fig. 2, the LDP code for  $k = 3$  is generated as follows:

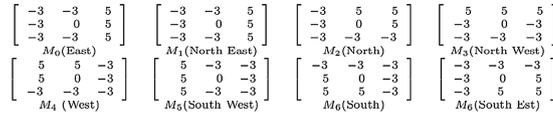


Fig. 1. Kirsch masks

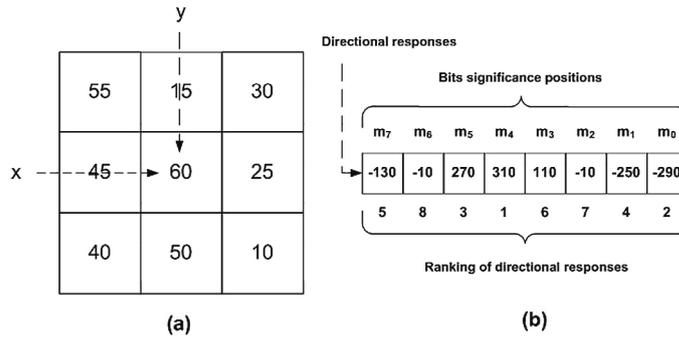


Fig. 2. Kirsch mask application response value of a pixel  $(x, y)$  with a gray value 60. (a) pixel  $(x, y)$  with the 8-neighborhood. (b) The middle row shows the directional response values, the row below represent the ranking of those responses. The ranking of responses is considered with the responses in absolute values.

- The response values are  $(m_0, \dots, m_7) = (-130, -10, 270, 310, 110, -10, -250, -290)$ , and with 310, -280, 270, being the largest values in absolute term, making  $m_4, m_0$  and  $m_5$  the most significant bits.  $m_5 = 270$  is the 3<sup>th</sup> most significant directional response value.
- The LDP code,  $LDP_{x,y}$ , of the pixel  $(x, y)$  is then 49.

$$\begin{aligned}
 LDP_{x,y}(m_0, m_1, \dots, m_7) &= \sum_{i=0}^7 S(m_i - m_{s_k}) \times 2^i \\
 &= 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 \\
 &\quad + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 \\
 &\quad + 0 \times 2^1 + 1 \times 2^0 \\
 &= 49
 \end{aligned}$$

3. **Production of the LDP descriptor:** Given an image  $I$  of size  $M \times N$ , the LDP code of I, denoted by  $LDP(I)$  is defined as

$$LDP(I) = (LDP_{x,y})_{0 \leq x \leq M-1, 0 \leq y \leq N-1} \quad (5)$$

LDP histogram can then be generated using Eq. 6, that is also called LDP descriptor. With  $k = 3$ , there are  $56 (= {}^8 C_3)$  distinct values generated and used to encode the image. The histogram  $H$ , with 56 bins, used to represent the image is generated using Eq. 6.

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(LDP_{x,y}, C_i) \quad (6)$$

where  $C_i$  is the  $i^{th}$  LDP component,  $i = 1, \dots, {}^8 C_3$  and  $p$  is defined as

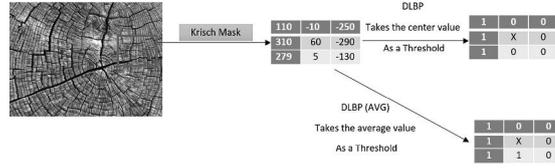
$$p(x, a) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

Given a texture,  $T$ , and the number of significant bits  $k$ , a feature vector  $ldb_{k,T}$  is generated and represented as

$$ldb_{k,T} = (H_1, H_2, \dots, H_{56}) \quad (8)$$

### 2.3 Directional Local Binary Pattern

LDP proposed to solve the problem with LBP. In fact, LBP depends on neighboring pixels intensity which makes it unstable. Instead, LDP considers the edge response value in different directions. As it is well known, gradients are more stable than the gray levels. But the problem with LDP is to select the value of the number of significant bit  $k$ . The value  $k = 3$  has widely been used in literature. Through our research, we established that the change in the value of the



**Fig. 3.** The figure shows the computation of DLBP, by first calculating the eight directional edge responses and then generate the DLBP code in two different way, first, by using the center value as a threshold, secondly, using the average value as a threshold.

k affects performance. Our proposed method, Directional Local Binary Pattern (DLBP) takes the best of both LDP and LBP, by first computing the gradient directional responses since it is more stable than the local neighboring following LDP concept. And generates the code following the LBP concept.

The DLBP features are an eight binary code assigned to each pixel of an input window. DLBP descriptors are calculated in three steps:

1. Computation of Kirsch kernel application response values is similar to LDP.
2. In this step, two ways are proposed to generate DLBP code,
  - (a) **Generation of DLBP code including the center pixel:** It is based on the values generated in the first step. For each pixel  $(x, y)$ , its binary DLBP code can be generated by comparing the value,  $v(x, y)$ , to the response values  $m_0, \dots, m_7$ . The DLBP code,  $DLBP_{x,y}(m_0, \dots, m_7)$ , of the pixel  $(x, y)$  can then be calculated using Eq. 9.

$$DLBP_{x,y}(m_0, m_1, \dots, m_7) = \sum_{i=0}^7 S(m_i - v(x, y)) \times 2^i \quad (9)$$

- (b) **Generation of the DLBP code,  $DLBP(AVG)_{x,y}$ , based on the average Kirsch mask application response values:** The average value is computed based on first step (see Eq. 10). This average is used as a threshold of  $m_0, m_1, \dots, m_7$ . The DLBP code,  $DLBP(AVG)_{x,y}$ , of the pixel  $(x, y)$  for the directional response  $(m_0, \dots, m_7)$ , is computed using Eq. 11.

$$AVG_{x,y} = \frac{\sum_{i=0}^7 m_i + v(x, y)}{9} \quad (10)$$

$$DLBP(AVG)_{x,y} = \sum_{i=0}^7 S(m_i - AVG_{x,y}) \times 2^i \quad (11)$$

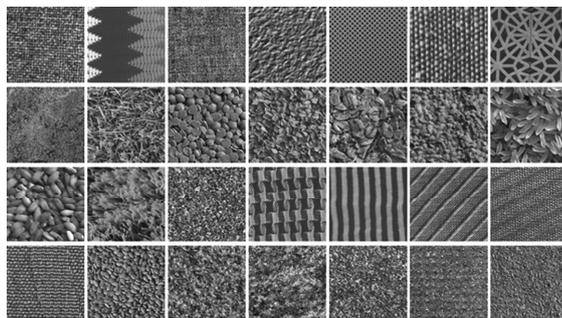
Figure 3 shows an example of computation of DLBP code of a pixel.

3. The production of DLBP descriptor is similar to that of LDP descriptor using DLBP code generated in the previous step.

### 3 Experiments

#### 3.1 Data Set

In our experiment, we gathered 3200 texture images from the Kylberg texture dataset [6]. These images are divided into 20 categories, each categories has 160 images. All the selected image have the size of  $576 \times 576$ . In Fig. 4 a sample of each category is shown. We used python-fortran framework to implement the proposed features using opencv and scikit-learn toolkit [9]. In this experiment, there are two main components in textural classification: Feature Extraction and Features Classification. During the features extraction stage all the proposed methods are calculated for each Image. In Features Classification, each image is classified according to the extracted features using 20 % as a test data set and the remains as a training data set.



**Fig. 4.** The sample images of texture from Kylberg

#### 3.2 Results and Discussion

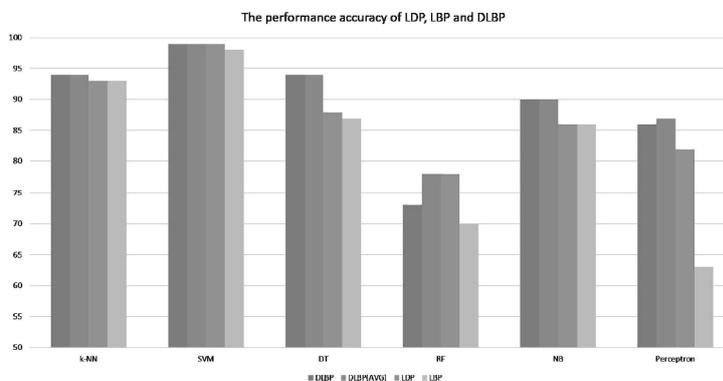
In this section, a comparison between Local Directional Pattern, Local Binary Pattern and the presented method Directional Local Binary pattern.

The performance of LDP, LBP, DLBP and DLBP(AVG) will be compared using 6 different classifiers (K-neareast neighbor algorithm (k-NN), Support Vector Machine (SVM), Perceptron, Naive-Bayes (NB), Decision Tree (DT)) in different conditions.

Table 1 and Fig. 5 show the accuracies of LDP, LBP, DLBP and DLBP(AVG) using six different classifiers. Both DLBP and DLBP(AVG) methods are the best in performance using k-NN classifier. As the performance increases by 1 % compared to LDP and LBP. It can also be observed that the worst performances were achieved by LDP and LBP with accuracy of 93 %. For SVM classifier the three features methods LDP, DLBP and DLBP(AVG) performed equally, with an accuracy of 99 %, except LBP with accuracy of 98 %. With the DT classifier,

**Table 1.** Accuracies measures in % of LDP, LBP, DLBP and DLBP(AVG) using six different classifiers,  $Gain_A$  is the gain when using average DLBP, and  $Gain_C$  is the gain when using center DLBP

		Feature methods					
		DLBP	DLBP(AVG)	LDP	LBP	$Gain_C$	$Gain_A$
Classifiers	<b>k-NN</b>	<b>94</b>	<b>94</b>	93	93	+1	+1
	<b>SVM</b>	<b>99</b>	<b>99</b>	<b>99</b>	98	0	+1
	<b>DT</b>	<b>94</b>	<b>94</b>	88	87	+6	+7
	<b>RF</b>	73	<b>78</b>	<b>78</b>	70	+5	+8
	<b>NB</b>	<b>90</b>	<b>90</b>	86	86	+4	+4
	<b>Perceptron</b>	86	<b>87</b>	82	63	+5	+14



**Fig. 5.** The performance accuracy of LDP, LBP, DLBP and DLBP(AVG) using six different classifiers (Color figure online)

there was an increase by 6 % in the performance, as both DLBP and DLBP(AVG) performed very well (94 %), compared to LDP (88 %) and LBP (87 %). Using RF classifier, DLBP achieved the best performance with 78 %, yielding an increase of 5 %. Both NB and Perceptron Classifiers better accuracies using DLBP, with increases 6 % and 5 %, respectively. Table 1 shows that the best accuracies were achieved by DLBP(AVG) when using the Kylberg texture dataset. Improvements in performances range from 1 % to 5 %.

## 4 Conclusion

Directional Local Binary Pattern (DLBP) has been proposed, which enables users to select the number of significant bits for the coding features as it is done with LDP. DLBP builds on the strengths of both LDP (Local Directional Pattern) and Local Binary Pattern (LBP). It first computes the gradient directional responses since it is more stable than the local neighboring following LDP

concept. It then generates the code following the LBP concept. The performances of LDP, LBP and DLBP were evaluated using Kylberg texture dataset of 3200 texture images using six different classifiers (K-nearest neighbor algorithm (KNN), Support Vector Machine (SVM), Perceptron, Naive-Bayes (NB), Adaboost, Decision Tree (DT)) in different conditions. Results show that DLBP outperforms the existing LDP and LBP.

## References

1. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(12), 2037–2041 (2006)
2. Burrascano, P., Fiori, S., Mongiardo, M.: A review of artificial neural networks applications in microwave computer-aided design (invited article). *Int. J. RF Microw. Comput. Aided Eng.* **9**(3), 158–174 (1999)
3. Haralick, R.M., Shanmugam, K., Dinstein, I.H.: Textural features for image classification. *IEEE Trans. Syst. Man Cybern. SMC.* **3**(6), 610–621 (1973)
4. Jabid, T., Kabir, M.H., Chae, O.: Local Directional Pattern (LDP) for face recognition. In: 2010 Digest of Technical Papers International Conference on Consumer Electronics (ICCE) (2010)
5. Jabid, T., Kabir, M.H., Chae, O.: Robust facial expression recognition based on local directional pattern. *ETRI J.* **32**(5), 784–794 (2010)
6. Kylberg, G.: Kylberg texture dataset v. 1.0 (2011)
7. Musci, M., Queiroz Feitosa, R., Costa, G.A., Fernandes Velloso, M.L.: Assessment of binary coding techniques for texture characterization in remote sensing imagery. *Geosci. Remote Sens. Lett. IEEE* **10**(6), 1607–1611 (2013)
8. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7), 971–987 (2002)
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
10. Pietikäinen, M., Hadid, A., Zhao, G., Ahonen, T.: *Computer Vision Using Local Binary Patterns*. Computational Imaging and Vision, vol. 40. Springer, New York (2011)
11. Shabat, A.M., Tapamo, J.R.: A comparative study of local directional pattern for texture classification. In: 2014 World Symposium on Computer Applications & Research (WSCAR), pp. 1–7 (2014)
12. Song, C., Yang, F., Li, P.: Rotation invariant texture measured by local binary pattern for remote sensing image classification. In: 2010 Second International Workshop on Education Technology and Computer Science, pp. 3–6 (2010)



## **Chapter 5**

# **An Improved Scheme of Local Directional Pattern for Texture Analysis**

# An Improved Scheme of Local Directional Pattern for Texture Analysis with an Application to Facial Expressions

Abuobayda M. Shabat and Jules-Raymond Tapamo

School of Engineering, Howard College Campus, University of KwaZulu-Natal,  
Durban 4041, South Africa

abshabat@gmail.com, tapamoj@ukzn.ac.za

**Abstract.** In this paper, several extensions and modifications of Local Directional Pattern (LDP) are proposed with an objective to increase its robustness and discriminative power. Typically, Local Directional pattern generates a code based on the edge response value for the eight directions around a particular pixel. This method ignores the center value which can include important information. LDP uses absolute value and ignores sign of the response which carries information about image gradient and may contain more discriminative information. The sign of the original value carries information about the different trends (positive or negative) of the gradient and may contain some more data. Centered Local Directional Pattern (CLDP), Signed Local Directional Pattern (SLDP) and Centered-SLDP (CSLDP) are proposed in different conditions. Experimental results on 20 texture types using 5 different classifiers in different conditions shows that CLDP in both upper and lower traversal and CSLDP substantially outperforms the formal LDP. All the proposed methods were applied to facial expression emotion application. Experimental results show that SLDP and CLDP outperform original LDP in facial expression analysis.

**Keywords:** Texture features. Local Directional Pattern. Centered Local Directional Pattern. Signed Local Directional Pattern. Centered-SLDP. Classification. Facial expression.

## 1 Introduction

In computer vision, texture is a very significant aspect. It presents information about the spatial properties like color or pixel intensity which can be extracted from an image. It describes the relationship between the value of the gray value in a particular pixel and its neighbors. Several textural features methods have been proposed, including Gray Level co-occurrence Matrix (GLCM) [1], Grey-Level Run Length Matrix (GLRLM) [2], Gabor Filter [3], Local Binary pattern [4] and Local Directional Pattern [5] and many others. Local Binary Pattern (LBP) was an eye catching for its simplicity and excellent accuracy in extracting data from images. It has been used in various applications, such as face and

hand recognition [6]. The LBP operator generates binary digits, from the binary derivation that describes the neighboring pixels, which is utilized as an integral measure for regional image contrast. It takes the center value as a threshold for the regional  $3 \times 3$  neighboring pixels, hence generating one binary digit if the neighbor pixel is larger or equal to the threshold, otherwise it generates zero binary digit. Inspired by the success of LBP, many scholars have proposed several adjustments of LBP [7–10]. However, LBP is sensitive to illumination changes and noise.

A stabler feature method, based on the computation of the directional information, Local Directional Pattern was proposed by Jabid et al. [5]. Unlike LBP which is based on computing the pixel intensity, LDP computes the directional information around the pixel using a gradient operator. It has been used in various applications; signature verification [11], face recognition [12], and facial expression recognition [5]. Despite the great achievement of LDP in pattern recognition and computer vision, its fundamental working mechanism still needs more investigation. Zhong [13] proposed an Enhanced Local Directional Pattern (ELDP) by taking the two most prominent directional edge response value. ELDP code is then generated by converting the two values into an octal digit. The result established the robustness of ELDP against non-illumination changes. LDP codes generation is based on the edge response values in the eight directions around the central pixel, but it doesn't take into account the center pixel value. Due to the fact that the center pixel is a very significant factor, ignoring it may lead to a critical lost of information. Centered Local Directional Pattern (CLDP) is proposed with an aim to include the center pixel value based on its relation with the neighboring pixels. Another issue with the classical LDP is that it is encoded using the absolute value, however, the sign of the original value indicates a trend (positive or negative) which may hold more information and it is applied in the proposed Signed Local Directional Pattern (SLDP) method.

The remainder of this study is organised as follows. In section two, Local Directional Pattern, CLDP, SLDP and CSLDP, for texture analysis are presented. Section three, is devoted to the evaluation of discrimination performances of feature methods presented. Section four presents conclusion and recommendations of the study.

## 2 Local features for texture analysis

In this section, the original LDP is presented together with the proposed methods, CLDP, SLDP and CSLDP.

### 2.1 Local Directional Pattern

LBP is considered unstable because it extensively depends on the neighboring pixels intensity which makes it vulnerable and sensitive to random noise. To overcome these problems, Jabid et al. [5] proposed the Local Directional Pattern (LDP). Unlike LBP which considers the intensities of the neighboring pixels,

LDP considers edge response values in eight different directions. To calculate the eight directional edge response values of a particular pixel, Kirsch masks, shown in Fig 1, are used.

$$\begin{array}{cccc}
\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
M_0(\text{East}) & M_1(\text{North East}) & M_2(\text{North}) & M_3(\text{North West}) \\
\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\
M_4(\text{West}) & M_5(\text{South West}) & M_6(\text{South}) & M_6(\text{South Est})
\end{array}$$

Fig. 1: Kirsch masks

Given a pixel  $(x, y)$  of an image,  $\mathbf{I}$ . For each direction  $i$  and using the corresponding mask  $M_i$ , the  $i^{\text{th}}$  directional response  $m_i(x, y)$  can be computed as

$$m_i(x, y) = \sum_{k=-1}^1 \sum_{l=-1}^1 M_i(k+1, l+1) \times I(x+k, y+l) \quad (1)$$

A vector  $(m_0, \dots, m_7)$  is derived from each of the eight directions, where for each pixel  $(x, y)$   $m_i$  represents  $m_i(x, y)$ . The  $k$  most significant responses are selected from the directional response vector. Thus, placing the corresponding positions to 1 bit code, leaving the remaining  $(8 - k)$  to 0 bit code. The LDP code,  $LDP_{x,y}$ , of the pixel  $(x, y)$  base on the directional response  $(m_0, \dots, m_7)$ , is derived using Equation 2.

$$LDP_{x,y}(m_0, m_1, \dots, m_7) = \sum_{i=0}^7 s(m_i - m_k) \times 2^i \quad (2)$$

where  $m_k$  is the  $k^{\text{th}}$  most significant response and  $s(x)$  is defined as

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

From the LDP transformed image, an histogram,  $H$ , is extracted.  $H$  is defined as

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(LDP_{x,y}, C_i) \quad (4)$$

Where  $C_i$  is the  $i^{\text{th}}$  LDP pattern value,  $i = 1, \dots, 8$ ,  $C_3$  and  $p$  is given as

$$p(x, a) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases}$$

## 2.2 Centered Local Directional Pattern

The original LDP codes are generated based on the value of the edge response in eight directions around pixel, but this method ignores the center pixel value. However, the center pixel may contain more information. In LDP, central pixels are not included; as a result only  ${}^8C_3 = 56$  patterns can be generated. If the central pixel is considered,  ${}^9C_4 = 126$  patterns will be generated. Consequently, CLDP enables the extraction of more information, that will potentially lead to better characterization of visual artefacts. The CLDP feature method is calculated in three steps;

1. Calculate of the eight directional responses. This remains the same as in the first and the second steps in LDP calculation (see Equation 2).
2. Compute the average  $m_{AVG}$  of the the 8 neighbouring pixels (see Equation 5) as a threshold, Generate 1 binary code if the center pixel is greater or equal to the threshold, otherwise generates 0 binary code (see Equation 6).

$$m_{AVG} = \frac{1}{8} \sum_{i=0}^7 m_i \quad (5)$$

$$CLDP_{threshold} = s(m_c - m_{AVG}) \quad (6)$$

Where  $m_c$  is the center pixel value and  $s(x)$  is defined in Equation 3.

3. Using the response values computed in the first step, and the threshold obtained in the previous step, the CLDP can be calculated as

$$CLDP = \sum_{i=0}^7 s(m_i - m_k) \times 2^i + CLDP_{threshold} \quad (7)$$

There are two ways to generate the binary code in CLDP, depending on the direction:

- *CLDP-UP*: in this option, the computation of the binary code begins from the center pixels and walks up (anticlockwise) through all the neighbouring pixels, in the following order  $(m_c, m_0, \dots, m_7)$  as shown in Fig 2. The  $CLDP_{UP}$  can be calculated as

$$CLDP_{UP} = \sum_{i=0}^7 s(m_i - m_k) \times 2^i + s(m_c - m_{AVG})$$

- *CLDP-Down*: in this other option, the computation of the binary code begins from the central pixels and walk down (clockwise) through all the neighbouring pixels, in the following order  $(m_c, m_0, m_7, \dots, m_1)$  as shown in Fig 2.  $CLDP_{UP}$ , as represented, starts from the center pixel  $m_0$  and walks up through all neighboring pixels; then  $CLDP_{Down}$  starts from the center pixel  $m_0$  and walks down through all neighboring pixels.  $CLDP_{Down}$  can be calculated as

$$CLDP_{Down} = \sum_{i=0}^7 s(m_{7-i} - m_k) \times 2^i + s(m_c - m_k) \times 2 + s(m_c - m_{AVG})$$

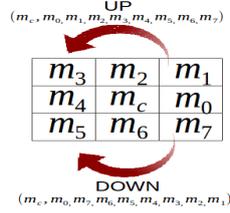


Fig. 2: Representation of  $CLDP_{UP}$  and  $CLDP_{Down}$

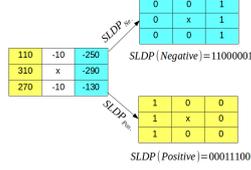


Fig. 3: A computing example for Signed Local Directional Pattern

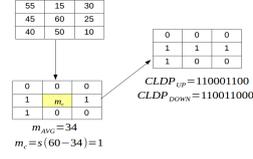


Fig. 4: A computing example for Centered Local Directional Pattern

Fig. 4 shows an example of computing the Centered Local Directional Pattern. To construct the CLDP descriptor, the calculation of the CLDP code for each pixel  $(x, y)$  is necessary. The histogram  $H$  obtained from the transformed Image can be defined as

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(CLDP_{(x,y)}, C_i) \quad (8)$$

### 2.3 Signed Local Directional Pattern

LDP and SLDP are for most of the steps similar; the only difference thing is the way the most prominent edges are chosen. On the LDP, the selection of the most prominent edges is based on the absolute values, while for the SLDP the signs (positive or negative) of edges are considered. Typically, LDP is encoded using the absolute value, however the sign of the original value indicates the trends (positive or negative) of the gradient and may hold more data. SLDP features are based on eight bit binary codes assigned to each pixel of an input image.

In case of the positive trends, three of the most prominent edges are chosen as calculated in equation 9.

$$SLDP(Pos)_{x,y} = \sum_{i=0}^7 s(m_i - m_k) \times 2^i \quad (9)$$

On the other hand, in the negative trends are calculated as:

$$SLDP(Neg)_{x,y} = \sum_{i=0}^7 s((-m_i) - m_k) \times 2^i \quad (10)$$

where  $s$  is defined a in Equation 3. Fig.3 shows an example of computing SLDP. Histogram  $H$  for both directions are obtained from the transformation has 56 bins and can be defined as

$$H_i(P/N) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(SLDP_{(x,y)}(P/N), C_i) \quad (11)$$

## 2.4 Centered Signed Local Directional Pattern

The SLDP codes are generated based on two different directions (positive or negative) of the gradients. However, this method ignores the center pixel value. In the original SLDP,  ${}^8C_3 = 56$  patterns only are generated, but if the center pixel is included  ${}^9C_4 = 126$  patterns will be generated, which will include more information. Center-SLDP (CSLDP), adds the center pixel for both directions (positive or negative). Their computation is done in two steps:

1. Compute SLDP as in the subsection 2.3 above.
2. CSLDP coding of the directional response is generated based on the first step and the calculation of the center pixel obtained in subsection 2.2. The CSLDP can be calculated as

$$CSLDP = SLDP(Pos/Neg) + CLDP_{threshold} \quad (12)$$

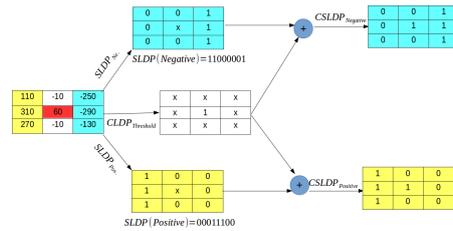


Fig. 5: A computing example for Centered Signed Local Directional Pattern

## 3 Experimental Analysis

The main steps used for texture classification are:

1. *Feature operator selection:* This step consists of picking a features method (LDP, CLDP and SLDP) and determining the descriptions for each pixel, creating a suitable scale for the textual description of the image.
2. *Local feature extraction:* The resulting description of the image is formed through a concatenation of sub-regional histogram of local pattern.
3. *Classification:* Match the unknown image description (test set) with all known images (training set) using different classifiers in different conditions.

Python-Fortran, OpenCV and scikit-learn frameworks [15] are used for experiments.

### 3.1 Classification Evaluations

The effectiveness of these methods were evaluated using different classification measures, learning curve, accuracy, precision, recall, F-score and Cohen’s kappa. The learning curve is a very useful algorithm that evaluates the sanity of an algorithm. It plots the relation between the training set size and the performance. In a basic manner it shows the starting point where the classifier begins to learn. Accuracy is the number of samples classified correctly, for example if the classifier accuracy is 50% it means that the classifier manages to classify correctly 50% of the dataset

$$Accuracy = \# \text{ of samples correctly classified} / \# \text{ of samples} \quad (13)$$

Precision is the ratio of positive predictions to all positive classes value predicted.

$$precision = \text{True positive} / (\text{True positive} + \text{False positive}) \quad (14)$$

Sensitivity is the ratio of positive predictions to all positive classes in test data.

$$recall = \text{True positive} / (\text{True positive} + \text{False negative}) \quad (15)$$

F-score conveys the balance between the precision and the recall.

$$F - score = 2 * ((precision * recall) / (precision + recall)) \quad (16)$$

Cohen’s kappa is a very good measure that can handle both multi-class and imbalanced class problem very well. It calculates the agreement between categorical data. If the value is less than or equal 0, it indicates that the classifier is useless. Table 1 shows the interpretation of Kappa value.

Table 1: Interpretation of Kappa( $\kappa$ )

Strength of argument	Value of $\kappa$
Poor agreement	$\kappa \leq 0.20$
Fair agreement	$0.20 < \kappa \leq 0.40$
Moderate agreement	$0.40 < \kappa \leq 0.60$
Good agreement	$0.60 < \kappa \leq 0.80$
Very good agreement	$0.80 < \kappa \leq 1.00$

Each classifier is trained using different parameters as shown in Table 2.

### 3.2 Textural Application

In this section, all the presented method are evaluated for texture analysis using Kylberg dataset.

**Kylberg Datasets** Kylberg dataset [14] consists of 28 categories, each categories has 160 images. All selected images have the size of  $576 \times 576$ . In Fig. 6 a sample of each category is shown.

Table 2: Classifiers parameters

Classifiers	Parameters
SVM	Polynomial linear kernel, Configuration parameter $c = 0.025$
k-NN	$k=5$
DT	Entropy, The minimum number of split is 10
RF	The number of the trees is 10, The maximum depth of the tree is 5
Adboost	The maximum number of estimator is 50, Learning rate = 1
Gaussian NB	autoselected
Perceptron	The number of passes over the training data = 100, Constant eta = 0.1

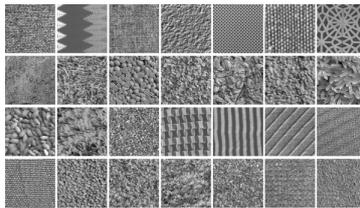


Fig. 6: The sample images of texture from Kylberg

**Experimental Results and Discussion for Kylberg Dataset** In this paragraph, we evaluate the power of the proposed descriptor for texture analysis using 5 different classifiers (K-nearest neighbor algorithm(KNN), Support Vector Machine (SVM), Perceptron, Naive-Bayes(NB), Decision Tree(DT)), under two different conditions:

1. Classification without preprocessing: In this option, raw feature vectors generated are fed into classifiers.
2. Classification with with preprocessing:
  - Standardization (Z-score normalization): Re-scale the features so that they'll have the attributes of a standard normal distribution with the mean ( $\mu$ ) equals to 0 and the standard deviation ( $\sigma$ ) equals to 1.
  - Min-Max scaling: The data is normalized to a specified range - usually 0 to 1. A Min-Max scaling is typically performed using Equation 17.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (17)$$

The Kylberg dataset is split into two: 80% of the dataset as a training set and 20% as test set, and 10 cross-validation is used. The performance of LDP, CLDP (up or down), SLDP (Positive or negative) and CSLDP(positive or negative) will be compared using 6 different classifiers (K-nearest neighbor algorithm(KNN), Support Vector Machine (SVM), Perceptron, Naive-Bayes (NB), Decision Tree (DT)) under different conditions. Regarding the length of the proposed descriptor, the basic LDP and SLDP has 56 bins and both CLDP and CSLDP have 126 bins. F-score and the performance accuracy are used to evaluate the effectiveness of our proposed methods. We begin by describing the performance of the proposed method without any preprocessing for the feature vectors, Fig. 10a

Table 3:  $F - score$  of LDP, CLDP, SLDP and CSLDP using six classifiers

Classifiers	LDP	CLDP <sub>UP</sub>	CLDP <sub>Down</sub>	SLDP(Pos.)	SLDP(Neg.)	CSLDP(Pos.)	CSLDP(Neg.)
k-NN	0.928	0.953	0.954	0.911	0.897	0.954	0.957
SVM	0.992	0.999	0.999	0.988	0.978	0.988	0.985
DT	0.897	0.924	0.916	0.85	0.848	0.916	0.87
RF	0.688	0.807	0.752	0.745	0.716	0.741	0.769
NB	0.867	0.91	0.911	0.868	0.869	0.929	0.929
Perceptron	0.756	0.888	0.891	0.559	0.568	0.83	0.808

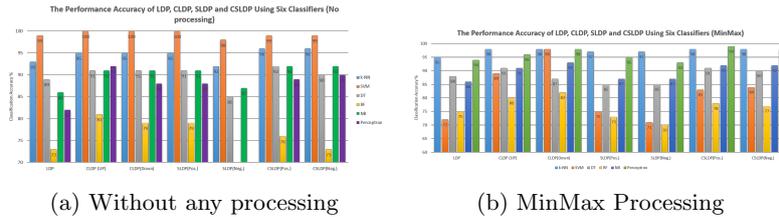


Fig. 7: The classification performance of LDP, CLDP, SLDP and CSLDP using six classifiers

shows the performance accuracy of LDP, SLDP, CLDP and CSLDP in six different classifiers. The results establish that the addition of the center pixel to the LDP in CLDP has a substantial impact on the performance, where there is an increase in the performance ranging from 1% to 9% according to each classifiers. It was noted that the trend (up or down) to calculate the edge responses vector in CLDP has no effect; as both have proximity the same accuracy. The result also shows that the sign of the gradient (positive or negative) in SLDP has improved the performance compared to LDP which use only the absolute value. For example, the accuracy for SLDP is equal to 95% and 100% compare to 93% and 99% in LDP using NB and SVM, respectively. For CSLDP, the two properties were merged, adding the center pixel value and the sign of the gradient. The  $F - score$  in Table 3 shows that the value for CSLDP is always greater than LDP except for DT classifier. It is clear that the best accuracies are achieved for the Kylberg texture dataset was 100% for both CLDP and SLDP using SVM classifier. Table 3 shows that the best performance is distributed among CLDP and CSLDP for all the classifier, however, LDP did not provide the best performance in any of the classifiers.

When features extracted are preprocessed using standarlization or MinMax before the classification, it can be noticed that there is an improvement in performance by 1% to 7% in all the classifiers except SVM which decreases by 1% (see Fig. 10a and 7b). For example, the best performance of the perceptron classifier was 92% but when we process the data it improved the performance to 99%.

### 3.3 Facial Expression Application

**Extended Cohn-Kada Dataset (CK+)** Extended Cohn-Kanade Dataset (CK+) dataset has 593 sequences from 123 persons. For each person seven facial

expression neutral, sadness, surprise, happiness, fear, anger and disgust were captured. The size of each image is  $640 \times 490$  pixel. Fig. 8 shows a sample of each expression.



Fig. 8: A sample of face expression images from Cohn-Kanade dataset

**Experimental Results and Discussion for CK+** Usually in facial applications the data size is fixed, so we calculated the learning curve to determine how much training dataset is sufficient to teach the classifiers. Note that in each of the two methods LDP and CLDP, the SVM began to be learned when the sample size was between 0 and 500 with an accuracy around 85% as shown in figs. 9a and 9b. On the other hand, we also noticed that SVM learned faster on SLDP with an accuracy around 90% utilizing the same sample size. These results demonstrate the strength and effectiveness of the methods despite the small sample size. It is found that the learning curve starts to flatten when the sample size is around 1500 with an accuracy of  $\simeq 98\%$  with all the features methods which indicate that the classifiers are gaining less knowledge.

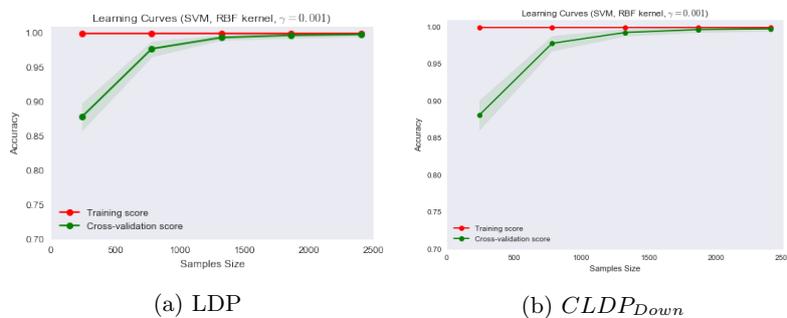


Fig. 9: Learning curve of LDP,  $CLDP_{Down}$  using SVM

In table 4 the average kappa scores when applying each classifier to 40% of dataset as a training dataset and reminder as a test dataset are shown. It was noticed that LDP performance was not the best in any of the classifiers. On the contrary, for instance, its performance with (DT) was 0.64 compare to the best performance 0.75 using CLDP. Which establishes the strength and efficiency of the presented methods. The best classifier performance was for both SVM and

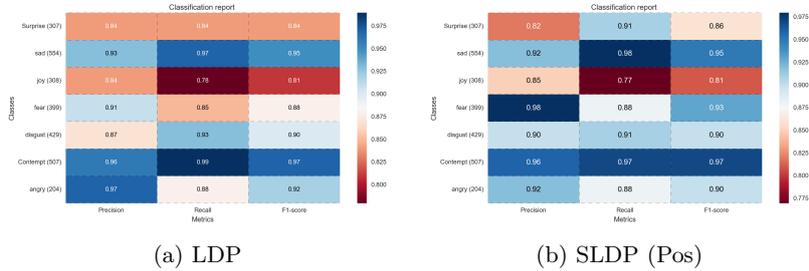


Fig. 10: Classification report of LDP and SLDP (Pos) using SVM

perceptron with accuracy ranging from 0.98 to 0.99 in all feature methods. This is why we chose SVM to plot a learning curve and the classification report as it will be seen later. For DT classifier there was a 10% improvement in the SLDP method compared to LDP, LDP accuracy was 0.64 compared to SLDP at 0.75. The weak performance of all classifiers was when NB was used to judge the methods.

Table 4: The average kappa scores of LDP, CLDP and SLDP

Features	DT	SVM	NB	k-NN	Perceptron
<b>LDP</b>	0.64	0.98	0.61	0.82	0.98
<b>CLDP (UP)</b>	0.62	0.99	0.58	0.84	0.99
<b>CLDP (Down)</b>	0.62	0.99	0.62	0.83	0.99
<b>SLDP (Pos)</b>	0.75	0.99	0.58	0.83	0.99
<b>SLDP (Neg)</b>	0.73	0.99	0.61	0.83	0.99

In Fig.10 a classification report is plotted for each facial expression emotions class, where the x-axis shows the performance for precision, recall and F-score and y-axis shows the facial expression classes and their images number. Once again, our features methods achieved the best score on almost every facial expression class. The F-score of our SLDP outperforms the original LDP in every facial expression class.

## 4 Conclusion

Several extensions and modifications of Local Directional Pattern (LDP) have been proposed, CLDP, which takes into account the center value, SLDP, which considers the two different trends of the gradient and CSLDP, which includes the center for both directions (positive or negative). LDP, CLDP, SLDP and CSLDP were tested using Kylberg dataset of 3200 images using six different classifiers in different conditions. The performance of the proposed operators was also investigated on facial expression analysis. Results show that CLDP and CSLDP outperform the existing LDP. This shows that the center pixel and

the directions of the gradient are very important in the extraction of textural features.

## References

1. Eleyan, A. and Demirel, H., 2011. Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering & Computer Sciences*, 19(1), pp.97-107.
2. Galloway, M.M., 1975. Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2), pp.172-179.
3. Bovik, A.C., Gopal, N., Emmoth, T. and Restrepo, A., 1992. Localized measurement of emergent image frequencies by Gabor wavelets. *IEEE Transactions on Information Theory*, 38(2), pp.691-712.
4. Ojala, T., Pietikinen, M. and Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1), pp.51-59.
5. Jabid, T., Kabir, M.H. and Chae, O., 2010, January. Local directional pattern (LDP) for face recognition. In *Proceedings of the IEEE International Conference on Consumer Electronics(ICCE)*. pp. 329-330.
6. Wang, X., Gong, H., Zhang, H., Li, B. and Zhuang, Z., 2006, August. Palmprint identification using boosting local binary pattern. In *Proceedings of 18th International Conference on Pattern Recognition (ICPR'06)*. 3, pp. 503-506.
7. Ahonen, T., Hadid, A. and Pietikainen, M., 2006. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12), pp.2037-2041.
8. Zhang, G., Huang, X., Li, S.Z., Wang, Y. and Wu, X., 2004. Boosting local binary pattern (LBP)-based face recognition. In *Advances in biometric person authentication*. pp. 179-186. Springer Berlin Heidelberg.
9. Ojala, T., Pietikainen, M. and 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), pp.971-987.
10. Pietikinen, M., Hadid, A., Zhao, G. and Ahonen, T., 2011. *Local binary patterns for still images* (pp. 13-47). Springer London.
11. Jabid, T., Kabir, M.H. and Chae, O., 2010. Robust facial expression recognition based on local directional pattern. *ETRI journal*, 32(5), pp.784-794.
12. Kabir, M.H., Jabid, T. and Chae, O., 2010, August. A local directional pattern variance (LDPv) based face descriptor for human facial expression recognition. In *Proceedings of Advanced Video and Signal Based Surveillance (AVSS), Seventh IEEE International Conference*. pp. 526-532.
13. Zhong, F. and Zhang, J., 2013. Face recognition with enhanced local directional patterns. *Neurocomputing*, 119, pp.375-384.
14. Kylberg, G., 2011. *Kylberg Texture Dataset v. 1.0*. Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University.
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(Oct), pp.2825-2830.



## **Chapter 6**

# **Angled Local Directional Pattern for Texture Analysis with an Application to Facial Expressions**

# Angled Local Directional Pattern for Texture Analysis with an Application to Facial Expressions

Abuobayda M Shabat<sup>1</sup> and Jules-Raymond Tapamo<sup>1,\*</sup>

<sup>1</sup>School of Engineering, Howard College Campus, University of KwaZulu-Natal, Durban, 4041, South Africa

\*tapamoj@ukzn.ac.za

**Abstract:** Local Binary Pattern (LBP) is currently one of the most common feature extraction methods used for texture analysis. However, LBP suffers from random noise, because it depends on image intensity. A recently introduced more stable feature method, Local Directional Pattern (LDP), uses the gradient space instead of the pixel intensity. Typically, LDP generates a code based on the edge response values for eight directions around pixels. Yet, despite the great achievement of LDP, it has two drawbacks. The first is the chosen number of significant bits  $k$ , which is always three in the experiments conducted to date. Secondly, the original LDP method is structured around the responses of the application of the Kirsch Masks on the 8-neighbourhood of each pixel of the image and ignores the centre pixel value, despite it being very important in many applications. This paper presents Angled Local Directional Pattern (ALDP), which is an improved version of Local Directional Pattern (LDP), for texture analysis. Experimental results on two different texture dataset (Kylberg and KTHTIPS2b), using six different classifiers [K-nearest neighbour algorithm (k-NN), Support Vector Machine (SVM), Random Forest (RF), Perceptron, Naive-Bayes(NB), and Decision Tree (DT)], show that ALDP substantially outperforms both LDP and LBP methods. The ALDP has been evaluated with the Cohn-Kanade database to recognize the facial expressions emotion. Results indicate a very high recognition rate for the proposed method. An added advantage is that ALDP needs a selection of the number significant bits ( $k$ ) as opposed to LDP.

## 1. Introduction

Texture can be described as complex visual form composed of units or sub-patterns that have a wide variety of characteristics such as luminosity, color, size, slope, shape. Texture is determined as the spatial variation of pixel intensities. It can help to segment an image into different homogeneous regions where the criteria of similarity of a region are based on texture features. Texture analysis is the operation of Machine Learning to characterize texture in an image [1]. Texture analysis is used in a variety of applications, including medical imaging, remote sensing and security. Several feature methods have been proposed in literature, including Local Binary Pattern (LBP) [7], Local Directional Pattern (LDP)[3], Local Directional Pattern variance (LDPv) [8], Directional Local Binary Pattern (DLBP) [9] and many others.

Local Binary Pattern has gained a lot of interest as a result of its simplicity and excellent performance in texture analysis. A study by Song et al. [11] employed LBP operator to calculate texture for multi-spectral remote sensing. The outcome showed more than 4% increase in performance, compared to Grey Level Co-occurrence Matrix (GLCM) method. The successful application of

LBP in a variety of domains has inspired many several modifications [12, 13, 14, 15, 16] to improve its success rate. Guo [26] proposed a completed LBP (CLBP) scheme which improves the discrimination capability of LBP descriptors. It presents two operators, the CLBP-Sign, which is equivalent to the conventional LBP, and CLBP-Magnitude (CLBPM), which measures the local variance of magnitude. This study demonstrated that the sign component is more significant than the magnitude component in preserving the local information difference. Zhao [27] presented a local binary code (LBC) which is based on extracting the grayscale different scale instead of the structural information as, on the CLBP. The results show that LBC is more robust against the modification of the illumination condition or the alteration of the texture scale. However, LBP suffers from random noise because it depends on neighbouring pixels intensity. A more stable feature method, the Local Directional Pattern (LDP), was introduced by Jabid et al. [3], which considers the edge response of the application of Kirsch gradient operator rather than the raw pixel intensities as is done with LBP. LDP has been used in several applications, such as textural classification [17], signature verification [4], and facial expression recognition [3]. Despite the outstanding achievement of LDP in many computer vision applications, its underlying working mechanism still requires more investigation. Kabir et al. [8] presents Local Directional Pattern variance (LDPv), which extracts the texture and contrast information to characterize the facial components. Principal Component Analysis (PCA) was utilized in LDPv to reduce the dimensionality of the most significant element on LDP. Results show that the LDPv method achieved a higher recognition rate compared to LBP and LDP. Zhong and Zhang [10] present Enhanced Local Directional Pattern (ELDP) that employs the directions of the two most significant edge response values. Results show that ELDP is more robust to non-monotonic illumination changes compared to LDP. Shabat and Tapamo [9] present Directional Local Binary Pattern (DLBP), which uses the centre value as a threshold for the eight directional edge response values. Directional Local Binary Pattern is applied on texture classification and results show that it outperforms both LDP and LBP. Angled Local Directional Pattern (ALDP) is proposed in this paper to resolve two problems encountered when using LDP. The first problem is the value of the number of significant bits  $k$ . The choice of  $k = 3$ , as established in the literature, is empirical. A careful investigation revealed that the change in the value of  $k$  affects the performance of the LDP. Secondly, the original LDP codes are generated based on values of the edge responses when Kirsch mask is applied in the eight directions around a pixel, but this method ignores the centre pixel value, which leads to lost information.

## 2. Features methods for texture analysis

An important process in texture analysis is feature extraction. In this section, Local Binary Pattern (LBP), Local Directional Pattern (LDP) together with the proposed Angled Local Directional Pattern (ALDP) are presented.

### 2.1. Local Binary Pattern

A detailed study of Local binary pattern (LBP) is presented by Ojala et al. [2]. Local Binary Pattern has since been used in several computer vision applications. It relabels each pixel of an image by constructing an 8-bit binary string corresponding to the 8-neighbourhood. This binary string is obtained through the following process: the gray value  $g_p$  of each pixel,  $p$ , is compared to the gray value of all its 8-neighbours, where the 8-neighbours are represented as shown in Fig 1,

$n_i(p)$  represents the  $i^{th}$  neighbour of pixel  $p$ . After the comparison, if  $g_p \leq g_{n_i(p)}$ , a 1 is produced, otherwise a 0 is returned. The binary code obtained is then converted into a decimal number. In other words, for each pixel,  $p$ , its LBP code,  $LBP(p)$ , is computed as

$$LBP(p) = \sum_{i=0}^7 S(g_{n_i(p)} - g_p)2^i \quad (1)$$

where

$$S(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$n_3(p)$	$n_2(p)$	$n_1(p)$
$n_4(p)$	$p$	$n_0(p)$
$n_5(p)$	$n_6(p)$	$n_7(p)$

Fig. 1: 8-neighbors of a pixel  $p$

## 2.2. Local Directional Pattern

Local Directional Pattern code is calculated by first applying Kirsch Masks ( $M_0, \dots, M_7$ ); these masks are shown in Fig.2. Given an image,  $I$ , an 8-dimensional vector ( $m_0, \dots, m_7$ ) can be calculated for the eight directions as follows:

$$m_i = \sum_{l=-1}^1 \sum_{k=-1}^1 I(x+l, y+k) \times M_i(l, k) \quad (3)$$

Local Directional Pattern code is then generated using the  $k$  most significant responses. Hence, the bits corresponding to the top  $k$  Kirsch masks application responses are set to 1, and the remaining  $(8 - k)$  bits are set to 0. For a pixel at the position  $(x, y)$ , the LDP code  $LDP_{x,y}$  is derived using Equation 4.

$$LDP_{x,y}(m_0, m_1, \dots, m_7) = \sum_{i=0}^7 S(m_i - m_k) \times 2^i \quad (4)$$

where  $m_k$  is the most prominent response and  $S(x)$  defined as in 1.

Fig.3 gives an example of the calculation the LDP code.

The histogram  $H$  is employed on the transformed image (LDP) to encode the image as a feature vector. The histogram obtained from the transformation can be defined as

$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$
East $M_0$	North East $M_1$	North $M_2$	North West $M_3$
$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$
West $M_4$	South West $M_5$	South $M_6$	South East $M_7$

Fig. 2: The Kirsch Mask

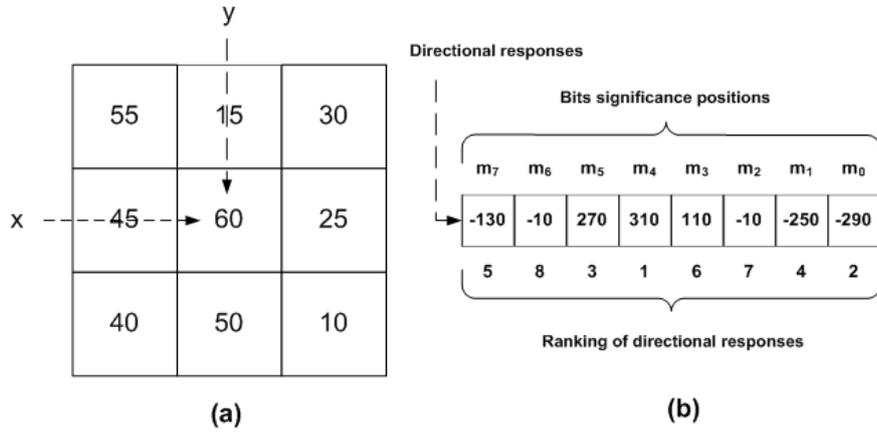


Fig. 3: Kirsch Mask application responses on a pixel  $(x, y)$

$$H_i = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} p(LDP_{x,y}, C_i) \quad (5)$$

where  $C_i$  is the  $i^{th}$  ranked LDP value,  $i = 1, \dots, 8$ ,  $C_k$  and  $p$  is defined as

$$p(x, a) = \begin{cases} 1 & \text{if } x = a \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

### 2.3. Angled Local Directional Pattern

There are two problems associated with an LDP operator: first, the number of significant bits,  $k$ , has to be determined. The classic LDP empirically chooses  $k = 3$ . It has been established that the change in the value of the  $k$  affects performance. Secondly, classic LDP codes are generated based on the value of edge responses in eight directions around each pixel, based on the 8-neighborhood only, and it ignores the center pixel, although the centre pixel is very important in many applications. In Angled Local Directional Pattern (ALDP), instead of choosing  $k$  values of the  $3 \times 3$  window, the centre value for each angle will be taken as a threshold. Angled Local

Directional Pattern features are made up of an 8-bit binary code produced for each pixel. Angled Local Directional Pattern features are calculated in three steps:

1. Generate the Kirsch Masks application responses  $(m_0, \dots, m_6, m_7)$  as in the classic LDP. Fig. 3 gives an example.
2. Calculate the angular vector values, where each angle  $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$  has a different number of vectors, and each vector contains three values. The central value in each vector will be chosen as a threshold for the other two neighbouring pixels. A binary code of 1 is generated if the threshold is greater than both neighbours; otherwise 0 binary code is generated (see Fig. 4). For example,  $p_0$  is one of the vectors for the  $0^\circ$ , which contains three values,  $m_4$ ,  $c$  and  $m_0$ . Since  $c$  is the central value,  $c$  is the threshold for  $m_4$  and  $m_0$ . If the value of  $c$  is greater than  $m_4$  and  $m_0$ , 1 binary code is generated; otherwise, 0 binary code is generated. Equations 7 to 14 below show all the vectors values for all the angles:

For  $0^\circ$ :

$$p_0 = b(m_0 - c, m_4 - c) \quad (7)$$

$$p_1 = b(m_3 - m_2, m_1 - m_2) \quad (8)$$

$$p_2 = b(m_7 - m_6, m_5 - m_6) \quad (9)$$

For  $90^\circ$ :

$$p_3 = b(m_7 - m_0, m_1 - m_0) \quad (10)$$

$$p_4 = b(m_6 - c, m_2 - c) \quad (11)$$

$$p_5 = b(m_3 - m_4, m_5 - m_4) \quad (12)$$

For  $45^\circ$ :

$$p_6 = b(m_5 - c, m_1 - c) \quad (13)$$

For  $135^\circ$ :

$$p_7 = b(m_3 - c, m_7 - c) \quad (14)$$

where

$$b(r, s) = \begin{cases} 1 & \text{if } r \geq 0 \text{ and } s \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

Finally, the ALDP code  $ALDP_{x,y}(p_0, p_1, \dots, p_7)$  of the pixel at the position  $(x, y)$  with the angle vector values,  $(p_0, p_1, \dots, p_7)$  is derived using equation 16, as

$$ALDP_{x,y}(p_0, p_1, \dots, p_7) = \sum_{i=0}^7 p_i \times 2^i \quad (16)$$

3. Compute ALDP histogram; this step remains the same as the third step on LDP. Fig. 5 shows an example of computing ALDP.

### 3. Experimental results and discussion

Performance of ALDP was evaluated by testing its ability to classify textures and its suitability to recognize facial expression.

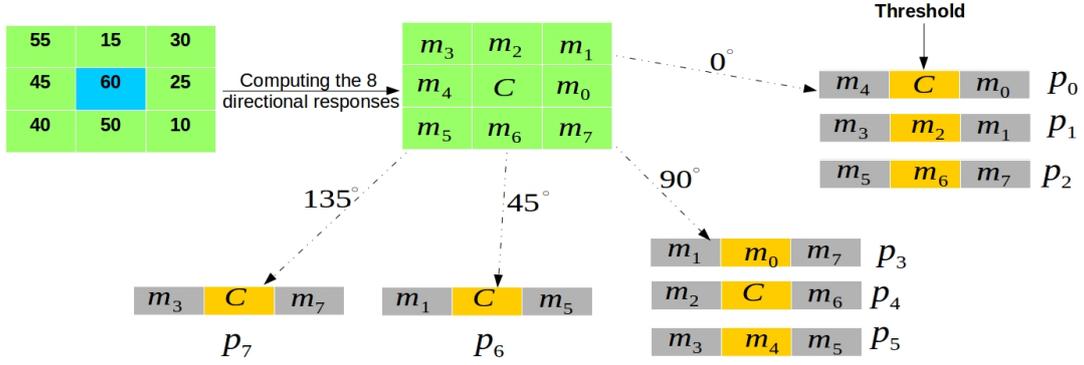


Fig. 4: ALDP code generation process. This consists of computing the eight directional response values using Kirsch Masks, and extracting the vector for each angle ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). In each vector, the center value is used as threshold for two neighbours; 1 is generated if the threshold is greater than both neighbours, and a 0 is generated otherwise.

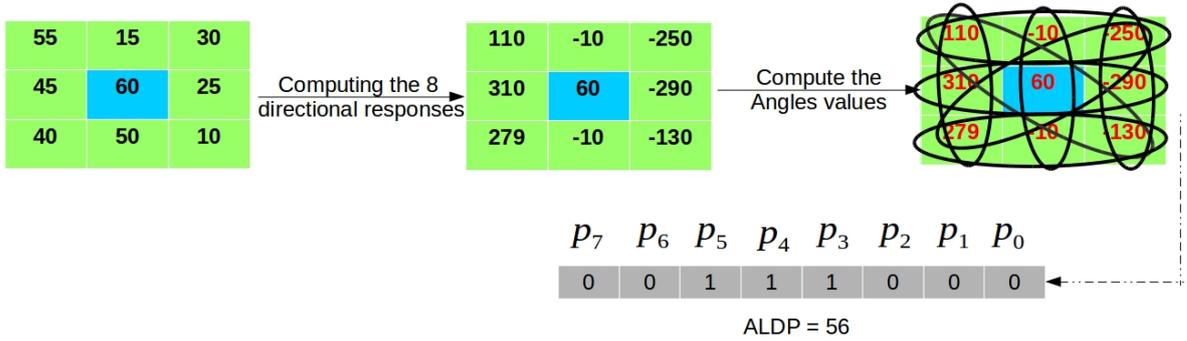


Fig. 5: Computing of ALDP code of a pixel

### 3.1. Experiment 1: Texture Analysis

**3.1.1. Data set:** In this experiment, two texture image datasets are used. They were chosen because of the variety of their characteristics in terms of the number of classes and the number of samples. The Kylberg dataset has 28 categories of 160 images each, with grayscale images of different man-made and natural textured surfaces. Categories are very similar with regard to scale and illumination. All selected images have a size of  $576 \times 576$ . In Fig. 6(a), some sample images are shown.

KTH-TIPS2b dataset has 11 categories of 432 images each. Each category has images with variations in scale, illumination and pose. Each selected image has  $200 \times 200$  pixels. In Fig. 6(b), some

image samples of KTH-TIPS2b dataset are shown. Classifier parameters are given in Table 1. In this article, the classification model used has the usual two components: Feature Extraction and Feature Classification. Datasets used are split into two sets, 80% as a training dataset and the remaining as a test dataset. Each classifier is trained using different parameters, as shown in Table 1.

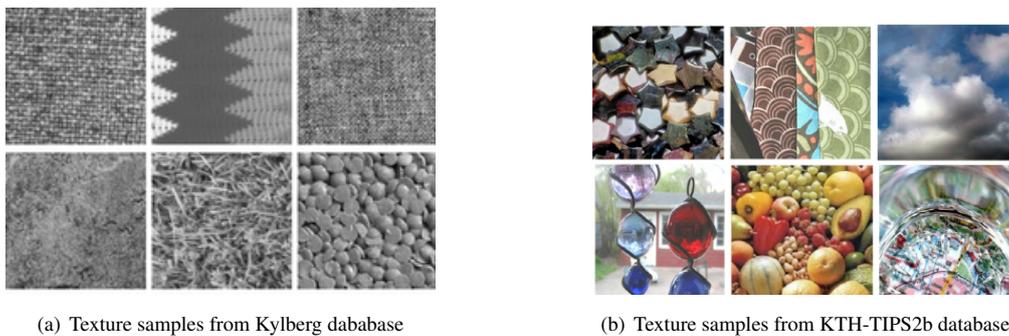
**Table 1** Classifier parameters

Classifiers	Parameters
SVM	Polynomial linear kernel Configuration parameter $c = 0.025$
k-NN	$k = 5$
DT	Entropy The minimum number of split is 10
RF	The number of the trees is 10 The maximum depth of the tree is 5
Gaussian NB	autoselected
Perceptron	The number of passes over the training data = 100 Constant $\eta = 0.1$

Accuracy is used to evaluate the classifiers, which compute the number of samples classified correctly, as shown in Eq.17.

$$\text{Accuracy} = \frac{\text{No. of samples correctly classified}}{\text{No. of samples}} \quad (17)$$

The proposed features extraction method is implemented using the python-fortran framework. The Scikit-learn toolkit [6] is used for the classification.



**Fig. 6:** Textures Data sets

**3.1.2. Performance of ALDP on Texture Analysis:** In this section, the performance of LDP and ALDP are compared, using six different classifiers under different conditions: K-nearest neighbor algorithm (k-NN); Support Vector Machine (SVM); Perceptron; Decision Tree (DT); Random Forest (RF); Naive-Bayes (NB) and Perceptron.

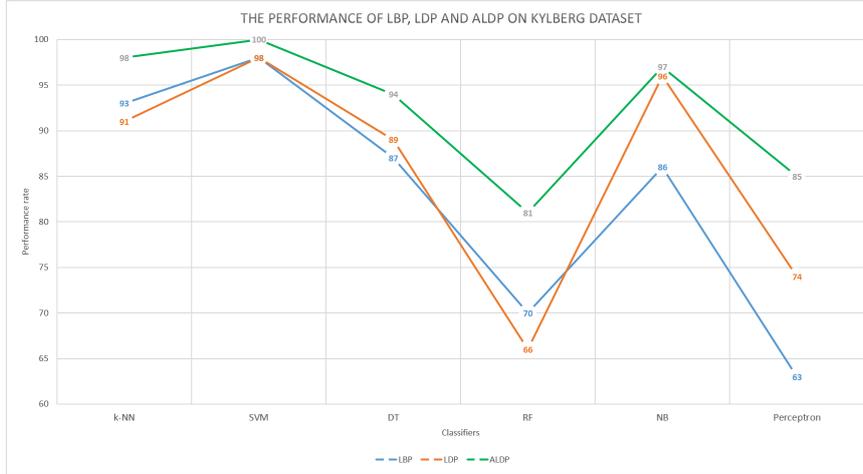


Fig. 7: The performances of LDP and ALDP using Kylberg dataset and six classifiers

### Kylberg Results

Table 2 shows the performance of LBP, LDP and ALDP using six different classifiers. With the Kylberg dataset ALDP achieves the best accuracy (100%), using the SVM classifier. Angled Local Direction Pattern improvement of the performance ranges from 1% to 15%, in all classifiers as shown in Table 2. For example, ALDP performance using k-NN is 98%, while LDP performance is 91%. This shows a performance improvement of 7%. It is also noticeable that LBP and LDP are consistently outperformed by ALDP (see Fig. 7).

**Table 2** The performance accuracy of LBP, LDP and ALDP applied on Kylberg dataset using six classifiers

Kylberg						
	k-NN	SVM	DT	RF	NB	Perceptron
LBP	0.93	0.98	0.87	0.70	0.86	0.63
LDP	0.91	0.98	0.89	0.66	0.96	0.74
ALDP	0.98	1.0	0.94	0.81	0.97	0.85

### KTH-TIPS2-b Results

Table 3 shows the performances of LBP, LDP and ALDP with six different classifiers. It is clear that the best accuracy (94%) was achieved by ALDP using the SVM classifier, on KTH-TIPS2-b texture dataset. The improvement of performance with ALDP ranges from 5% to 14%, and it is consistent through classifiers. For example, the ALDP success rate using k-NN is 88%, while the LDP success rate is 80%, which is an improvement of 8%. It is also important to note that both LDP and LBP performed poorer for all classifiers (see Fig. 8).

**Table 3** The performance accuracy of LBP, LDP and ALDP applied on KTH-TIPS2-b dataset using six classifiers

KTH-TIPS2-b						
	k-NN	SVM	DT	RF	NB	Perceptron
LBP	0.80	0.79	0.68	0.53	0.50	0.43
LDP	0.80	0.88	0.75	0.56	0.56	0.54
ALDP	0.88	0.94	0.81	0.63	0.65	0.69

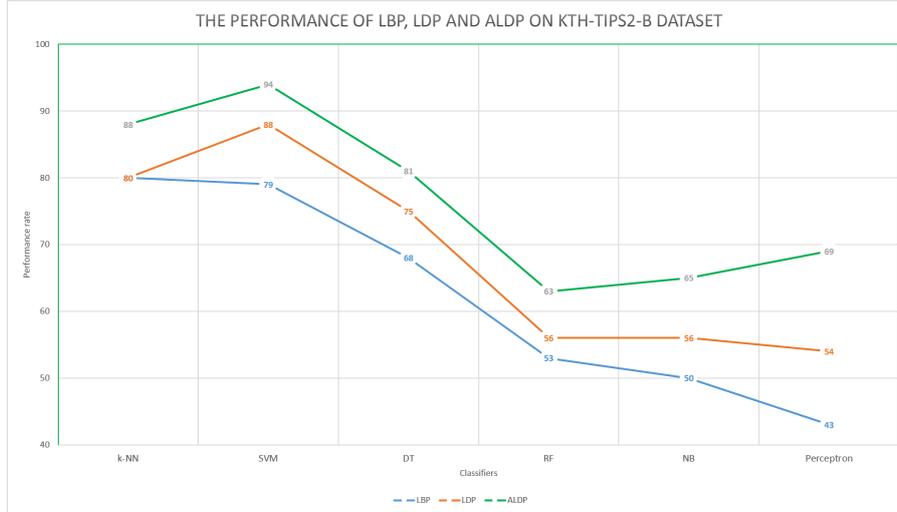


Fig. 8: The performance accuracy of LDP and ALDP applied on KTH-TIPS2-b dataset using six classifiers

### 3.2. Experiment 2: Identifying Facial Emotional Expression

After establishing the power of the ALDP in classifying the texture in the previous experiment and solving the problem of selecting the most edge without reducing the performance of the LDP but on the contrary improving the performance of the LDP, we implemented a practical application to examine the strength of ALDP in real life application. In fact, we evaluated how well ALDP characterizes facial emotion (i.e. joy, sadness, anger, disgust, fear, contempt and surprise) based on facial image. In this paper, we replicated and expanded the experiment done by Kabir [8] and compared the performance of the ALDP with LDP, LDPv and LBP methods.

**3.2.1. Data set:** In this work, we attempted to identify facial emotional expressions using the well-known extended Cohn-Kanade data set (CK+) [19]. This data set contains seven facial expressions (joy, sadness, anger, disgust, fear, surprise and contempt) taken from 210 adults aged between 18 to 50 years: 69% were female, 13% Afro-American, 81% Euro-American and 6% other groups. Fig.9 shows a sample of face expression images from the CK+ data set. This data set contains 593 image sequences of 123 subjects. Each sequence has consecutive images, beginning with a neutral facial expression (first image) right up to the image of a visible emotional expression on the face (last image). In this experiment, 400 images were selected for each emotional expression (joy, sadness, anger, disgust, fear and surprise).

**3.2.2. Performance of ALDP on Facial Expression Identification:** In this experiment facial image was detected using Haar-like features [20] and then cropped from the original image and normalized to a size of  $200 \times 200$ . Fig.10 shows an example of detection and cropping of facial regions, using Haar-like features.

No preprocessing was performed because LDP and ALDP are not affected by the noise and illumination changes. We chose to divide each facial image  $200 \times 200$  into  $10 \times 10$  blocks, and the size of each block is  $20 \times 20$  pixels. In this experiment, two modes were employed to split the data. In the first mode, 10-cross validation [21] was used to split the training set into 10 smaller sets. In the



Fig. 9: The sample facial expression images from the Cohn-Kanade data set

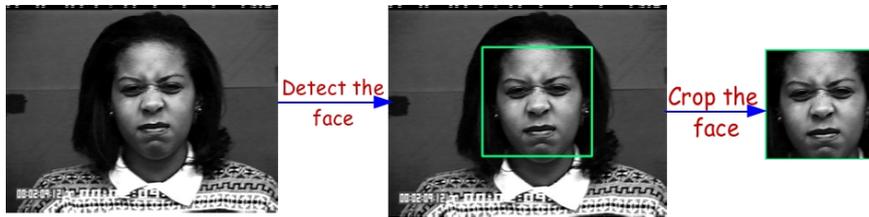


Fig. 10: Detecting and cropping the facial region from the original image using Haar-like features

second mode, different values were employed as a training data set varying from 10% to 90% of the data set and the rest as a test data set and performances were computed for each value.

Table 4 shows the results of the comparison between the LDP and the ALDP using 10 cross validation. The best recognition rates were achieved by ALDP. Note that the performance of ALDP using both classifiers SVM (linear and polynomial) and Perceptron is almost equal to the performance of the LDP and the difference not more than 1% in favour of ALDP. There is an increase in the recognition rate for ALDP from 4% to 11% compared to LDP in both DT and NB.

**Table 4** 7-class expression recognition using four different classifiers

Features method	SVM (Linear)	SVM (Polynomial)	DT (Entropy)	Perceptron	NB
<b>LDP</b>	0.98	0.98	0.78	0.98	0.75
<b>ALDP</b>	0.99	0.99	0.82	0.99	0.86

Figs.11 and 12 show the recognition rates for different testing set cases. For instance, in Fig. 11, when the testing data set was 30% of the whole data set, the performance for ALDP using DT was 86%. Note that even with a small training data set, ALDP still works well. For instance, in Fig.12, when 80% of the data set was used as a testing data set and the rest (20%) as a training data set, ALDP achieved a very high recognition rate with 97%. This demonstrates the strength of the proposed method.

Furthermore, Table 5 exhibits the effectiveness of the ALDP, showing how it compares to previous studies. Although this study includes a new facial expression (contempt) which was not employed in previous studies, and a larger number of subjects and without any kind of preprocessing, ALDP achieves the best performance with 99%. The improvement in performance is due to several fac-

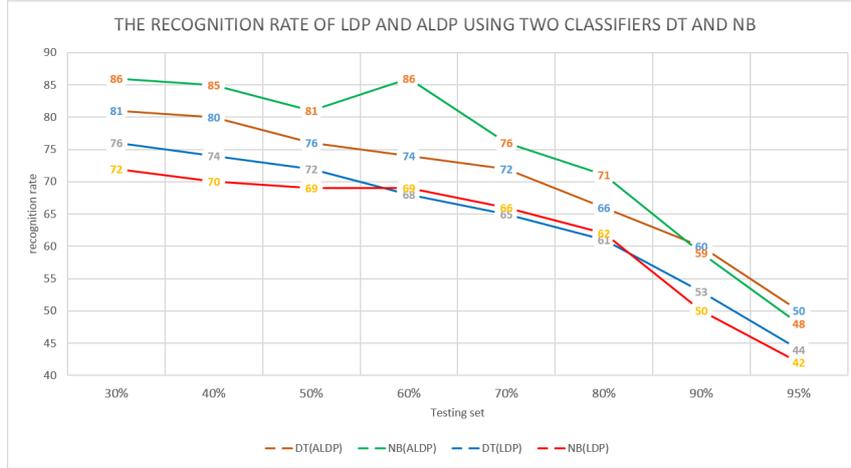


Fig. 11: The performance of LDP and ALDP using the two classifiers DT and NB

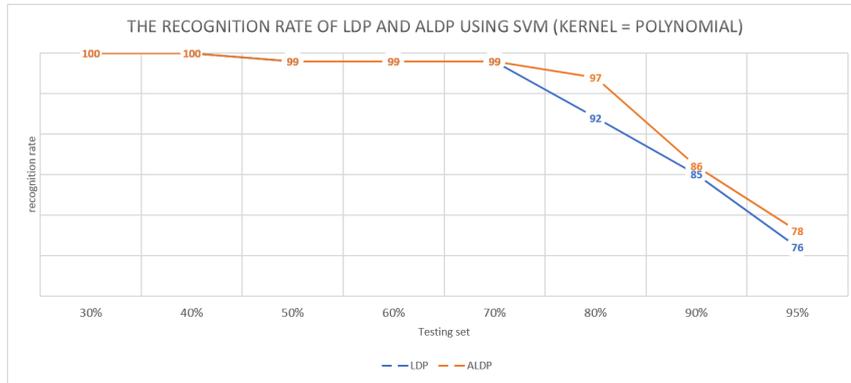


Fig. 12: The performance of the LDP and ALDP using the SVM classifier

**Table 5** The performance rate of previous studies and ALDP applied on the Cohn-Kanade data set

Features method	Performance
ICA [22]	0.60
EICA [22]	0.66
Zenike Moment (10 order) [23]	0.73
log-Gabor feature [24]	0.92
LDP [4]	0.93
VLBP [25]	0.96
LDPv [8]	0.97
ALDP	0.99

tors. The first factor takes into account the value of the central pixel, which may carry important information, rather than ignoring its value and applying it only as a threshold. The central pixel has the same weight as the neighbouring pixels. Another factor influencing performance is the selection of the number of significant bits which is based on the pixel relation to one another and is not the same as in the LDP where the number of significant bits is adaptively chosen.

## 4. Conclusion

Angled Local Directional Pattern (ALDP) is proposed in this study, which resolves two problems of LDP. It takes into account the centre value and does not have to select the number of significant bits  $k$ . Both LDP and ALDP were tested on two texture data sets (Kylberg and KTH-TIPS2-b) using six different classifiers [K-nearest neighbour algorithm (KNN), Support Vector Machine (SVM), Perceptron, Naive-Bayes (NB), Adaboost, Decision Tree (DT)] under different conditions. An improvement in the performance ranging from 1% to 15% was observed, depending on the chosen classifiers. Results showed that ALDP outperforms the classic LDP. This proves the fact that for each kernel both the centre pixel and neighbours in all directions are very important factors in texture analysis. The ALDP has also been applied to the Cohn-Kanade database with the objective of recognising facial expressions. Results obtained showed that ALDP outperforms several earlier studies' methods. Although this study has a new facial expression (contempt) which was not employed in the previous studies, a larger number of subjects and there was no preprocessing.

## 5. References

- [1] Burrascano, P., Fiori, S. and Mongiardo, M.: A review of artificial neural networks applications in microwave computeraided design (invited article). *International Journal of RF and Microwave ComputerAided Engineering*, 1999, **9**, (3), pp. 158-174.
- [2] Ojala, T., Pietikinen, M. and Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern recognition* 1996, **29**, (1), pp. 51-59.
- [3] Jabid, T., Kabir, M.H. and Chae, O.: Local directional pattern (LDP) for face recognition. *Proc. Int. Conf. IEEE on Consumer Electronics*, January 2010, pp. 329-330.
- [4] Jabid, T., Kabir, M.H. and Chae, O.: Robust facial expression recognition based on local directional pattern. *ETRI journal*, 2010, **32**, (5), pp. 784-794.
- [5] Kylberg, G., *Kylberg Texture Data set 2011 1.0*.
- [6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J.: *Scikit-learn: Machine learning in Python*. *Journal of Machine Learning Research*, 2011, **12**, (10), pp. 2825-2830.
- [7] Wang, L. and He, D.C.: Texture classification using texture spectrum. *Pattern Recognition*, 1990, **23**, (8), pp. 905-910.
- [8] Hasanul Kabir, T.J. and Chae, O.: Local Directional Pattern Variance (LDPv): A robust feature descriptor for facial expression recognition. *The International Arab Journal of Information Technology*, 2010, **9**, (4), pp. 382-391.
- [9] Shabat, A.M. and Tapamo, J.R.: Directional Local Binary Pattern for Texture Analysis. *Proc. Int. Conf. Springer in Image Analysis and Recognition*, July 2016, pp. 226-233.
- [10] Zhong, F. and Zhang, J.: Face recognition with enhanced local directional patterns. *Neuro-computing*, 2013, **119**, pp. 375-384.

- [11] Song, C., Yang, F. and Li, P.: Rotation invariant texture measured by local binary pattern for remote sensing image classification. Proc. Int. Conf. second International Workshop on Education Technology and Computer Science (ETCS), March 2010, pp. 3-6.
- [12] Ahonen, T., Hadid, A. and Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence, 2006, **28**, (12), pp. 2037-2041.
- [13] Zhang, G., Huang, X., Li, S.Z., Wang, Y. and Wu, X.: Boosting local binary pattern (LBP)-based face recognition. In Advances in biometric person authentication Springer Berlin Heidelberg, 2004, pp. 179-186.
- [14] Rodriguez, Y. and Marcel, S.: Face authentication using adapted local binary pattern histograms. Proc. Int. Conf. European Conference on Computer Vision, May 2006, pp. 321-332. Springer Berlin Heidelberg.
- [15] Ojala, T., Pietikainen, M. and Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence, 2002, **24**, (7), pp. 971-987.
- [16] Pietikinen, M., Hadid, A., Zhao, G. and Ahonen, T.: Computer vision using local binary patterns. Springer Science & Business Media, 2011, **40**.
- [17] Shabat, A.M. and Tapamo, J.R.: A comparative study of Local directional pattern for texture classification. Proc. Int. Conf. IEEE World Symposium on Computer Applications & Research (WSCAR), January 2014, pp. 1-7.
- [18] Mallikarjuna, P., Fritz, M., Targhi, A. T., Hayman, E., Caputo, B., Eklundh, J. O.: The kth-tips and kth-tips2 databases. **2006**
- [19] Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z. and Matthews, I.: The extended cohn-kanade data set (ck+): A complete data set for action unit and emotion-specified expression. Proc. Int. Conf. IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, June 2010, pp. 94-101.
- [20] Viola, P. and Jones, M.: Rapid object detection using a boosted cascade of simple features. Proc. Int. Conf. Computer Vision and Pattern Recognition on IEEE Computer Society, 2001. **1**, pp. I-511.
- [21] Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In ijcai, 1995, **14**, (2), pp. 1137-1145.
- [22] Uddin, M.Z., Lee, J.J. and Kim, T.S.: An enhanced independent component-based human facial expression recognition from video. IEEE Transactions Consumer Electronics, 2009, **55**, (4), pp. 2216-2224.
- [23] Lajevardi, S.M. and Hussain, Z.M.: Higher order orthogonal moments for invariant facial expression recognition. Digital Signal Processing, 2010, **20**, (6), pp. 1771-1779.
- [24] Lajevardi, S.M. and Hussain, Z.M.: Feature extraction for facial expression recognition based on hybrid face regions. Advances in Electrical and Computer Engineering, 2009, **9**, (3), pp. 63-67.

- [25] Zhao, G. and Pietikainen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions pattern analysis and machine intelligence*, 2007, **29**, (6), pp.915-928.
- [26] Guo, Z., Zhang, L. and Zhang, D., 2010. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6), pp.1657-1663.
- [27] Zhao, Y., Huang, D.S. and Jia, W., 2012. Completed local binary count for rotation invariant texture classification. *IEEE transactions on image processing*, 21(10), pp.4492-4497.



## **Chapter 7**

# **Circular Local Directional Pattern for Texture Analysis**

# Circular Local Directional Pattern for Texture Analysis

Abuobayda M. Shabat · Jules-Raymond Tapamo

Received: date / Accepted: date

**Abstract** This paper presents a novel texture feature extraction method, Circular Local Directional Pattern (CILDP), that is inspired by Local Binary pattern(LBP) and Local Directional Pattern(LDP). This method relies on circular shape to compute the directional edge responses based on Kirsch Masks using different radiuses. The performance of the proposed method is evaluated using five classifiers on textures from the Kylberg dataset. Results achieved establish that the proposed method consistently outperforms LBP and LDP when different radiuses are considered.

**Keywords** Texture Analysis · Local Binary Patterns · Local Directional Pattern · Classification

## 1 Introduction

Texture analysis is an important aspect employed in many image analysis and computer vision applications such as object classification and face identification. Although, texture analysis is extremely used in computer vision and the many efforts to define it in universal terms, texture analysis loss a precise definition. One of the better description of texture analysis, defining it as a function of the spatial variation in pixel intensities [1]. Many of the features methods have been

proposed, such as grey-level co-occurrence matrix (GLCM) [2], local binary pattern [3], scale-invariant feature transform (SIFT) [4], speeded up robust features (SURF) [5], local directional pattern (LDP) [6], directional local binary pattern (DLBP) [7] and so much more. Some of the methods have been applied for surface characterizations and texture analysis applications[8,9,10]. Among all the features methods, local binary pattern is the most popular feature. It has been employed in several applications, including facial recognition [11], texture analysis [9] and remote sensing [12]. Instigated by the power and the simplicity of LBP, many researchers proposed different improvement on LBP, such as dominant local binary pattern (DLBP) [13], completed local binary pattern (CLBP)[14] and center-symmetric local binary pattern (CSLBP) [15], etc.

The research in the recent years has started to focus on directional information instead of intensity information. The reason for that is because the directional encoded is more stable than the pixel intensity [16]. Jabid *et al.*[6] presented a low-level feature, Local Directional Pattern, which uses the edge responses of eight different directions about each pixel. Luo *et al.*[16] presented the local line directional pattern (LLDP) using the line direction response instead of the gradient response. Shabat and Tapamo[7] presented the directional local binary pattern (DLBP) using the center pixel as a threshold for the eight directional response values of the neighborhood.

Motivated by LBP and LDP , in this paper we propose a new LDP-structure descriptor, Circular Local Directional Pattern.

## 2 Features Methods for texture analysis

The most important process in texture analysis is the features extraction. In this section, Local Binary Pattern (LBP),

---

A.M. Shabat  
School of Engineering,  
Howard College Campus,  
University of KwaZulu-Natal,  
Durban, 4041, South Africa  
Tel.: +27-74-2804107  
E-mail: abshabat@gmail.com

J.R. Tapamo  
School of Engineering,  
Howard College Campus,  
University of KwaZulu-Natal,  
Durban, 4041, South Africa  
E-mail: tapamoj@ukzn.ac.za

Local Directional Pattern (LDP) and the proposed Circular Local Directional Pattern (CILDP) are presented below.

## 2.1 Local Binary Pattern

Local Binary Pattern changes the value of each pixel in the image. This modification is based on the relationship of each pixel in its 8-neighborhood. For each pixel  $p$ , LBP operator makes its gray value  $g(p)$  a threshold, and for each neighbor  $(n_i(p))$  (for  $i = 0, 1, \dots, 7$ ) if its gray value is greater than  $g(p)$  then set gray value corresponding binary LBP bit code to 1 binary bit, otherwise, set it to 0. At the end convert the binary code obtain into decimal. An example is shown in Fig.1. When the LBP codes are produced for all the pixel of an image, an histogram of these codes is generated and can be used as a texture feature.

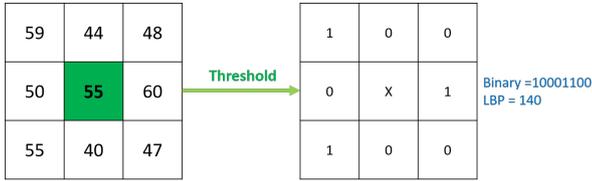


Fig. 1: LBP Example

## 2.2 Local Directional Pattern

The LDP operator calculates an eight-bit binary code by comparing the different directional edge response values in the 8-neighborhood of each pixel. The directional edge response vectors are computed using Kirsch Mask  $M_n$  in a given 3 neighborhood.  $M_n$  have eight different rotations ( $M_0, \dots, M_7$ ) as shown in Fig.2.

$$m_n = p_n * M_n(x, y), n = 1, \dots, 7 \quad (1)$$

Where  $p_n$  is the gray value of the  $n^{th}$  neighbor.

$$\begin{array}{cccc} \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\ M_0(\text{East}) & M_1(\text{North East}) & M_2(\text{North}) & M_3(\text{North West}) \\ \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\ M_4(\text{West}) & M_5(\text{South West}) & M_6(\text{South}) & M_7(\text{South Est}) \end{array}$$

Fig. 2: Kirsch Masks

The  $k$  most significant response are chosen to generate the binary code. Hence the top  $k^{th}$  values are set to be 1, leaving

$(8 - k)$  values to 0. LDP is derived using Equation 2.

$$LDP_{x,y}(m_0, m_1, \dots, m_7) = \sum_{n=0}^7 s(m_n - m_k) \times 2^n \quad (2)$$

where  $m_k$  is the most prominent directions and  $s(x)$  is define as

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

An example is shown in Fig.3.



Fig. 3: LDP Example

## 2.3 Circular Local Directional Pattern

In contrast to the regular LDP, which uses eight pixels in  $(3 \times 3)$  window, CILDP uses the circle shape to allocate a set of points using different radius (1, 2, 3) as shown in Fig.4. Which lead to better analysis specially with textures with different scale. However, it is limited to eight pixels because the kirsch mask has eight values only.

Consider an image  $I$  and let  $p_c$  be the arbitrary pixel at the point  $(x_c, y_c)$ . Moreover, let  $p_n$  denote to the gray value of a sampling point in an evenly spaced circular neighborhood of 8 points and radius  $R$  around  $p_c$ .

$$p_n = I(x_n, y_n), \quad n = 0, \dots, 7 \quad (3)$$

Where

$$x_n = x_c + R \cos(2\pi p/8) \quad (4)$$

$$y_n = y_c + R \sin(2\pi p/8) \quad (5)$$

Given a point  $p_c = (2, 2)$  and radius  $R = 2$ . The circular neighborhood of 8 points  $(p_0, \dots, p_7)$  are located using equations 4 and 5. Table 1 shows the coordinates  $x_n$  and  $y_n$  in each point.

Bilinear interpolation is employed to compute the gray value  $p_n$  for all that points that doesn't correspond with the center of a pixel as you saw in example 1, point  $p_1, p_3, p_5$  and

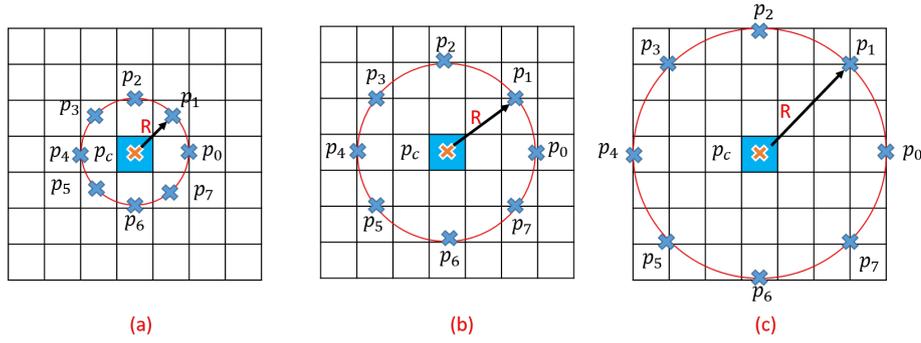


Fig. 4: The circular a.(8,1) b.(8,2) c.(8,3) neighborhoods. Bilinear interpolation is used to compute all the points that doesn't fall in the center of a pixel.

Table 1: Example 1- Compute the circular neighborhood of 8 points in radius  $R = 2$

$p_n$	$x_n$	$y_n$
$p_0$	4	2
$p_1$	3.14	0.585
$p_2$	2	0
$p_3$	0.585	0.585
$p_4$	0	2
$p_5$	0.585	3.41
$p_6$	2	4
$p_7$	3.41	3.41

$p_7$  doesn't corresponding with any point in the image matrix. The equations below, compute bilinear interpolation to determine the gray value of the unknown point using four known neighbor points as shown in Fig.5.

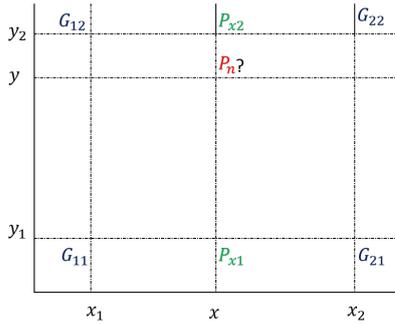


Fig. 5: Finding the gray value  $p_n$  using four known points ( $G_{11}, G_{12}, G_{21}$  and  $G_{22}$ )

$$p_{x1} = \frac{x_2 - x}{x_2 - x_1} G_{11} + \frac{x - x_1}{x_2 - x_1} G_{21} \quad (6)$$

$$p_{x2} = \frac{x_2 - x}{x_2 - x_1} G_{12} + \frac{x - x_1}{x_2 - x_1} G_{22} \quad (7)$$

$$p_n = \frac{y_2 - y}{y_2 - y_1} p_{x1} + \frac{y - y_1}{y_2 - y_1} p_{x2} \quad (8)$$

Where  $G_{11}, G_{12}, G_{21}$  and  $G_{22}$  is the gray value for four different points in different location.

After all the points around a particular pixel are declared, eight directional edge response vectors are computed using Krirsch mask  $M_n$ .  $M_n$  have eight different orientations ( $M_0, \dots, M_7$ ).

$$m_n = p_n * M_n(x, y), n = 1, \dots, 7 \quad (9)$$

In order to generate the CILDP code, the  $k$  most significant response are chosen to generate the binary code. Hence the top  $k^{th}$  values are set to be 1, leaving  $(8 - k)$  values to 0. CILDP is derived using Equation 10.

$$CILDP = \sum_{n=0}^7 s(m_n - m_k) \times 2^n \quad (10)$$

where  $m_k$  is the most prominent directions and  $s(x)$  is define as

$$s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

---

#### Algorithm 1 Extraction of CILDP feature of an Image

---

**Input:**  $I$  // image

**Output:**  $H$  // CILDP Histogram of the image  $I$

- 1: Convert  $I$  into gray level  $I_G$
  - 2: Compute the circular neighborhood for the eight points using different radiuses.
  - 3: Calculate the directional information by applying Kirsch Masks on the original image.
  - 4: Compute the CILDP code by selecting the three most significant directional responses.
  - 5: Compute the histogram  $H$  from the transformed image.
-

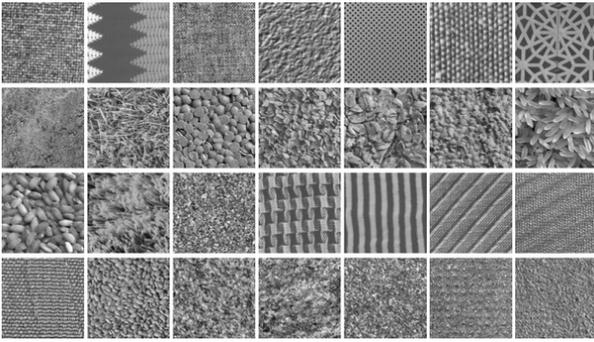


Fig. 6: Example of one sample of each category in the 28 classes in the Kylberg dataset

### 3 Experimental results and discussion

In this section, a description for the classification performance applied to Kylberg Dataset using the proposed method CILDP, LDP and LBP are presented. The proposed methods are implemented using python framework, scikit-learn library and opencv.

#### 3.1 Image Dataset

To evaluate the proposed method Kylberg image dataset is used. The Kylberg dataset consists of 4480 texture surfaces of 28 categories, with 160 samples per category as shown in Fig.6. The images are homogeneous in terms of illumination and scale. The standard size of each sample is 576 pixels and it is available in different rotations  $\theta \in [0, \frac{1}{6}\pi, \frac{2}{6}\pi, \dots, \frac{11}{6}\pi]$ .

#### 3.2 Classification Evaluation

The effectiveness of these methods were evaluated using different classification measures, learning curve, accuracy, precision, recall, F-score and Cohen's kappa. The learning curve is a very useful algorithm that evaluate the sanity of an algorithm. It plots the relation between the training set size and the performance. In a basic manner it shows the starting point where the classifiers begins to learn. Accuracy is the number of samples classified correctly, for example if the classifier accuracy is 50% it means that the classifier manage to classify correctly 50% of the dataset

$$Accuracy = \frac{\# \text{ of samples correctly classified}}{\# \text{ of samples}} \quad (11)$$

Precision is the ratio of a number of positive predictions to all the number of positive classes value predicted.

$$precision = \frac{TP}{TP + FP} \quad (12)$$

Sensitivity is the ratio of a number of positive predictions to all number of positive classes in test data.

$$recall = \frac{TP}{TP + FN} \quad (13)$$

Where  $TP$  is the number of samples correctly classified as positive,  $FP$  is the number of samples incorrectly classified as positive and  $FN$  is the number of samples incorrectly classified as negative.

$F$  - score conveys the balance between the precision and the recall.

$$F - score = 2 \frac{precision * recall}{precision + recall} \quad (14)$$

Cohen's kappa is a very good measure that can handle very well both multi-class and imbalanced class problem. It calculates the agreement between categorical data. If the value is less than or equal 0 it indicates that the classifier is useless. Table 2 shows the interpretation of Kappa value.

Table 2: The interpretation of Kappa

Interpretation	conditions on $\kappa$
Poor agreement	$\kappa \leq 0.20$
Fair agreement	$0.20 < \kappa \leq 0.40$
Moderate agreement	$0.40 < \kappa \leq 0.60$
Good agreement	$0.60 < \kappa \leq 0.80$
Very good agreement	$0.80 < \kappa \leq 1.00$

The dataset is split into two datasets 80% as a training set and 20% test set using 10 cross validations. Each classifier is trained using different parameters as shown in Table 3.

Table 3: Classifiers parameters

Classifiers	Parameters
SVM	Polynomial linear kernel Configuration parameter $c = 0.025$
k-NN	$k=5$
DT	Entropy The minimum number of split is 10
RF	The number of the trees is 10 The maximum depth of the tree is 5
Perceptron	The number of passes over the training data = 100 Constant $\eta = 0.1$

#### 3.3 Result Discussions

The performance of the proposed CILDP, LBP and LDP are tested in texture classification problem using Kylberg dataset. In this paper, five classifiers have been applied k-nearest neighbours, support vector machine, Decision Tree,

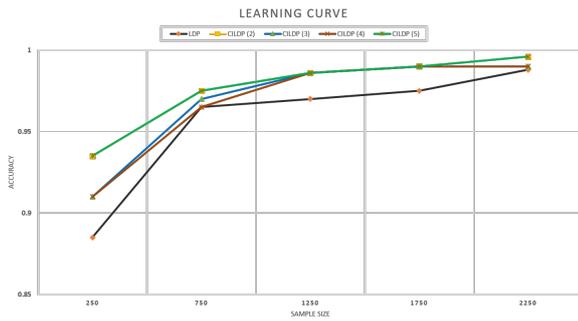


Fig. 7: The Learning Curve of LDP and CILDP

random forest and perceptron. Table 4 shows the performance rate of CILDP, LDP and LBP using five different classifiers on Kylberg dataset. Overall, it can be seen that the CILDP in various distances were matched or higher than the other methods. Fig.7 shows the learning curve for the proposed method CILDP and LDP using SVM as a classifier. Learning curve describes the relation between the performance and the experience, where the performance is measured by the accuracy and the experience is the amount of the training dataset or the number of iterations (cross validation) used to enhance the parameter of a classifier. In each graph there are two lines, the average square error on training set and the average square error on cross validation set. We mentioned that the performance rises when the training samples are between 250 to 750. Above 1500 the classifier starts to gain less knowledge and doesn't improve much. The CILDP method in different distance parameters did very well compared to the LDP.

CILDP had the highest performance at 99% using SVM. CILDP improved the performance range from 1% to 24% in all the classifiers in Table 4. For example, CILDP performance using Perceptron classifier is 87%, while LDP performance is 74% and LBP is 63%. This shows performance improvement of 11% compared to LDP and 24% in LBP. Table 4 illustrates that the proposed method CILDP either matched or outperformed LDP and LBP.

Table shows the performance of SVM using 10% as a training dataset and the remains as a test dataset.

In table5 different classifications evaluation is used to evaluate the algorithms CILDP and LDP. 10% of the dataset were used as a training dataset and the remainder as a test dataset. CILDP at distance 5 was the best Cohen kappa value with 96% which interpreter as very good agreement. All the results indicate the superiority of CILDP compared with LDP.

## 4 Conclusion

In this paper, we proposed a new texture feature method CILDP. The method is based on the circle shape to compute the directional edge vector using different radius. To evaluate the performance of the proposed method a comparison experiment between CILDP, LDP and LBP have been done using five different classifiers to classify 28 categories of texture from Kylberg dataset. The result establishes the effect of using various radius between the center pixel and the points around it on the performance. More investigation will be done in the future using different datasets and parameters.

## References

- Pietikinen M, Hadid A, Zhao G, Ahonen T. Local binary patterns for still images. In *Computer vision using local binary patterns*. Springer London. pp. 13-47 (2011).
- Haralick RM, Shanmugam K. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*. Nov(6), pp. 610-21 (1973).
- He DC, Wang L. Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing*. Jul, 28(4), pp. 509-12 (1990).
- Lowe DG. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*. Nov 1, 60(2), pp. 91-110 (2004).
- Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (SURF). *Computer vision and image understanding*. Jun 30, 110(3), pp. 346-59 (2008).
- Jabid T, Kabir MH, Chae O. Local directional pattern (LDP) for face recognition. In *Proc. of Consumer Electronics Conference (ICCE)*, Digest of Technical Papers International. Jan 9, pp. 329-330 (2010).
- Shabat AM, Tapamo JR. Directional Local Binary Pattern for Texture Analysis. In *International Conference Image Analysis and Recognition*. Jul 13, pp. 226-233 (2016). Springer International Publishing.
- Soh LK, Tsatsoulis C. Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on geoscience and remote sensing*. Mar, 37(2), pp. 780-95 (1999).
- Memp T, Pietikinen M. Texture analysis with local binary patterns. *Handbook of Pattern Recognition and Computer Vision*. 3, pp. 197-216 (2005).
- Shabat AM, Tapamo JR. A comparative study of Local directional pattern for texture classification. In *World Symposium on Computer Applications & Research (WSCAR)*. Jan 18, pp. 1-7 (2014).
- Ahonen T, Hadid A, Pietikinen M. Face recognition with local binary patterns. *European conference on computer vision*. pp. 469-81 (2004).
- Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*. Jul, 24(7), pp. 971-87 (2002).
- Liao S, Law MW, Chung AC. Dominant local binary patterns for texture classification. *IEEE transactions on image processing*. May, 18(5), pp. 1107-18 (2009).
- Guo Z, Zhang L, Zhang D. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*. Jun, 19(6), pp. 1657-63 (2010).

Table 4: The performance accuracy of CILDP, LDP and LBP applied on Kylberg dataset using five classifiers

<b>Classifiers</b>	<b>LBP</b>	<b>LDP</b>	Circular LDP (Distance 2)	Circular LDP (Distance 3)	Circular LDP (Distance 4)	Circular LDP (Distance 5)
<b>k-NN</b>	<b>93</b>	91	90	<b>93</b>	91	<b>93</b>
<b>SVM</b>	98	98	98	<b>99</b>	98	98
<b>DT</b>	87	89	<b>91</b>	89	<b>91</b>	<b>91</b>
<b>RF</b>	70	64	72	75	<b>77</b>	75
<b>perceptron</b>	63	74	71	<b>87</b>	86	84

Table 5: Using different evaluation measures to evaluate CILDP and LDP

<b>Feature</b>	<b>Accuracy</b>	<b>Cohen Kappa</b>	<b>f1_score</b>	<b>precision</b>	<b>recall</b>
LDP	0.92	0.9	0.9	0.89	0.91
CILDP 2	0.95	0.94	0.95	0.96	0.95
CILDP 3	0.93	0.92	0.93	0.925	0.93
CILDP 4	0.94	0.93	0.94	0.94	0.935
CILDP 5	0.95	0.96	0.95	0.95	0.95

15. Heikkil M, Pietikinen M, Schmid C. Description of interest regions with center-symmetric local binary patterns. In ICVGIP, Dec 13, 6, pp. 58-69 (2006).
16. Luo YT, Zhao LY, Zhang B, Jia W, Xue F, Lu JT, Zhu YH, Xu BQ. Local line directional pattern for palmprint recognition. Pattern Recognition. Feb 29, 50, pp. 26-44 (2016).



# Chapter 8

## Discussion

In previous chapters, we introduced five enhancements to the LDP method, and all of these improvements demonstrated their effectiveness and higher performance compared to LDP. In this chapter, we compare LDP enhancements, using two types of datasets. The chapter is split into two sections, the first applying the proposed methods on the KTH-TIPS2b dataset and the second presenting the application of the methods on facial expression recognition.

### 8.1 Classification evaluation

The effectiveness of the proposed methods were evaluated using different classification measures, learning curve, accuracy, confusion matrix, precision, sensitivity, F-score and Cohen's kappa. The learning curve is a very useful method that evaluates the sanity of an algorithm. It plots the relation between the training set size and the performance. Basically it shows the point where the classifier begins to learn. Accuracy is the rating of samples classified correctly over the total number of samples. Equation 8.1, gives the formula to calculate the accuracy.

$$Accuracy = \frac{\text{No. of samples correctly classified}}{\text{No. of the samples}} \quad (8.1)$$

Another way to evaluate a classifier is to use confusion matrix. Confusion matrix is an illustrated table which allows you to view the performance of the classifier in detail and it highlights the correct classification and misclassification areas. As shown in Figure 8.1, each column displays the predicted class and each row provides the actual class.

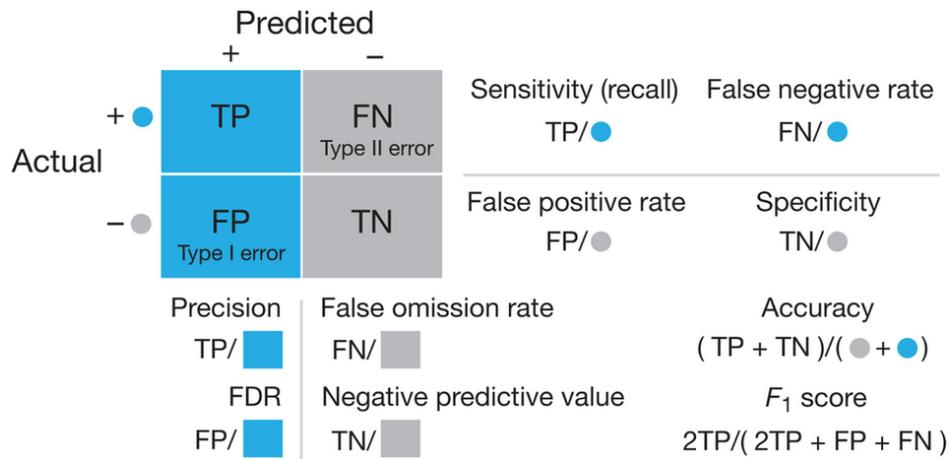


Fig. 8.1 "Blue and gray circles indicate cases known to be positive (TP + FN) and negative (FP + TN), respectively, and blue and gray backgrounds/squares depict cases predicted as positive (TP + FP) and negative (FN + TN), respectively. Equations for calculating each metric are encoded graphically in terms of the quantities in the confusion matrix. FDR, false discovery rate."Lever et al. (2016)

Precision is the ratio of a number of positive predictions to all the number of positive classes of value predicted.

$$precision = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (8.2)$$

Sensitivity is the ratio of a number of positive predictions to all number of positive classes in test data.

$$recall = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \quad (8.3)$$

F-score conveys the balance between the precision and the recall.

$$F\ score = 2 \times \left( \frac{precision \times recall}{precision + recall} \right) \quad (8.4)$$

Cohen's kappa (Pontius Jr and Millones, 2011) is a very good measure that can handle both multi-class and imbalanced class problems very well. It calculates the agreement between categorical data. If the value is less than or equal to 0, it indicates that the classifier is useless. Table 8.1 shows the interpretation of Kappa values.

Table 8.1 Interpretation of Kappa( $\kappa$ )

Strength of argument	Value of $\kappa$
Poor agreement	$\kappa < 0.20$
Fair agreement	$0.20 < \kappa \leq 0.40$
Moderate agreement	$0.40 < \kappa \leq 0.60$
Good agreement	$0.60 < \kappa \leq 0.80$
Very good agreement	$0.80 < \kappa \leq 1.00$

## 8.2 Applying the enhancement methods to KTH-TIPS2b dataset

### 8.2.1 Dataset

The KTH-TIPS2b dataset (Fritz et al., 2004) has images of eleven materials, containing four different samples for each material. Each image was captured in diverse scale and illumination conditions at a resolution of  $1280 \times 960$  pixels. All images were cropped to  $200 \times 200$  pixels. Some samples are shown in Figure 8.2. The first experiment performs texture analysis of the KTH dataset, using six types of classifiers.



Fig. 8.2 KTH-TIPS2b samples

### 8.2.2 Experimental Results

To display the outcomes in a clear and correct manner, we present the results of each features method separately, that is, we present LDP, ALDP, DLBP, CLDP, SLDP and CILDP. Figure 8.3 and 8.4 shows the histogram of both methods LDP and LBP.

#### Learning curve

Figures 8.5 to 8.15 show the learning curve for the proposed methods and LDP. The x-axis represents the training samples and the y-axis represents the accuracy. The learning curve shows the performance versus the size of the data used for training.

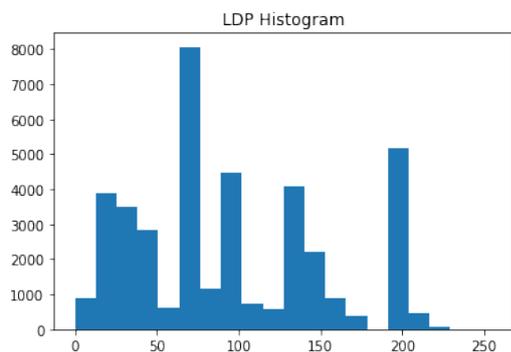


Fig. 8.3 The histogram applied on LDP image

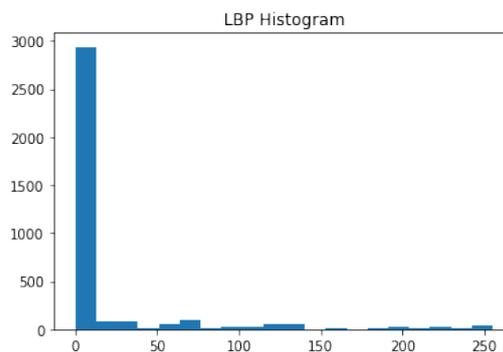


Fig. 8.4 The histogram applied on LDP image

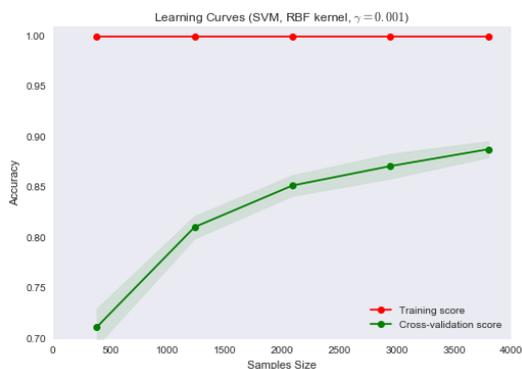


Fig. 8.5 The Learning Curve of LDP

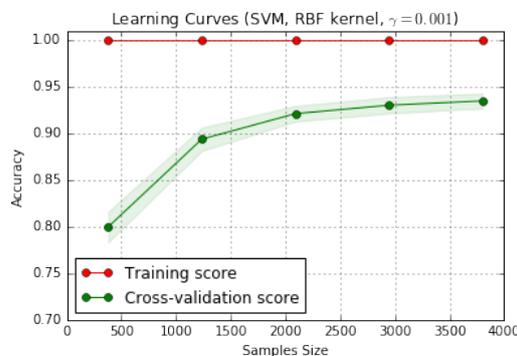


Fig. 8.6 The Learning Curve of ALDP

The CLDP in both directions (Up and Down) was the fastest in learning, with an accuracy between 80% and 85% when the size of the data used was between 0 and 500 samples. The other methods were slower learners, with an accuracies of 70%, except for the ALDP, which scored an accuracy of 80%, and LDP (Negative), which scored an accuracy of 75%.

We also observed that the performance in all the methods increased rapidly when the training dataset size was from 500 to 1500 samples by about 10%.

We encountered that the performance increased very slowly after the training dataset size reached above of 2500 samples. That is, the classifier begins to gain less knowledge by an accuracy of 5%.



Fig. 8.7 The Learning Curve of DLBP

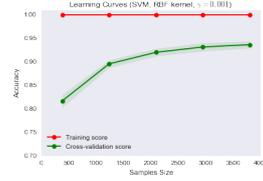


Fig. 8.8 The Learning Curve of CLDP (UP)

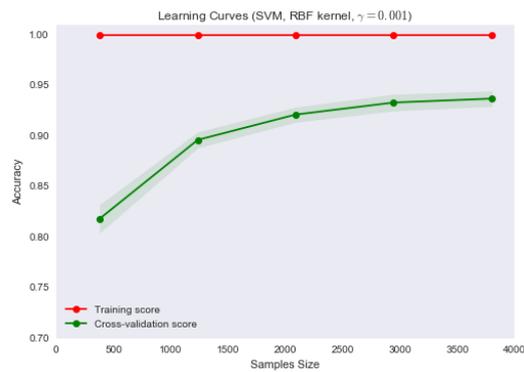


Fig. 8.9 The Learning Curve of CLDP(DOWN)

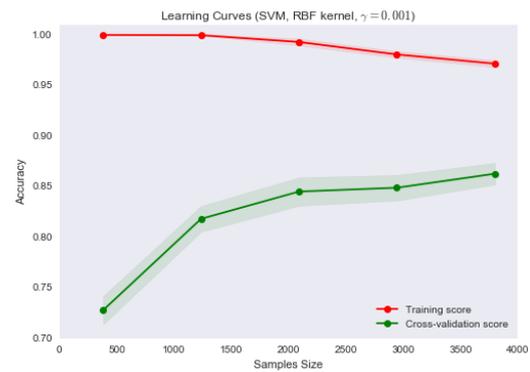


Fig. 8.10 The Learning Curve of LDP(Positive)

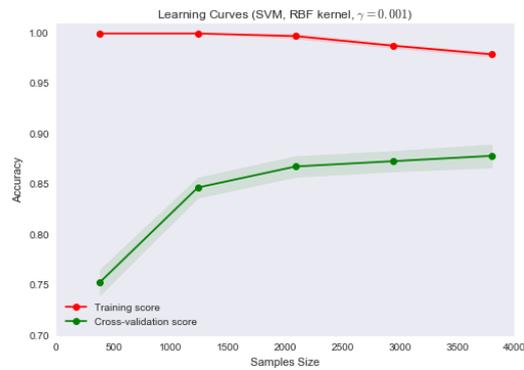


Fig. 8.11 The Learning Curve of LDP(Negative)

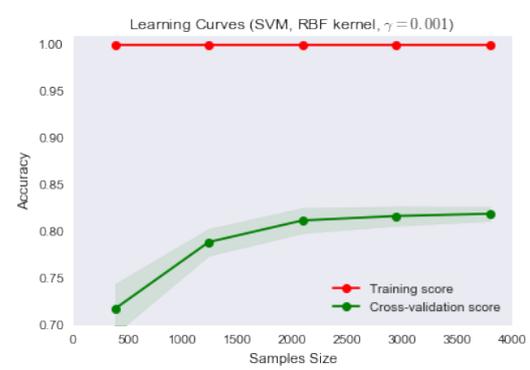


Fig. 8.12 The Learning Curve of CILDP(2)

**Classification report**

Figures 8.16 to 8.26 below demonstrate the classification evaluation for each of the textural classes. The x-axis represents the following evaluation criteria precision; recall, F-score, and the y-axis representing 11 textural classes.

We provide a detailed explanation for the strength of detecting and recognizing each class of the textural classes separately. We selected the F-score to compare and evaluate the

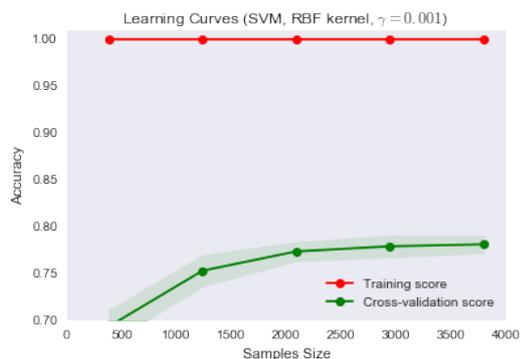


Fig. 8.13 The Learning Curve of CILDP(3)

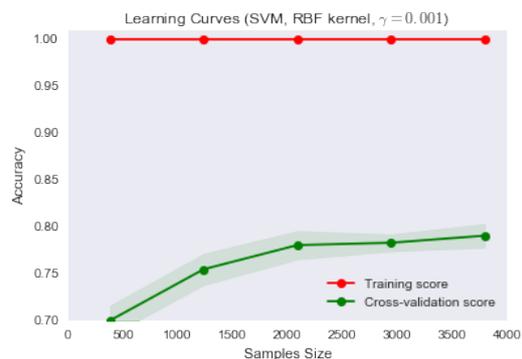


Fig. 8.14 The Learning Curve of CILDP(4)

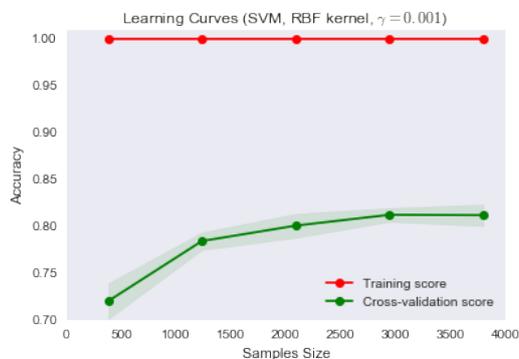


Fig. 8.15 The Learning Curve of CILDP(5)

methods proposed; this is because it is based on the other two evaluation measures (precision and recall). The CLDP (Up and Down) method was the best at extracting the most useful data from the images. The classification rate for the classifier was the highest when SLDP features were used to determine the textural classes of wool, cracker, brown\_bread, corduroy, cork, white\_bread and aluminum, with an F-score between 80% and 98%, in contrast to LDP, which had a worse performance with a loss between 5% and 17% of accuracy rate.

It was also noted that in wood, linen and lettuce classes, the ALDP was the most capable of extracting valuable information for the classifier. Its classification performance was superior to the performance of the LDP classification by 2% to 12%.

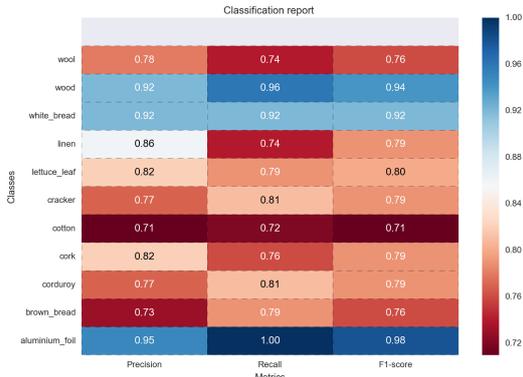


Fig. 8.16 The Classification Report of LDP

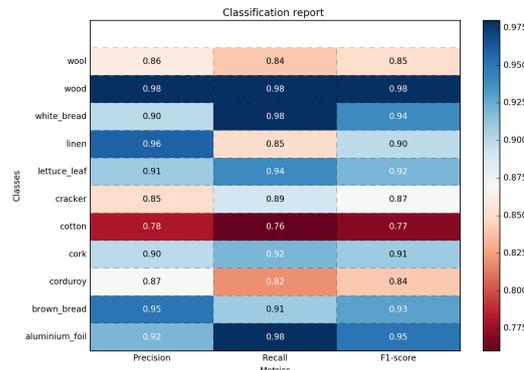


Fig. 8.17 The Classification Report of ALDP



Fig. 8.18 The Classification Report of DLBP



Fig. 8.19 The Classification Report of CLDP (UP)

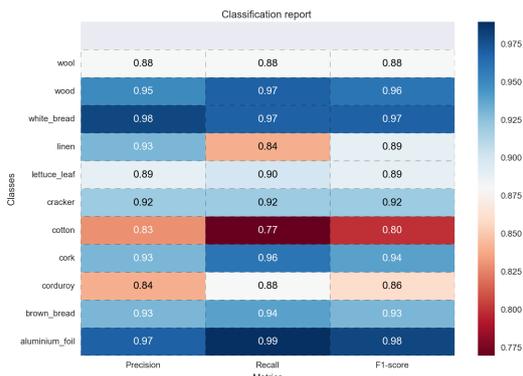


Fig. 8.20 The Classification Report of CLDP(DOWN)

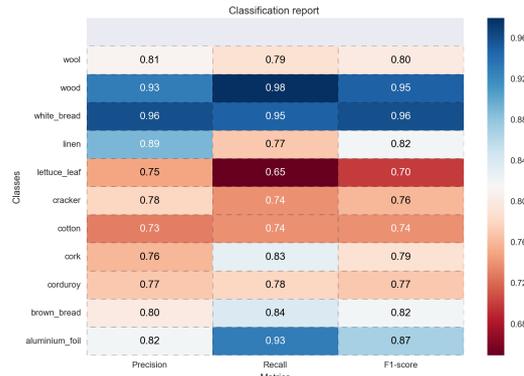


Fig. 8.21 The Classification Report of LDP(Positive)

**Average Kappa Scores**

We added the average Kappa scores to verify the performance of all the proposed methods, as shown in Table 8.2. We noted that the results do not show anything new, as they correspond to the results presented in the learning curve and the classification reports.

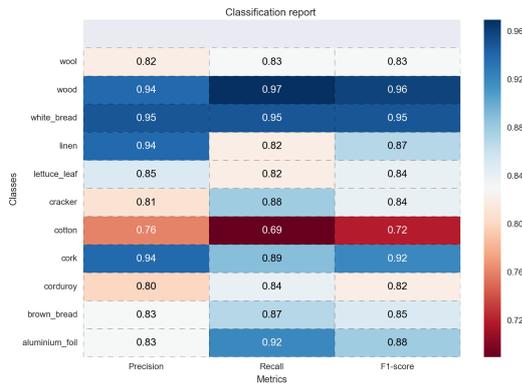


Fig. 8.22 The Classification Report of LDP(Negative)

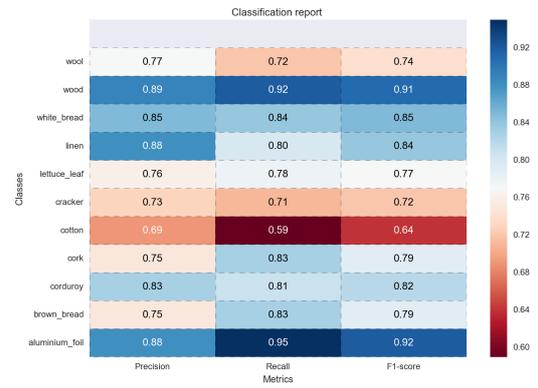


Fig. 8.23 The Classification Report of CILDP(2)



Fig. 8.24 The Classification Report of CILDP(3)

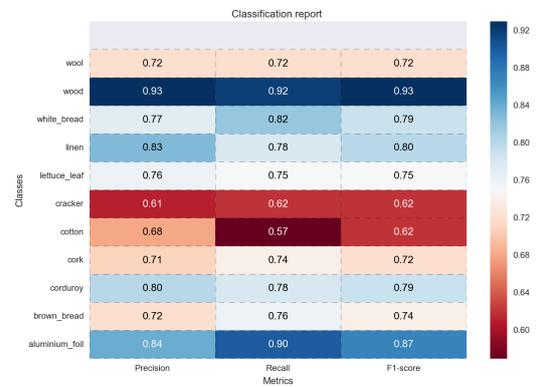


Fig. 8.25 The Classification Report of CILDP(4)

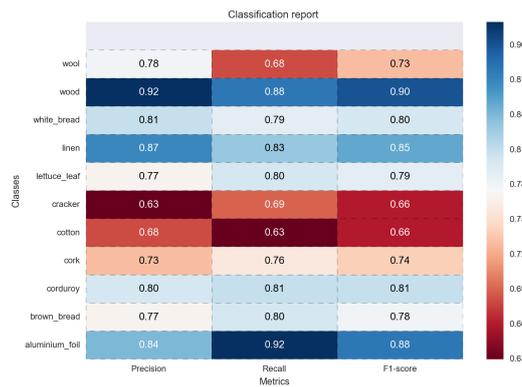


Fig. 8.26 The Classification Report of CILDP(5)

Table 8.2 Average kappa scores for all the methods applied on the KTH-TIPS2b dataset

Feature	SVM	SVM ( C )	NB	k-NN	Perceptron	DT(Entropy)
<b>LDP</b>	0.85	0.89	0.55	0.7	0.58	0.66
<b>ALDP</b>	<b>0.91</b>	0.95	<b>0.64</b>	0.84	0.67	<b>0.75</b>
<b>CLDP (Down)</b>	<b>0.91</b>	<b>0.96</b>	0.61	<b>0.85</b>	<b>0.81</b>	<b>0.75</b>
<b>CLDP (Up)</b>	<b>0.91</b>	<b>0.96</b>	0.61	<b>0.85</b>	0.79	<b>0.75</b>
<b>CILDP (2)</b>	0.8	0.88	0.45	0.78	0.52	0.68
<b>CILDP (3)</b>	0.77	0.86	0.43	0.8	0.52	0.67
<b>CILDP (4)</b>	0.77	0.86	0.42	0.79	0.6	0.65
<b>CILDP (5)</b>	0.78	0.85	0.42	0.78	0.6	0.65
<b>DLBP</b>	0.83	0.89	0.54	0.7	0.63	0.69
<b>LDP Negative</b>	0.86	0.92	0.52	0.75	0.6	0.68
<b>LDP (Positive)</b>	0.83	0.91	0.53	0.77	0.6	0.69

## 8.3 Facial expression application

### 8.3.1 Dataset

The Extended Cohn-Kanade Dataset (CK+)(Lucey et al., 2010) has 593 sequences from 123 persons. For each person seven facial expressions were captured showing neutrality, sadness, surprise, happiness, fear, anger and disgust. The size of each image is  $640 \times 490$  pixel. Figure 8.27 shows a sample of each expression.



Fig. 8.27 Samples of the CK+ dataset

### 8.3.2 Experimental Results

To display the outcomes in a clear and correct manner, we present the results of each features method separately. That is, we present the results in the following order: LDP, ALDP, DLBP, CLDP, SLDP and CILDP.

#### Learning Curve

Figures 8.28 to 8.38 show the learning curve for the proposed methods and LDP. The x-axis represents the training samples and the y-axis represents the accuracy. The learning curve shows the performance versus the size of the data used for training. The classifier learned faster when it used the SLDP feature in both gradients (Positive and Negative), with an accuracies of 90% when the size of the training samples was between 0 and 500. The accuracy of LDP, ALDP, DLBP and CLDP (Up and Down) was less than 90%, using the same training samples. However, CILDP (3,4,5) had a worse accuracy of less than 80%, but this was expected, since CILDP works better on large-scale texture. The performance of most of the methods increased rapidly by 10% when the training sample size was between 500 and 1500 samples. When the size of the sample reached 1500 samples in SLDP (Positive and Negative), the classifier became sufficient and the amount of learning became constant, with a very high performance of 99%. This indicates the effectiveness of the SLDP, as the classifier does not require a larger quantity of samples for training.

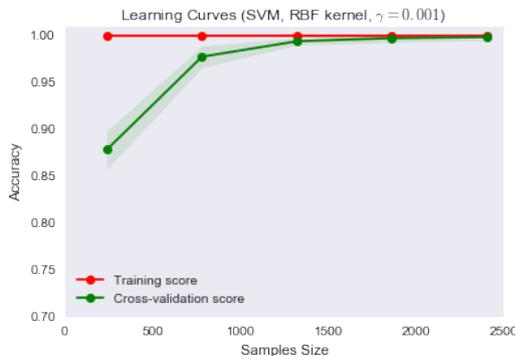


Fig. 8.28 The Learning Curve of LDP

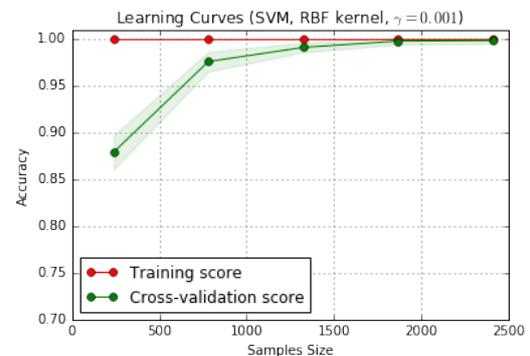


Fig. 8.29 The Learning Curve of ALDP

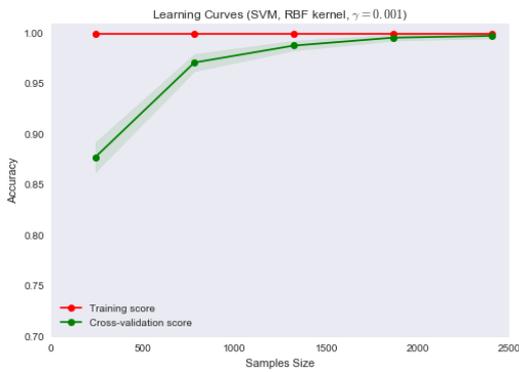


Fig. 8.30 The Learning Curve of DLBP



Fig. 8.31 The Learning Curve of CLDP (UP)

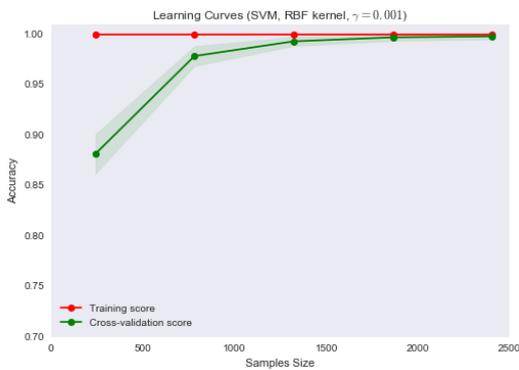


Fig. 8.32 The Learning Curve of CLDP(DOWN)

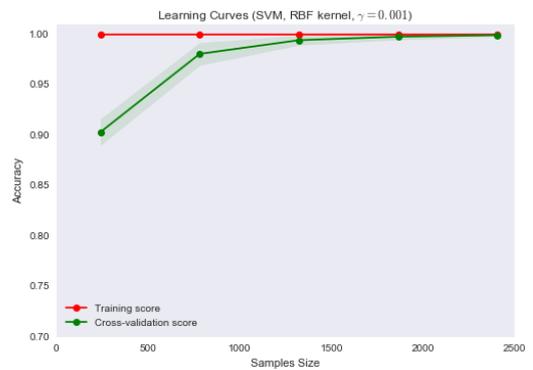


Fig. 8.33 The Learning Curve of LDP(Positive)

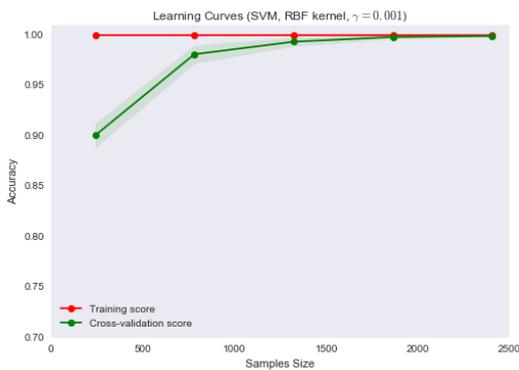


Fig. 8.34 The Learning Curve of LDP(Negative)

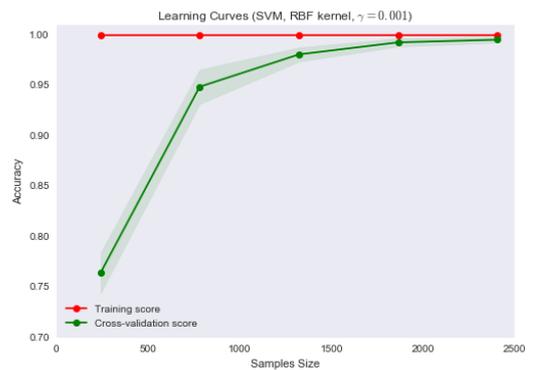


Fig. 8.35 The Learning Curve of CILDP(2)

**Classification report**

Figures 8.39 to 8.49 below demonstrate the classification evaluation for each facial expression emotion class (surprise, sadness, joy, fear, disgust, contempt and anger). The x-axis represents

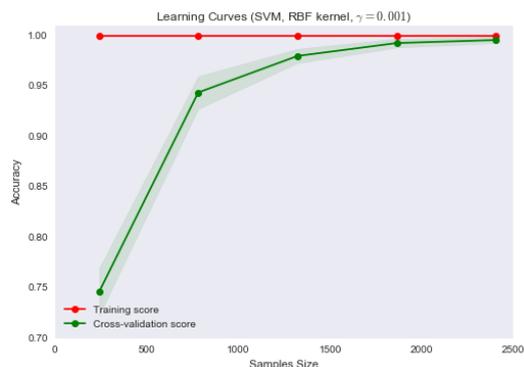


Fig. 8.36 The Learning Curve of CILDP(3)

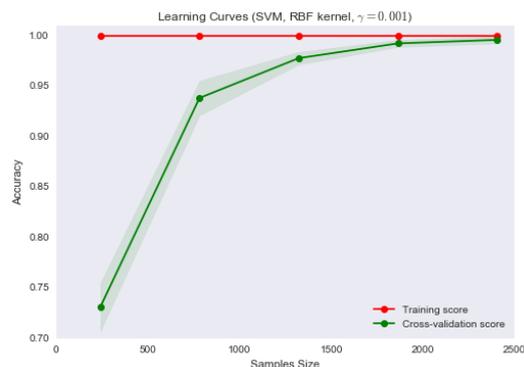


Fig. 8.37 The Learning Curve of CILDP(4)

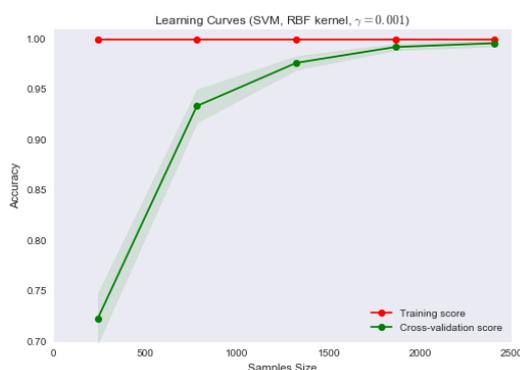


Fig. 8.38 The Learning Curve of CILDP(5)

the evaluation criteria: precision, recall, F-score, and the y-axis representing seven facial expression emotion classes.

We provide a detailed explanation of the strength of detecting and recognizing each class of facial expression separately. F-score was selected to compare and evaluate the execution of the methods provided. This is because it is based on the other two evaluation measures (precision and recall). We began with our first comparison with the surprise expression, and found that the best method to recognize this expression was SLDP in both directions (Positive and Negative), with 98% compared to the LDP 92%. The worst method to identify the surprise expression was CILDP, using distance 5, which was expected because CILDP is best when large-scale sized texture is used.

For the facial expression joy, the LDP was the worst identifier, with a success rate of about 94% compared to the rest of the methods where the rate of the joy recognition was above 97%.

For the third time, the SLDP (Positive and Negative) method was found to be the best to recognize facial expressions. The recognition rate of the fear expression was about 97%, and the other methods ranged from 88% to 95%.

Likewise, with the disgust expression, LDP did not get the best recognition rate, as it performed 2% less than the best performance of SLDP (Positive and Negative) which was an F-score of about 98%.

For both contempt and anger expressions, all methods performed equally with a very high success rates of 99% to 100%, except for the LDP, where the F-score was 98% for the anger expression.

All Methods were generally good in classifying the facial expressions, with F-scores between 88% and 100%, and the SLDP in both directions, positive and negative had the upper hand in recognizing all facial expressions.

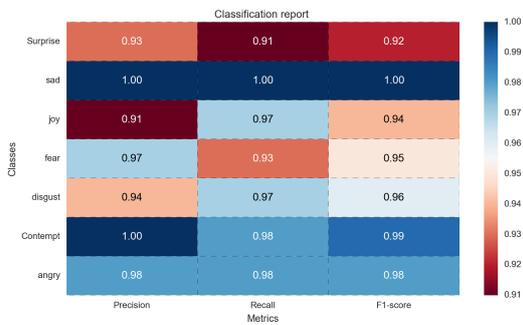


Fig. 8.39 The Classification Report of LDP

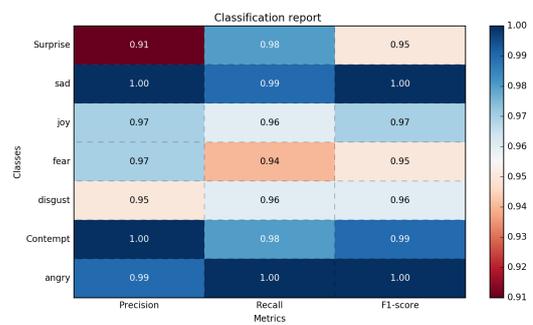


Fig. 8.40 The Classification Report of ALDP

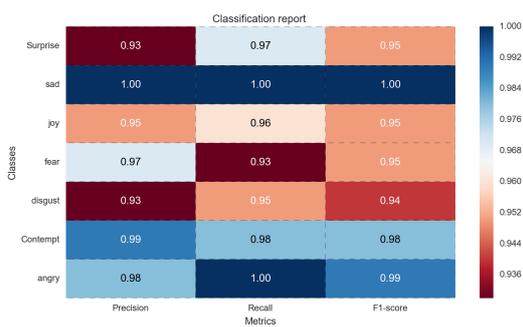


Fig. 8.41 The Classification Report of DLBP

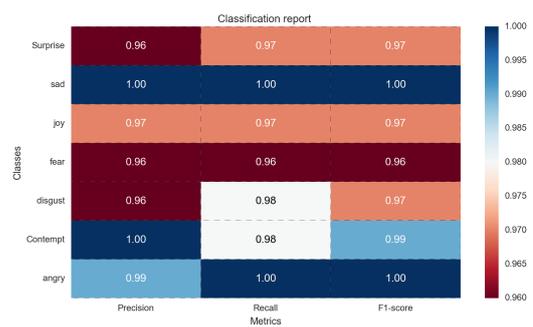


Fig. 8.42 The Classification Report of CLDP (UP)

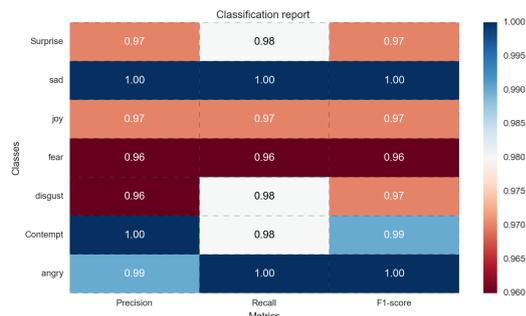


Fig. 8.43 The Classification Report of CLDP(DOWN)

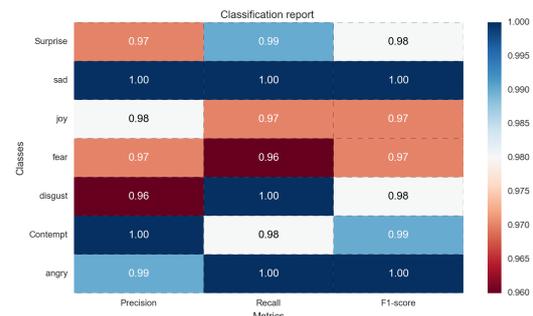


Fig. 8.44 The Classification Report of LDP(Positive)

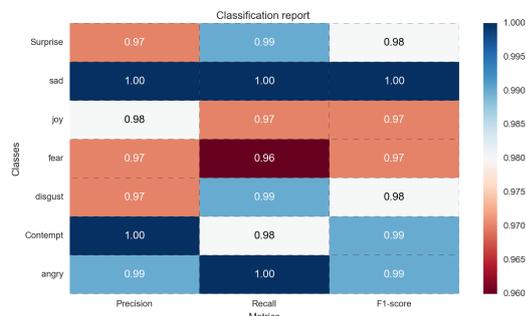


Fig. 8.45 The Classification Report of LDP(Negative)



Fig. 8.46 The Classification Report of CILDP(2)



Fig. 8.47 The classification Report of CILDP(3)

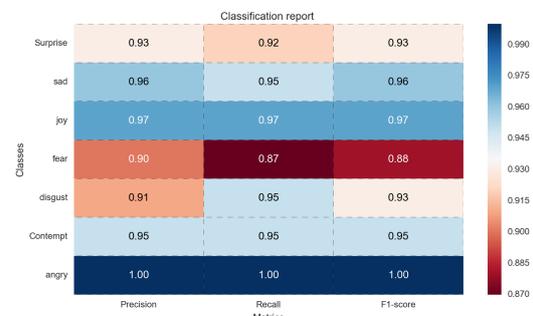


Fig. 8.48 The classification Report of CILDP(4)

**Average Kappa scores**

We added the average Kappa scores to verify the performance of all the proposed methods, as shown in Table 8.3. Note that the results did not come with anything new, as they correspond to the results presented in the learning curve and the classification report.

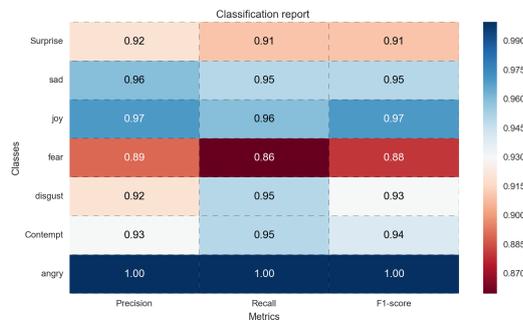


Fig. 8.49 The Classification report of CILDP(5)

Table 8.3 The average kappa scores for all the methods applied on the Cohn-Kanade Dataset

Feature	SVM	SVM(C)	NB	k-NN	Perceptron	DT
<b>ALDP</b>	0.99	0.99	<b>0.78</b>	0.91	<b>0.99</b>	0.72
<b>CLDP (Down)</b>	<b>1</b>	0.99	0.66	<b>0.92</b>	<b>0.99</b>	0.65
<b>CLDP (Up)</b>	<b>1</b>	0.99	0.62	0.91	<b>0.99</b>	0.64
<b>CILDP (2)</b>	0.99	0.99	0.63	0.9	0.98	0.57
<b>CILDP (3)</b>	0.99	0.98	0.65	0.9	0.98	0.71
<b>CILDP (4)</b>	0.99	0.98	0.66	0.91	0.98	0.64
<b>CILDP (5)</b>	0.99	0.98	0.68	0.91	0.98	0.68
<b>DLBP</b>	0.99	0.99	0.48	0.91	<b>0.99</b>	0.74
<b>LDP</b>	<b>1</b>	<b>1</b>	0.65	0.91	<b>0.99</b>	0.69
<b>LDP (Neg)</b>	<b>1</b>	<b>1</b>	0.62	0.91	<b>0.99</b>	<b>0.76</b>
<b>LDP (Pos)</b>	<b>1</b>	<b>1</b>	0.6	0.91	<b>0.99</b>	0.73



# Chapter 9

## Conclusions

The LDP descriptor is considered to be a successful feature extraction method in many texture analysis applications, especially against the applications that suffer from random noise and lighting conditions. However, this method suffers from many shortcomings. In this thesis, we have proposed DLBP, CILDP, CLDP, ALDP, and experiments conducted display very promising results. Using Circular Local Directional Pattern (CILDP), we have been able to improve on its inability to extract information from large-scale texture. Another shortcoming in LDP computation is that some information is ignored and the number of the most significant Kirsch Masks application edge response values is arbitrarily chosen as three. The DLBP, ALDP, CLDP and SLDP methods are proposed with the objective of solving the above shortcomings and to increase LDP robustness.

An empirical study in paper I was performed to investigate if the change in the number of most significant edge response values has any effect on performance. The results proved that any change in the number of most significant edge values has a positive or negative impact on performance, depending on the application.

Paper I, which showed one of the LDP limitations, the choice of the number of significant bits. Directional Local Binary Pattern (DLBP) was introduced in paper II with the aim of overcoming this shortcoming. Inspired by both LBP and LDP, DLBP uses the central pixel as a threshold to set the bits of the binary DLBP code, based on the Kirsch Masks application edge response values. Results show that DLBP outperforms both LBP and LDP.

In paper III, several extensions were proposed aiming to overcome some LDP shortcomings and increase its robustness. Centered Local Directional Pattern (CLDP) was proposed aiming to add the central pixel value to the LDP computations, which have been neglected. Also Signed Local Directional Pattern (SLDP) was introduced, which take into consideration the sign of the gradient directions (Positive and Negative), unlike LDP, which assumes the absolute value of the gradient directions. Centered Signed-LDP (CSLDP) extension merges

both ideas of CLDP and SLDP. Results establish the importance of the central pixel and the sign of the gradient directions.

Angled Local Directional Pattern (ALDP) was introduced in paper IV which computed the angle values for the gradient directions, using four orientations ( $0^\circ, 45^\circ, 90^\circ, 135^\circ$ ). This method resolved two limitations of the LDP: choosing the number of the significant bits and the overlooking of important information in the central pixel.

In paper V, Circular Local Directional Pattern (CILDP) was proposed with the aim of enabling LDP to extract good information from a large-scale texture. This method takes advantage of the circular shaped properties to compute the gradient direction using different radiuses.

The evaluation of these methods was performed using two datasets, KTH-TIPS2b and Extended Cohen-Kanade Dataset (CK+), which were discussed intensively in Chapter 8. In this comparison, five classifiers (SVM, NB, k-NN, Perceptron and DT) and seven classification evaluation techniques (learning curve, accuracy, confusion matrix, precision, sensitivity, F-score and Cohen's Kappa) were used. Results demonstrated the effectiveness of the proposed methods compared to LDP.

We analyzed the importance of the results shown in the thesis and demonstrated that the objectives set have been achieved. We can state that most of the proposed methods prove the effectiveness, especially in the application identifying facial expressions, where their performance is higher compared to previous studies. It can be observed throughout experiments that the proposed methods have improved LDP performance, but at the expense of computational cost and memory required.

Although the goals set for this thesis have been accomplished, there is room for expansion and improvement. In this thesis, we compared the proposed methods with only LBP and LDP and it would be useful to enlarge the research by adding other methods. We also mentioned previously that these extensions add burdens. In particular, there is plenty of scope to improve the running of both the LDP and the proposed improvements. One way to achieve the latter is to use parallel computing.

# References

- Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302.
- Burt, P. and Adelson, E. (1983). The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540.
- Campbell, F. and Robson, J. (1968). Application of fourier analysis to the visibility of gratings. *The Journal of physiology*, 197(3):551.
- Chellappa, R., Kashyap, R., and Manjunath, B. (1993). Model-based texture segmentation and classification. *Handbook of Pattern Recognition and Computer Vision*, pages 277–310.
- Chen, G., Liang, S., and Chen, J. (2014). The extraction of plantation with texture feature in high resolution remote sensing image. In *Earth Observation and Remote Sensing Applications (EORSA), 2014 3rd International Workshop on*, pages 384–387. IEEE.
- Clausi, D. and Jernigan, M. (1998). A fast method to determine co-occurrence texture features. *IEEE Transactions on Geoscience and Remote Sensing*, 36(1):298–300.
- Clausi, D. and Zhao, Y. (2002). Rapid extraction of image texture by co-occurrence using a hybrid data structure. *Computers & Geosciences*, 28(6):763–774.
- Connors, R. and Harlow, C. (1980). A theoretical comparison of texture algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):204–222.
- Cross, G. and Jain, A. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):25–39.
- Duda, R., Hart, P., and Stork, D. (1973). *Pattern classification*, volume 2. Wiley New York.
- Flexner, S. (1987). *The Random House Dictionary of the English Language: Unabridged*. Random House dictionaries. Random House Information Group.
- Fritz, M., Hayman, E., Caputo, B., and Eklundh, J.-O. (2004). The kth-tips database.
- Galloway, M. (1975). Texture analysis using gray level run lengths. *Computer graphics and image processing*, 4(2):172–179.
- Garg, S. and Kaur, R. (2016). Face recognition using improved local directional pattern approach for low resolution images. *International Journal of Advanced Research in Computer Science*, 7(6).

- Gelzinis, A., Verikas, A., and Bacauskiene, M. (2007). Increasing the discrimination power of the co-occurrence matrix-based features. *Pattern Recognition*, 40(9):2367–2372.
- Gotlieb, C. and Kreyszig, H. (1990). Texture descriptors based on co-occurrence matrices. *Computer Vision, Graphics, and Image Processing*, 51(1):70–86.
- Guo, Z., Zhang, L., and Zhang, D. (2010). A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663.
- Haralick, R. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804.
- Haralick, R., Shanmugam, K., and Dinstein, I. (1973). Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, SMC-3(6):610–621.
- Heikkilä, M., Pietikäinen, M., and Schmid, C. (2006). Description of interest regions with center-symmetric local binary patterns. In *Computer vision, graphics and image processing*, pages 58–69. Springer.
- Heikkilä, M., Pietikäinen, M., and Schmid, C. (2009). Description of interest regions with local binary patterns. *Pattern recognition*, 42(3):425–436.
- Jabid, T., Kabir, M., and Chae, O. (2010). Local directional pattern (ldp) for face recognition. In *Proceedings of the IEEE International Conference on Consumer Electronics*, pages 329–330.
- Jacobson, L. and Wechsler, H. (1982). The wigner distribution and its usefulness for 2-d image processing. In *In Proceedings of 6th International Joint Conference Pattern Recognition*, page 19.
- Joseph, L.J., G. L. S. G. S. I. and Santhi, N. (2016). Teeth feature extraction and matching for human identification using morphological skeleton transform. In *2016 International Conference on Energy Efficient Technologies for Sustainability (ICEETS)*, pages 802–807.
- Kabir, M., Jabid, T., and Chae, O. (2010). A local directional pattern variance (ldpv) based face descriptor for human facial expression recognition. In *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pages 526–532. IEEE.
- Kaizer, H. (1995). A quantification of textures on aerial photographs. *Boston University Research Laboratories, Technical Note 121*, AD69484.
- Kim, D., Lee, S., and Sohn, M. (2013). Face recognition via local directional pattern. *International Journal of Security and Its Applications*, 7(2):191–200.
- Laws, K. (1980). Rapid texture identification. In *24th annual technical symposium*, pages 376–381. International Society for Optics and Photonics.
- Leung, T. and Malik, J. (1996). Detecting, localizing and grouping repeated scene elements from an image. In *European Conference on Computer Vision*, pages 546–555. Springer.

- Lever, J., Krzywinski, M., and Altman, N. (2016). Points of significance: Classification evaluation. *Nature Methods*, 13(8):603–604.
- Li, B. and Zhang, J. (2015). Road extraction based on hierarchical line segment features from very high resolution remote sensing images. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 1857–1860. IEEE.
- Liao, S., Law, M., and Chung, A. (2009). Dominant local binary patterns for texture classification. *IEEE transactions on image processing*, 18(5):1107–1118.
- Liu, Y., Collins, R., and Tsin, Y. (2004). A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE transactions on pattern analysis and machine intelligence*, 26(3):354–371.
- Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. (2010). The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 94–101. IEEE.
- Luo, Y., Zhao, L., Zhang, B., Jia, W., Xue, F., Lu, J., Zhu, Y., and Xu, B. (2016). Local line directional pattern for palmprint recognition. *Pattern Recognition*, 50:26–44.
- Mao, J. and Jain, A. (1992). Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern recognition*, 25(2):173–188.
- Marr, D. (1982). Vision freeman. *San Francisco*, page 95.
- Megalooikonomou, V., Zhang, J., Kontos, D., and Bakic, P. (2007). Analysis of texture patterns in medical images with an application to breast imaging. In *Medical Imaging*, pages 651421–651421. International Society for Optics and Photonics.
- Miles, K., Ganeshan, B., Griffiths, M., Young, R., and Chatwin, C. (2009). Colorectal cancer: texture analysis of portal phase hepatic ct images as a potential marker of survival 1. *Radiology*, 250(2):444–452.
- Nanni, L., Lumini, A., and Brahnam, S. (2010). Local binary patterns variants as texture descriptors for medical image analysis. *Artificial intelligence in medicine*, 49(2):117–125.
- Ojala, T., Pietikäinen, M., and 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., and 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.
- Ojala, T. and Pietikäinen, M. (2004). Texture classification. *Machine Vision and Media Processing Unit*.
- Pan, W., Li, Z., Li, H., Yin, L., and Gong, B. (2013). Texture feature of remote sensing image for the recognition of hydrothermal uranium ore-field in south china. In *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, pages 3026–3028. IEEE.

- Park, M., Brocklehurst, K., Collins, R., and Liu, Y. (2009). Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(10):1804–1816.
- Phadikar, S. and Goswami, J. (2016). Vegetation indices based segmentation for automatic classification of brown spot and blast diseases of rice. In *Recent Advances in Information Technology (RAIT), 2016 3rd International Conference on*, pages 284–289. IEEE.
- Pontius Jr, R. G. and Millones, M. (2011). Death to kappa: birth of quantity disagreement and allocation disagreement for accuracy assessment. *International Journal of Remote Sensing*, 32(15):4407–4429.
- Richards, W. and Polit, A. (1974). Texture matching. *Kybernetik*, 16(3):155–162.
- Rosenfeld, A. and Kak, A. (1976). Digital picture processing.
- Ryu, B., Rivera, A. R., Kim, J., and Chae, O. (2017). Local directional ternary pattern for facial expression recognition. *IEEE Transactions on Image Processing*.
- Shabat, A. and Tapamo, J. (2014). A comparative study of local directional pattern for texture classification. In *Computer Applications & Research (WSCAR), 2014 World Symposium on*, pages 1–7. IEEE.
- Shabat, A. and Tapamo, J. (2016). Directional local binary pattern for texture analysis. In *International Conference Image Analysis and Recognition*, pages 226–233. Springer.
- Sivapalan, S., Chen, D., Denman, S., Sridharan, S., and Fookes, C. (2013). Histogram of weighted local directions for gait recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 125–130.
- Sklansky, J. (1978). Image segmentation and feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(4):237–247.
- Song, T., Meng, F., Luo, B., and Huang, C. (2013). Robust texture representation by using binary code ensemble. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE.
- Subrahmanyam, M., Wu, Q., Maheshwari, R., and Balasubramanian, R. (2013). Modified color motif co-occurrence matrix for image indexing and retrieval. *Computers & Electrical Engineering*, 39(3):762–774.
- Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(6):460–473.
- Tan, X. and Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, 19(6):1635–1650.
- Tomita, F. and Tsuji, S. (1990). *Computer Analysis of Visual Textures*. The Springer International Series in Engineering and Computer Science 102. Springer US, 1 edition.
- Tuceryan, M. and Jain, A. (1990). Texture segmentation using voronoi polygons. *IEEE transactions on pattern analysis and machine intelligence*, 12(2):211–216.

- Umarani, C., Ganesan, L., and Radhakrishnan, S. (2008). Combined statistical and structural approach for unsupervised texture classification. *International Journal of Imaging and Engineering*, 2(1):162–165.
- Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision*, 62(1-2):61–81.
- Varma, M. and 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.
- Voorhees, H. and Poggio, T. (1988). Computing texture boundaries from images. *Nature*, 333(6171):364–367.
- Weszka, J., Dyer, C., and Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 4(SMC-6):269–285.
- Yalniz, I. and Aksoy, S. (2010). Unsupervised detection and localization of structural textures using projection profiles. *Pattern Recognition*, 43(10):3324–3337.
- Zhang, L., Li, A., Zhang, Z., and Yang, K. (2016). Global and local saliency analysis for the extraction of residential areas in high-spatial-resolution remote sensing image. *IEEE Transactions on Geoscience and Remote Sensing*, 54(7):3750–3763.
- Zhou, H., Wang, R., and Wang, C. (2008). A novel extended local-binary-pattern operator for texture analysis. *Information Sciences*, 178(22):4314–4325.

