IMPROVED TEXTURE FEATURE EXTRACTION AND SELECTION METHODS FOR IMAGE CLASSIFICATION APPLICATIONS

ALSADEGH MOHAMED

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computing and Engineering

University of Huddersfield

September 2018

Abstract

Classification is an important process in image processing applications, and image texture is the preferable source of information in images classification, especially in the context of real-world applications. However, the output of a typical texture feature descriptor often does not represent a wide range of different texture characteristics. Many research studies have contributed different descriptors to improve the extraction of features from texture. Among the various descriptors, the Local Binary Patterns (LBP) descriptor produces powerful information from texture by simple comparison between a central pixel and its neighbour pixels. In addition, to obtain sufficient information from texture, many research studies have proposed solutions based on combining complementary features together. Although feature-level fusion produces satisfactory results for certain applications, it suffers from an inherent and well-known problem called "the curse of dimensionality''. Feature selection deals with this problem effectively by reducing the feature dimensions and selecting only the relevant features. However, large feature spaces often make the process of seeking optimum features complicated.

This research introduces improved feature extraction methods by adopting a new approach based on new texture descriptors called Local Zone Binary Patterns (LZBP) and Local Multiple Patterns (LMP), which are both based on the LBP descriptor. The produced feature descriptors are combined with other complementary features to yield a unified vector. Furthermore, the combined features are processed by a new hybrid selection approach based on the Artificial Bee Colony and Neighbourhood Rough Set (ABC-NRS) to efficiently reduce the dimensionality of the resulting features from the feature fusion stage.

Comprehensive experimental testing and evaluation is carried out for different components of the proposed approach, and the novelty and limitation of the proposed approach have been demonstrated. The results of the evaluation prove the ability of the LZBP and LMP texture descriptors in improving feature extraction compared to the conventional LBP descriptor. In addition, the use of the hybrid ABC-NRS selection method on the proposed combined features is shown to improve the classification performance while achieving the shortest feature length. The overall proposed approach is demonstrated to provide improved texture-based image classification performance compared to previous methods using benchmarks based on outdoor scene images.

These research contributions thus represent significant advances in the field of texture-based image classification.

Acknowledgements

All praise be to **Allah**, the Almighty, who has helped me complete this work.

The guidance of my supervisor Prof Joan Lu is acknowledged throughout my research.

I would like convey my gratitude for my family; late father and special thanks are due my precious mother for her supplications to Allah for me to succeed. I am also grateful to my brothers and sisters for their moral support through the years of my study.

List of Publications

Mohamed, A. & Lu, J. (2015). Analysis of GLCM Parameters for Textures Classification on UMD Database Images, INFOCOMP 2015: The Fifth International Conference on Advanced Communications and Computation, IARIA pp. 111-116.

Mohamed, A., & Lu, J. & Qiang Xu. (2016). Multi-texture classification using optimized Gabor Filter by Artificial Bee Colony, International Conference on Change, Innovation, Informatics and Disruptive Technology ICCIIDT'16, London- U.K, pp 139-144

Mohamed, A., & Lu, J. Qiang Xu. & Jinwen Ma. (2018). Optimizing Gabor Filter and Local Binary Patterns for multi-texture classification, Egyptian Computer Science Journal Vol. 42 No.2, pp 55-67.

List of Abbreviations

LBP	Local Binary Pattern
GF	Gabor Filter
ABC	Artificial Bee Colony
RS	Rough Set
NRS	Neighbourhood Rough Set
LZBP	Local Zone Binary Pattern
LMP	Local Multiple Pattern
GLCM	The Grey level Co-occurrence Matrix
GLDM	Grey Level Difference Matrix
GLRL	Grey Level Run Length
ACF	Autocorrelation Function
GMRF	Gaussian Markov Random Field
MRFs	Markov Random Fields
SAR	Synthetic Aperture Radar
TUs	Texture Units
CP	Central Pixel
TS	Texture Spectrum
LTP	Local Ternary Pattern
LQP	Quinary Pattern
FT	Fourier Transform
STFT	Short Time of Fourier Transform
HVS	Human Visual System
MSMD	Multi Scales Multi Directions
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
CLBP	Completed Local Binary Pattern
DLBP	Dominant Local Binary Pattern

HOG	Histograms of Oriented Gradient
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
PLS	Partial Least Square
BPNN	Back Propagation Neural Network
SVM	Support Vector Machine
KNN	K Nearest Neighbour
NB	Naïve Bayes
EA	Evolutionary Algorithm
SI	Swarm Intelligence
ACO	Ant Colony Optimisation
DE	Differential Evolution
CM	Confusion Matrix
CV	Computer Vision

List of Tables

List of Figures

Table of Contents

Chapter 1

Introduction

1.1 Overview and Motivation

Image classification is one of the most important topics in the computer vision field. Classification of images can be applied based on features corresponding to colours, shapes, or textures. However, texture-based image classification is more valuable in real-world applications and for processing natural images (Turtinen & Pietikäinen, 2006). For instance, in a natural or outdoor scene such as a jungle, it is difficult to distinguish a particular creature (e.g. a tiger) that is hiding behind a tree; or discriminate between different creatures (e.g. tigers and leopards) based on their colours or shapes. In addition, colour-based classification has several shortcomings, such as its inapplicability with infrared cameras images and night vision settings, as well as its sensitivity to varying illumination conditions (Castano, Manduchi, & Fox, 2001). The shape property is only appropriate when the scene contains regular objects, which represents a major limitation as most outdoor scenes of real-world applications contain shapes of a random nature. In the aforementioned situations, a texture-based approach can be more robust than other image features when used in the context of image classification.

For decades, texture-based features have been receiving considerable attention in image classification applications (Majid & Xianghua, 2008). The challenge of depending on texture for classification is that the texture in natural images is mostly random with a large number of variations in the visual appearance. Fig 1.1 shows samples from a wide variety of texture patterns that range between regular and stochastic. However, although a textured image usually involves complicated characteristics, most of these characteristics can be categorised as coarseness, contrast, or directionality of repetitive patterns, where such properties are required to identify most textures.

There are many problems that can disturb the texture of the image, such as changes in scales or orientations, non-uniform illumination, or the noise and blur in the image. These can be caused by many reasons, such as changes in the lighting conditions which results different illumination settings, or changes in camera position which results in different orientations, viewpoints or scales, or lack of focus which results in blur. These problems increase the difficulty of finding appropriate and distinctive texture features, which in turn degrades the performance of image classification systems.

Fig. 1. 1 Textures arranged according to the regularity of their structural variations (texture spectrum).

Figure 1.2 shows a typical classification process, which consists of a feature extraction stage and a classification stage (Di Cataldo & Ficarra, 2017). Features are obtained from texture samples using methods called feature extractors or descriptors, which convert these samples into features. In classification, the samples of images are categorised according to their features, which is done by introducing the features to a learning algorithm or classifier. The ability to classify different types of textures is the aim of classification systems. Good quality features and powerful classifiers should therefore be available to make that possible. The descriptors' aim is to extract powerful features that are distinctive enough to discriminate between different textures. Indeed, in texture classification, most researches pay significant attention to extracting powerful features in order to improve the overall classification results (L. Liu, Long, Fieguth, Lao, & Zhao, 2014). This research focuses on improving the feature extraction stage, as without investigating and developing suitable methods to produce discriminative features from texture, the outcome of classification is often unsatisfactory.

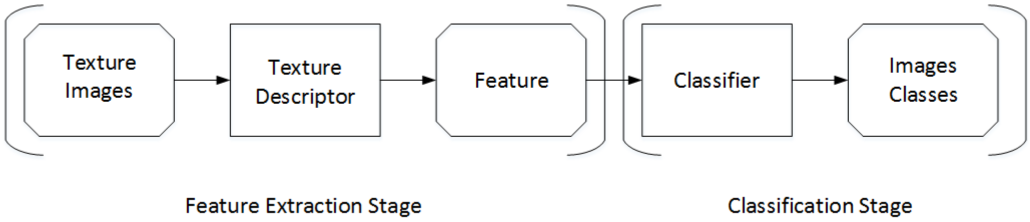


Fig. 1. 2 Texture-based image classification system.

Numerous feature extraction methods have been introduced based on texture (Majid & Xianghua, 2008). The Local Binary Patterns (LBP) method has emerged as one of the most effective descriptors in texture classification (Ojala, Pietikäinen, & Harwood, 1996). This method has shown outstanding performance with exceptional speed and powerful discriminative features. Furthermore, LBP has been utilised in many real-world texture analysis applications (Mäenpää & Pietikäinen, 2005; Ojala, Pietikainen, & Maenpaa, 2002). Motivated by the local binary patterns concept, this research proposes new texture descriptors. These descriptors aim to extract improved features from texture patterns in the context of image classification systems.

Identifying different textures is the ultimate goal of texture description systems. Extracting specific features from a single descriptor is usually inadequate in recognising different types of texture (Bashar & Ohnishi, 2002). Several limitations of texture applications can be overcome by integrating multiple features. The LBP descriptor has been utilised as a complementary tool with other feature extraction methods. LBP joined with local extension methods based on the LBP approach can provide improved features compared to using LBP separately (Ojala, Pietikainen, et al., 2002). Since LBP is considered a micro-texton, its features are used as complementary features with macro-texton of Gabor Filter (GF) (Liao, Law, & Chung, 2009). In a several studies, Gabor filter have shown excellent performance and have often outperformed other texture analysis methods. The main problem of Gabor filters is the fact that they are very computationally demanding (Randen & Husoy, 1999).

Integrated features result from concatenation between the features of the participating descriptors. The extracted features by different descriptors often have high feature dimensions, which makes integrating between these features result in a large feature space. In addition, part of features from this fusion process happens to be irrelevant, which can therefore have a negative effect and reduce the quality of the overall extracted features (Zhao, Sinha, & Ge, 2009). In general, depending on a large features size degrades the performance of image classification, where this problem is referred to as ‘the curse of dimensionality’ (Gheyas & Smith, 2010). To address this, a feature selection step is essential for reducing the features space and improving the quality of the extracted feature at the same time. However, effective feature selection that yield optimal set of features is not a trivial task, since the search space grows exponentially with the features length, which can make feature selection infeasible in some cases (A. L. Blum & Langley, 1997).

To select optimal or near-optimal features from the original feature space, a number of feature selection methods can be utilised, which are typically categorised into either wrapper methods or filter methods (Dash & Liu, 1997; Kohavi & John, 1997). Swarm intelligence algorithms are a type of wrapper methods, where the Artificial Bee Colony (ABC) algorithm is an example of swarm algorithms. ABC has recently gained more attention as modern feature selection method and has been proposed for many feature selection problems (Karaboga & Basturk, 2008). In contrast, the Rough Set (RS) is one of the filter methods, which has also been successfully applied for reducing features dimensionality (Pawlak, 2012). Based on a number of research studies, filter approaches have been shown to be computationally effective with limited feature numbers, whereas wrappers have usually outperform filters methods in terms of classification performance (A. L. Blum & Langley, 1997; Hoque, Bhattacharyya, & Kalita, 2014).

Finding an effective feature selection approach is essential in this research. Recently, some authors have proposed hybrid approaches to exploit advantages of both techniques (i.e. wrapper and filter) and yield an effective feature selection method (Gheyas & Smith, 2010). This approach has been adopted in this research by proposing and developing hybrid selection tool based on the ABC and Rough Set (RS) methods to process the features resulting from the proposed feature fusion stage.

1.2 Problem Statement

From the previous section, it is clear that the description of different texture characteristics is a difficult task, since textures exist in a wide variability and complexity. This research focuses on improving texture classification by utilising powerful and distinctive features. This section summarises the main problem statement of this research, which includes the following three points:

1. The most significant challenge in texture classification is the extraction of powerful features (L. Liu et al., 2014). LBP is one of the most effective texture descriptors due to

its simplicity and good performance, where it relies on the relationship of neighbouring pixels with the centre pixel (Ojala, Pietikainen, & Harwood, 1994; Ojala et al., 1996). However, texture descriptors like LBP do not efficiently exploit rich information from texture patterns, which reflects on the accuracy of texture classification. The challenge is to improve discriminative feature extraction power of the LBP method by efficiently exploiting the intensity values of texture patterns.

2. Texture surfaces contain a wide variety of characteristics, which is what makes extracting suitable texture features for images classification applications challenging (W.-C. Lin, Hays, Wu, Kwatra, & Liu, 2004). It is difficult, if not impossible, to develop one single method that has the ability to extract optimum information from different texture characteristics (Bashar & Ohnishi, 2002; Turtinen, 2007). In order to obtain more distinctive and powerful features from texture and thus improve the performance of image classification, one can combine features from different descriptors (Clausi & Deng, 2005; Ojala et al., 1996). However, the main problem with improving feature extraction through this fusion strategy is the resulting large feature space, which usually negatively affects the classification performance.

3. In general, the highest possible classification accuracy comes from longest feature length because there is a rich set of information collected from the image. However, combining features from different methods often results in irrelevant or redundant features that mostly have a negative effect on the classification performance. Irrelevant features not only increase the computation cost of the classifier, but also harm the relevant features and lead to a negative effect on the quality of the complete feature set, and in turn a reduced classification accuracy (Zhao et al., 2009). This problem, which results from combining a large number of features by different descriptors is referred to as "the curse of dimensionality" in image classification (Clausi & Deng, 2005). As such, to increase the quality of the features resulting from combining different feature descriptors, one must select only the relevant features from the complete feature set (Kohavi & John, 1997). The resulting features from the selection stage have a shorter feature length, which leads to improving the accuracy of the classifier and its processing time (Gheyas & Smith, 2010). However, feature selection is usually a challenging task, and many methods lack an effective strategy for selecting optimum features.

1.3 Research Questions

Following the declaration of the problem statement in the previous section, the focus of this thesis will now be dedicated to how to improve texture classification by extracting distinctive and powerful features efficiently. The proposed approach to achieve this aim is to integrate the developed discriminative features with other complementary features and efficiently select the relevant features from the complete feature set. The following research questions, associated with this strategy, will be investigated:

1. How can the features of the LBP method be improved by utilising intensity values of texture patterns?
2. How can the improved features obtained from the enhancement of the LBP method be combined with other suitable features to produce powerful features without the negative impact of a large feature space?
3. Can the feature space of the combined features be reduced effectively by depending on only the relevant features using a well-organised feature selection method that result an improved classification performance?

1.4 Research Aim and Objectives

The research questions explain the main purpose of this research, which is to develop distinctive texture-based features for image classification applications. This will be done by extracting the discriminative features from the texture in the image, followed by combining those features with other complementary features, and selecting only the relevant features from original feature set.

In particular, this approach depends on two main stages, which are: (i) extracting features from the texture by the new improved descriptors, which will then be combined with other complementary features, and (ii) selecting only the relevant features to avoid the curse of dimensionality. The goal of this approach is to increase the accuracy of images classification by extracting a diverse set of features from texture, then using the proposed selection method, dispose of irrelevant features, which have a negative effect on the classification performance. Based on this aim, the objectives of this research can be defined as follows:

1. To develop novel texture descriptors that are motivated by the LBP descriptor, where the power of LBP in texture classification will be harnessed for developing descriptors that have the ability to extract more distinctive and powerful features.
2. To implement an efficient approach for texture-based images classification through integrating a diverse set of features. These feature will be based on combining the features extracted by the newly developed descriptors with others complementary features to improve the classification performance.
3. To develop and implement a new hybrid feature selection method. This method will be based on the wrapper method using the ABC algorithm and the filter method using NRS, and will target the selection of only the relevant features from involved combined descriptors in order to avoid the curse of dimensionality and reduce the classifier's processing time.

1.5 Thesis Contributions

This thesis presents novel feature descriptors that are capable of extracting distinctive features from different texture characteristics in image classification applications. The research methodology is based on utilising the local binary pattern method for developing new feature descriptors, then using a new selection approach based on the ABC and NRS methods to process the features resulting from the suggested feature fusion process.

1. **Developing novel texture descriptors, called LZBP and LMP, for extracting local features**

   These texture descriptors have been developed based on addressing the limitation of the LBP descriptor. LBP does not always extract the important local feature from texture by binary thresholding for different intensity values of pixels. This means that LBP may fail to classify different texture characteristics. The developed descriptors have the ability to extract richer local features than LBP, where LZBP and LMP are superior in considering the intensity values of pixel in TUs through different quantization zones. The new descriptors have been tested using different benchmarks, which included a number of textures databases, and compared with well-known feature extraction methods. The reported results from several evaluation experiments demonstrate the suitability and competitiveness of the new descriptors to other feature extraction methods.

2. **Developing a hybrid feature selection method based on ABC and NRS to deal effectively with the high dimensionality of features**

   There are few research studies on using hybrid wrapper approaches based on filter feature selection. Many previous studies are based on using the wrapper approach exclusively, which is expensive in terms of computation when evaluating potential optimum features. In our work, hybridization of ABC as a wrapper approach with the NRS filter method is introduced to process the proposed combined feature set. The process of using the hybrid method on the proposed feature fusion involves utilising the NRS filter method to produce a pool of selected features, then using the ABC algorithm as a wrapper method to find the optimal part of features from this pool of selected features. In assessment results of the overall classification system performance, it was found that the proposed hybrid selection method can provide a good balance between accuracy and computation cost.

3. **Implementing feature fusion between the LZBP and LMP descriptors and the contrast of the texture image**

   In evaluation results of the LZBP and LMP descriptors, an improved classification performance was achieved. However, these descriptors were more effective with specific databases than others. Supporting LZBP and LMP by the contrast of the texture image in a feature-level fusion stage, applied with a multi-scale analysis, yielded better results for a diverse set of texture characteristics. The resulting large feature space was reduced to an acceptable feature length using the proposed hybrid ABC-NRS selection approach. The new hybrid selection approach was applied on LBP, LZBP and LMP after combining these descriptors with contrast measure as complementary features. The experimental results prove the ability and effectiveness of the Rough Set method to reduce the feature length of the histogram resulting from the multi-scale LBP, LZBP and LMP descriptors.

4. **I Implementing feature fusion between the local features of LZBP and LMP and the global feature of GF**

   In previous studies, integrating the features of LBP and GF produced better result than if either of them was applied separately. In this work, GF and multiscale LBP were integrated using the new hybrid feature selection method to select optimum feature with the least computational cost. ABC was applied on GF to select the relevant features by selecting the optimal filters, whereas NRS was used to achieve feature reduction of the multiscale LBP.

The new LZBP and LMP descriptors were also combined with complementary GF features. The results of the integrating these local descriptors with GF as global descriptor using hybrid selection methods improved both the classification performance and feature length. In addition, LZBP and LMP were also found to be suitable as complementary features to GF. The results from experiments demonstrate the suitability of the ABC algorithm for selecting the optimum feature resulting from a set of Gabor filters.

1.6 Thesis Layout

This thesis is divided into seven chapters, followed by two appendices.

Chapter 1 presents an introduction of this research, which includes an overview of the topic and motivation, the problem statement, the main aim of the research and the major objectives, and original contributions of the research.

Chapter 2 is the literature review chapter, where a general review of texture-based image classification is provided, including a discussion of texture characteristics, texture analysis methods, and texture tasks. The main studies related to texture-based image classification is then provided, including an overview of the most common texture descriptors, feature fusion approaches, and feature selection methods.

Chapter 3 discusses the research methodology, including the main contributions of the research, which focuses on developing new improved texture-based feature extraction and selection methods for images classification applications. The chapter includes a description of the new descriptors that are used to extract features from texture, as well as description of the new hybrid selection approach.

Chapter 4 is the design and implementation chapter, which begins by discussing the implementation of the proposed methods to improve feature extraction and selection. This includes a discussion of the process used by the new feature extraction methods and the new feature selection algorithms.

Chapter 5 presents the results of the experiments investigating the validity of the proposed feature descriptors, feature fusion, and selection method using two different models. This is achieved by conducting an extensive set of experiments on texture-based image classification.

Chapter 6 is the evaluation chapter, which presents the evaluation of the results of the new feature extraction methods and the proposed feature selection strategies targeting the feature fusion stage. The evaluation is done through comparing the proposed methods with other classical methods in order to measure and analyse the level of performance achieved by the proposed approach.

Chapter 7 presents a conclusion of this thesis, where a summary of the contributions of thesis, the achievements of this research, and suggestions for possible future work directions is given.

Appendix A contains a comparison of the Confusion Matrix results of the proposed descriptors with other common texture descriptors.

Appendix B contains a comparison of the Confusion Matrix results of proposed feature fusion approach and other common texture descriptors.

Chapter 2

Literature Review

In images processing, texture-based images classification is an active research topic that is still being developed by researchers. Texture is an important source that provides useful information about the image. The importance of texture properties appears clearly in natural images, where they are used for classification between images containing sky, sea, leaves and grass, which are hard to classify by other sources of information.

Textures classification depends on the extracted features. However, extracting powerful features based on texture is a real challenge due to the wide diversity of texture characteristics. There are different descriptors or analysis methods of texture that have been developed to deal with different texture characteristics. This chapter starts by introducing the concept of texture definition and characteristics, after which a literature-based introduction of common and widely used descriptors is provided. A significant attention is paid to the LBP descriptor in this research, in order to utilise it to develop new texture descriptors.

Feature fusion is also important step in order to improve features when dealing with different texture characteristics. However, this step results in a common problem on a feature level, called the ‘‘curse of dimensionality’’. Therefore, adopting feature fusion by an effective features selection method for relevant features is required to deal with this problem. This thus acts as a motivation to provide a detailed literature review on the available feature selection approaches.

2.1 Texture Overview

2.1.1 Texture Definition

There is no agreement on a specific definition on texture, although it is possible to recognise textures visually or by touching (Fan & Xia, 2003). Nevertheless, there have been many efforts to define texture, where Haralick and Shanmugam (1973) is considered to be the first researcher who recognised the importance of finding a definition of texture. Although there are numerous definitions of texture, which are either mentioned in the dictionaries or other sources, it is still currently difficult to define a specific formal definition of texture (Karu, Jain, & Bolle, 1996).

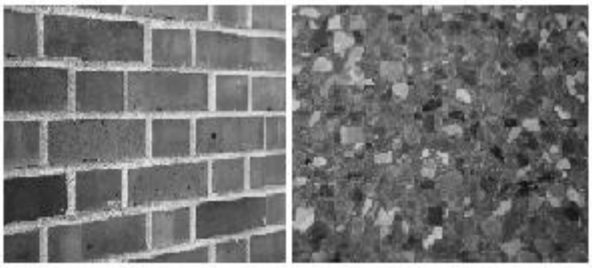
Texture is recognised by a local-neighbourhood property, as colour is recognized by a point property (Belongie, Carson, Greenspan, & Malik, 1998). Texture is a set of elements or primitives that may appear clearly, or sometimes hardly, on a surface of any substance. These primitives consist of a set of pixels with internal properties such as intensity values, which refer to tone. The relationship between these pixels of intensity values under conditions of the same tone constitute the structure of the texture.

Tiny sized elements such as grains of sand, or large scale elements such as a group of stars in the sky, may reflect the surface of the texture. However, the elements in the texture should be large enough not to resemble noise, and should not be too large to be seen as objects by themselves (Karu et al., 1996). In the universe, texture may appear in many things if viewed from a specific distance, as the scale or resolution is important for perception of texture. In (Chaudhuri, Sarkar, & Kundu, 1993), the authors state that an essential characteristic of texture is that "texture regions give different interpretations at different distances, and at different degrees of visual attention". Fig 2.1 (a) from near distance appears to be single star, which is not a texture, whereas if the viewing distance is increased, as in Fig 2.1 (b), the picture appears to a group of stars, which might show texture.

(a) (b)

Fig. 2. 1 Same primitive with different scales appearing in two images. (a) An image categorised by object shape. (b) An image categorised by texture.

The spatial variation of the textured surface is important for recognising and classifying the different textures. Special properties of texture can be recognised in our minds at the time we touch or see the picture. The perception of the person depends on the variations of pixels in a texture, which are used to discriminate between surface types in the image. One rather simple method for categorising the textures, is to divide them into stochastic or deterministic texture surface types, as demonstrated in the Fig 2.2, where in Fig 2.2(a), the first elements are regular with strictly deterministic nature, and mostly exist in synthetic texture, whereas in in Fig 2.2(b), the second set of elements are irregular or uncorrelated, and mostly found in natural texture.



(a) (b)

Fig. 2. 2 A texture with (a) clear elements (regular), (b) unclear elements (irregular).

Other researchers have used different categorisation for describing texture, which provides more details on its special properties. According to (Haralick & Shanmugam, 1973), texture is grouped into fine, coarse, or smooth; or rippled, molled, irregular, or lineated categories. In (Tamura, Mori, & Yamawaki, 1978), texture is divided and extended into groups of coarseness, contrast, directionality, line-likeness and regularity properties.

Coarse versus fine: When elements of the texture are far away in space and large in size, the texture is coarse, whereas, in contrast, a fine texture in one where elements of the texture are close in space and tiny in size.

High contrast versus low contrast: Contrast defines the variety in pixel value scale in the image between pixels with high values and those with low values. Depending on the structure of the texture, contrast is affected by numerous factors such as edges, sharpness and periods of repeating patterns. For example, if the image has sharp edges, it is considered to be of higher contrast.

Directional versus non-directional: This categorisation includes primitive shapes and the placement rule.

Line-like versus blob-like: This categorisation involves only primitive shapes of a texture, where a texture may take a line shape or a blob one.

Regular versus irregular: If there are no variations of elements in the texture by the placement rule, then the texture would be described as regular, whereas, big variations of elements in the placement rule would result in the texture being observed as irregular.

In a recent study by Rao and Lohse (1993), additional important classes were added, which are more understood by the perception of texture by humans. The groups included the classes and their opposites, such as repetitiveness versus irregularity, directional versus non-directional, and structurally complex versus simple. Laws (1980) defined other texture properties which play an important role in classifying textures, such as uniformity, density, coarseness, roughness, regularity, linearity, directionality, frequency and phase. From this wide range of categorisation of texture characteristics, without doubt, it is difficult for one method to be adequate for texture representation.

2.1.2 Texture Analysis Approaches

A wide variety of texture analysis methods have been invented to deal with different texture properties. These methods, which are used for extracting features from texture, have been divided into groups. Sonka, Hlavac, and Boyle (2014), mentioned two main approaches to texture analysis, which are the structural approach and the statistical approach. The structural methods are used to extract features from regular textures. Statistical methods are more appropriate to deal with irregular (random) textures. M Tuceryan and Jain (1998) introduced an extension to the texture analysis approaches, which included statistical, structure, model-based, and signal processing approaches.

2.1.2.1 Structural Approach

Structural methods based on geometric properties such as size and shape of primitives, or elements of texture, are called texels or texton (Karu et al., 1996; Zhu, Guo, Wu, & Wang, 2002). Here, texels are the smallest elements in the image, where together they reflect the impression about the region of texture. Structural methods consider some attributes that are constructed from the spatial distribution of primitives, such as repetition of the texture surface. They apply the placement rule for the spatial relationship of primitives (Haralick, 1979). For example, if the image was a brick wall, then the primitive in the texture is the brick, whereas the placement rule is performed on the bricks to detect their arrangement in space. Structural methods classify textures into classes, or segments of one image into different regions, based only on the good structure of primitives in the image (Murray, Lucieer, & Williams, 2010).

Among the methods based on the good structure of the image is that by (Huet & Mattioli, 1996), which applies morphology operations for determining the elements that constitute the texture, and that by (Mihran Tuceryan & Jain, 1990), which uses Voronoi tessellation to describe the shape properties of the texture elements.

2.1.2.2 Statistical Approach

Statistical methods are well known within the computer vision field, beginning from the earliest methods used to analyse grey texture images (M Tuceryan & Jain, 1998). Statistical methods replace placement rules for describing the texture with a general set of statistical tools, which are based on spatial distribution of the intensity values. Based on existing correlation between pixels in the spatial domain of complete or specific regions of the image, the methods are divided into first-order statistical methods and second-order statistical methods.

First-order statistical methods extract features from textures through the effectiveness of pixels as individual values, without considering the sets around the pixels. The grey-level histogram is one of these methods, which is based on calculating the intensity value of complete images (William H Nailon, McLaughlin, Spencer, & Ramo, 1996). Texture features derived from this method include the mean, standard deviation, variance, and average energy, which are mostly not sufficient to discriminate between different textures. These methods are statistically invariant to any rearrangement of pixels in the image (William Henry Nailon, 2010).

Second-order statistical methods consider more meaningful features from textures by utilising the intensity of the pixels and the interaction with the neighbouring pixels. They depend on measuring the distribution of grey-level values of pixels relative to each other in the spatial domain, which make such methods suitable for achieving higher discrimination of features, especially within random textures. There is supposed to be a correlation between pair of pixels in the grey-level of an image at random distances and orientation, and such correlation depends on the power of the method used (William Henry Nailon, 2010).

2.1.2.3 Signal Processing Approach

Signal processing is a modern approach in dealing with texture, which usually extracts the features by applying a set of filters to record the responses from the image. It utilises responses of these filters to create texture features by a multi-scale analysis, relying on multiple spatial filters with different frequency characteristics to perform the analysis (Majid & Xianghua, 2008). The benefit of the multi-scale methods is that they are capable of reaching the appropriate scales for different textures. Scale is the main challenge with texture analysis, and it is important to define the characteristics of texture with multi-resolution. The filter method processes the scale problem in the image by zooming into appropriate scales using different values of frequency.

Extracted features in filtering methods can be carried out in the spatial, frequency, or the spatial/spatial-frequency domain. Spatial domain methods, such as Laws masks, usually use the convolution operation between certain masks and the image (Laws, 1980). These masks directly extract properties of texture features such as edges, lines, and isolated dots (Xie, 2008). In the frequency domain, the process starts by transforming the image into the Fourier domain, before convolving it with a filter function. Joint spatial/spatial-frequency methods localise defective regions in the spatial domain by a window function. Fourier coefficients cannot localise specific regions in the image unless a window function is used. If the window function is Gaussian, it results in the Gabor transform, which is suitable in spatial and frequency domains localisation (Xie, 2008). Although, the Fourier spectrum was the first method to be developed and has been extensively applied with texture (William Henry Nailon, 2010), Gabor outperformed others in analysing texture (Majid & Xianghua, 2008). Gabor has thus emerged as one of the most popular filter methods to characterise texture using multi-scale channels with different frequencies to analyse different scales (Manjunath & Ma, 1996).

2.1.2.4 Model-based Approach

A model-based approach is essential for describing texture. Texture is described using a parametric approach, where these parameters are used as features for a specific texture analysis. There are different parametric methods used with texture. Random models such as the Gaussian Markov Random Fields (GMRF) or the Markov Random Fields (MRFs) models are mostly used for capturing information from texture. There are also other methods such as the fractal models, autoregressive models, and random field models (Majid & Xianghua, 2008). However, model-based methods are more suitable for synthesised textures, as their achievement is limited with random images of real applications (Xie, 2008).

2.1.3 Texture Applications

Extracted features from textures by any texture analysis algorithm is the pre-processing step for any application of computer vision, like texture segmentation and texture classification.

2.1.3.1 Texture Segmentation

The segmentation of texture involves partitioning an image that contains differences in texture characteristics into a number of regions as show in Fig 2.3. In texture segmentation, there is no previous knowledge about the property of texture regions that constitute the image. Segmentation happens if there are two or more different textures in the image, and usually the regions of the different textures are adjacent. There are two techniques to apply segmentation on the image based on the texture. One technique is based on the texture region, while the other is based on the boundary (A. Song & Ciesielski, 2003). A region-based approach determines the uniform regions of texture, which are the local regions consisting of a set of pixels with the same texture characteristic in the image. In this method, if the adjacent regions of texture are close to each other, the regions will be segmented carefully. The main problem with this method is that the local properties of the image that intended to be split into sub-regions usually depends on the homogeneousness of the texture regions (Y. Deng & Manjunath, 2001).

The boundary-based approach looks for changes in texture properties from any of neighbouring regions, which determines the border of the texture region. As a result, this method does not pay attention to the number of regions in the image. The boundary-based approach faces a problem if there is space between two regions, and such space was not determined as an adjacent region too (Sklansky, 1978; A. Song & Ciesielski, 2003).

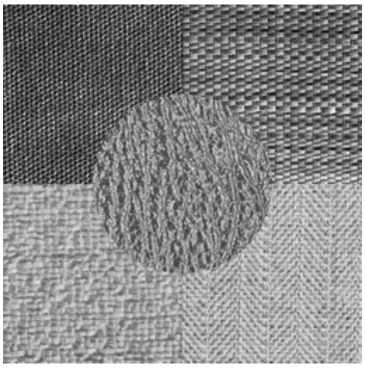


Fig. 2. 3 Sample image of five texture regions for segmentation.

2.1.3.2 Texture Classification

In the classification task, a group of unknown images are classified into a number of predefined classes, where each image in the group belongs to one of classes (see Fig 2.4) (Duda, Hart, & Stork, 2012). Fig 2.4 shows five classes, and a sample image that is supposed to belong to class 4. In classification, the images are supposed to be in the most appropriate class, and in the best classification results, every image is classified into the correct class. However, perfect classification is a difficult task.

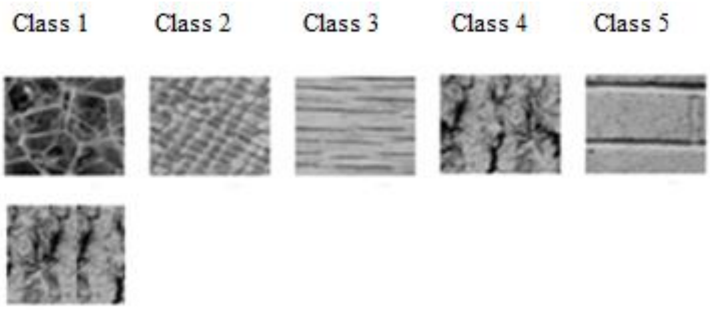


Fig. 2. 4 For a texture classification system, the sample on the bottom left needs to be classified to one of the five classes at the top.

In this work, significant attention is paid to textures classification, as texture has been utilised in several applications of image classification. Haralick and Shanmugam (1973) analysed textural features from different sources of images to classify a terrain of land. The classification was based on number of decision rules, and its ability depended on the different source images. Dell'Acqua and Gamba (2003) used SAR images to apply texture analysis by co-occurrence methods for determining the construction inside a city. The classification of buildings was based on some measurements like the length of windows. Siew, Hodgson, and Wood (1988) employed a number of matrices for texture features to classify different carpets. The matrices were the neighbouring Grey-level Dependence Matrix, the Grey-level Difference Method, and the Grey-level Run Length Method, where these matrices had the ability to discriminate between different textures in the carpets. For classification of wood texture images, Khalid, Yusof, and Meriaudeau (2010) applied a number of texture features methods, namely: Gary-Level Co-occurrence Matrix, Local Binary Patterns, Wavelet, Ranklet, Granulometry, and Laws' Masks. He also reported superior results by the Local Binary Patterns methods over the other methods.

In classification, the group of images to be classified is presented with feature vectors. These feature vectors result from one or more texture extractors, which describe the characteristics of images. The feature vectors are introduced into a classifier or a predictor, such as the Back Propagation Neural Network (BPNN) or the Support-Vector Machine (SVM), to be trained as inputs and determine an output, which predicts or classify the unknown images.

(1) Back Propagation Neural Network

A neural network is a learning method that is widely used in images processing applications (Park, Lee, & Kim, 2004; Tou, Tay, & Lau, 2009). Here, a simple explanation is given about a neural network, which is categorised as a common supervised machine. Fig 2.5 depicts a sketch of a neural network, which consists of a number of layers; an input layer, single or many hidden layers, and an output layer. The input layer is connected to the hidden layers by a set of weights, and the hidden layers are connected to the output by another set of weights. In image classification, the vector of extracted features from the image is connected to the input layer, and is match with the desired class labels in the output layer.

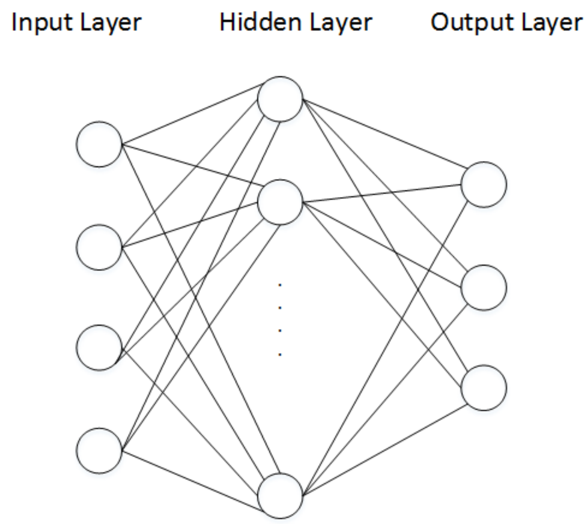


Fig. 2. 5 Model of MLP with one hidden layer.

The layers in a neural network consist of a number of nodes (neuron), where each node connects to the weights from the input layer by summation, and is compared with a threshold (or bias). Subsequently, it is converted to the output by a transfer function (2-1) (Zupan, 1994).

$$y = f\left(\sum_{i=1}^{N} w_i x_i - \beta\right) \tag{2-1}$$

where y is the output of the node, x_i is the i_{th} input, w_i the weight of the i_{th} connection, β is the threshold, and $f(.)$ is the activation function. The activation function can be a step function or a sigmoid function (2-2).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2-2}$$

The weights adjusted by the real output, using a common training algorithm call back propagation, are very close to the desired output. The adjustment or modification of the weights occurs in the training stage, where such stage commences with random weights. The modification of the weights is done layer-by-layer, starting from the output layer by calculating the error using the Mean Square Error (MSE), and then going backward in the hidden layer to change the weights. The MSE is calculated from expected and actual output vectors. This leads to a decreasing MSE when the procedure is repeated many times, resulting in new weights (Tu, 1996; Zupan, 1994).

(2) Support-Vector Machine

Support-vector machine is one of learning algorithms that has been applied in different applications of image classification (K. I. Kim, Jung, Park, & Kim, 2002; Rajpoot & Rajpoot, 2004). SVM is a more recent classifier than BPNN, where the last has a problem of over-fitting, while the former provides better results with data of short length (Candade & Dixon, 2004).

SVM relies on statistical learning for separating between different sets of data (Evgeniou, Pontil, & Poggio, 2000). SVM is designed for separating two classes of data in the classification stage, where the separation between the two classes of data is done by hyperplane. The hyperplane is supposed to separate between these different classes of data by maximal margin, such as desirable classes and undesirable classes of dataset. The operation starts by mapping training samples of feature vectors into a space of a higher dimensionality, which is called the feature space. In higher dimensional feature space, applying linear separation between different classes is possible. Then, the margin of separation between classes in the higher dimensional space of training samples is determined. The resulting optimal margin should be of as large width as possible between the two different classes. Fig 2.6 illustrates two different samples of data from the desirable class (+), and three samples from the undesirable class (o). These samples of (+) and (o) are determined by hyperplanes L1 and L2, respectively. There are points of data in the border of hyperplanes L1 with three points of sign (+) and L2 based on two points of signs (o), where these are called support vectors (Cortes & Vapnik, 1995).

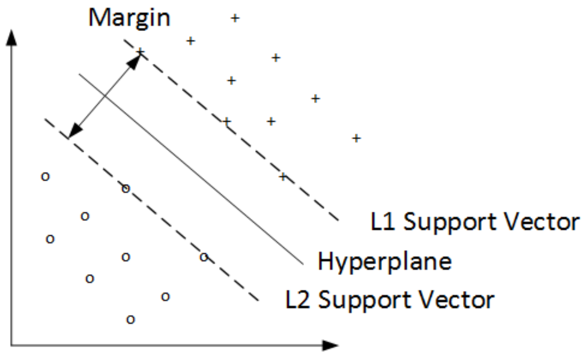


Fig. 2. 6 Linear SVM for two classes of data.

The following equation defines the form of the hyperplane:

$$w.x_i + \mathrm{b} = 0$$

where x_i is the attribute set for the ith training data, which belong to one of the two classes, (+) or (o). This training set of samples should find a linear classifier that separates all training data samples. In the training stage, the SVM of a couple of lines will search for the widest margin that separate between different classes of samples that are shared in training, such that: w x_i + b = 0 separates between different samples. The first class of samples belongs to w. x_i + b > 0, and that determines the hyperplane L1, whereas the second class of samples belongs to w. x_i + b < 0, which determine the hyperplane L2. The width between the hyperplanes is (w) (Ivanciuc, 2007). For more than two classes (i.e. a multi-classes problem), the classification is performed by a one-against-others approach using multi-class SVM (Hsu & Lin, 2002) .

2.2 Texture Feature Description

The purpose of texture descriptors is to extract information from the texture, in the form of numerical numbers called ‘‘features’’. This stage is critical for any texture application to capture significant properties of texture (Chellappa, Kashyap, & Manjunath, 1993). It is possible to find that features from a certain descriptor work better on particular applications; however, there is not a certain feature that is appropriate for different applications (W.-C. Lin et al., 2004). For discriminating features from texture, it is important to define which features are required to be computed, and what kind of processing of these features is needed, both of which depend on selecting the proper descriptor (Chellappa et al., 1993). Here, only the statistical and signal processing methods are reviewed, which are appropriate for real images.

2.2.1 Statistical Descriptors

In statistical mode, first order statistics are not suitable for defining properties of a texture. In this mode, the descriptors for texture tend to use higher order statistics such as Autocorrelation, Grey-level Run Length, Grey-level Co-occurrence Matrix, Grey-level Difference Matrix, and Local Binary Patterns.

2.2.1.1 Autocorrelation

Autocorrelation (AC) is used for assessment of the percentage of regularity in the texture using equation (2-3). AC describes the spatial organisation of texture by the correlation coefficient, which evaluates linear spatial relationships between texture primitives (Mihran Tuceryan & Jain, 1993). For example, to compare between fineness and coarseness of the texture, the AC function drops off quickly with fineness textures, whereas it drops off slowly with coarseness textures. Although AC can be applied for different texture characteristics, the features from the AC function are not effective enough for texture classification, especially with natural images (Pratt, 1991).

$$AC_{\Delta_u,\Delta_v}(x) = \frac{\sum_{u=1}^{M}\sum_{v=1}^{M} x(u,v)x(u+\Delta_u\,, v+\,\Delta_v)}{\sum_{u=1}^{M}\sum_{v=1}^{M} x^2\ (u,v)} \tag{2-3}$$

where x is the $M \times\ M$ image, and Δ_u, Δ_v are horizontal and vertical displacements.

2.2.1.2 Grey-level Run Length

Grey-level run length (GLRL) was introduced by Galloway (1974) as a means for extracting the properties of texture statistically. It calculates the features in the same line that has the same grey level value. GLRL can be applied in different directions, but is usually used with directions 0°, 45°, 90° and 135° (see Fig 2.7 for 0° and 45°).

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

0°	1	2	3	4
0	4	0	0	0
1	1	0	1	0
2	3	0	0	0
3	3	1	0	0

45°	1	2	3	4
0	4	0	0	0
1	4	0	0	0
2	3	0	1	0
3	3	1	0	0

Fig. 2. 7 Sample of pixels' values, and GLRL matrices at 0° and 45°.

For characteristic features, usually a number of measurement is taken instead of depending on produced run length matrices. For instance, short run emphasis, long run emphasis, grey-level non-uniformity, run length non-uniformity, and run percentage have all been used as feature extractors from texture. A longer run-length is used for describing a coarser texture, whereas a short run-length is used for fine texture. For differences between random and non-random textures, the result of GLRL is more uniformly distributed with a random texture. It is clear that feature extraction by GLRL is not difficult in calculation, however, the reported results from this method are not encouraging (Conners & Harlow, 1980).

2.2.1.3 Grey-Level Co-occurrence Matrix

The Grey Level Co-occurrence Matrix (GLCM) was introduced by Haralick in 1973 (Haralick & Shanmugam, 1973). Texture features can be calculated by GLCM, which is one of oldest statistical methods used for the analysis of texture. It creates a new matrix that is dependent on grey-level values of the original image matrix. The number of rows and columns in the original image is equal to the grey tones of the new matrix. The GLCM method provides information about the type of texture in the image from the relationship between pairs of pixels. The values inside the new matrix take two parameters into consideration: distance and angle, as in Fig 2.8.

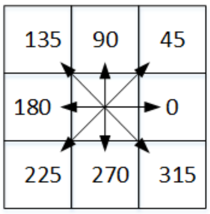


Fig. 2. 8 Eight directions of adjacency in GLCM.

The grey-level intensity values of two pixels with a particular spatial relationship compute the distance of GLCM. Angles determine the direction of the relationship between two pixels of the same grey-level, which can be horizontal, vertical or diagonal. Fig 2.9 explains the creation of the co-occurrence matrix (C-M) from the simple pixels' values of the image matrix (Im-M). GLCM determines differences between surface textures through the collection of elements around a diagonal in the matrix. For example, rough and smooth surfaces will be different, and can easily be classified using GLCM (Al-Janobi, 2001). The dimensions of GLCM are calculated by the grey-level of the image. More levels provide more accuracy in extracting the information from the texture, at cost of increasing computational complexity (Soh & Tsatsoulis, 1999).

1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

C-M

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Im-M

Fig. 2. 9 Images matrix of pixels' values converted into GLCM.

2.2.1.4 Grey-level Difference

The grey-level difference method extracts local features by calculating absolute differences between pairs of grey-level pixels (2-4) (Weszka, Dyer, & Rosenfeld, 1976). The method follows nearly the same strategy of GLCM, which calculates statistical features by considering the distribution of local contrast in different directions. After producing the matrix, a number of measurement are applied to extract the features, including the mean, entropy, contrast, and angular second moment.

$$f\ (\Delta_x\,,\Delta_y) = |f(x,y) - f(x + \Delta_x\,, y +\ \Delta_y)| \qquad (2\text{-}4)$$

where $x\ and\ y$ is the position of pixel, $\Delta_x\ and\ \Delta_y$ is the displacement in $x\ and\ y$ direction.

2.2.1.5 Feature-based Texture Units

The aforementioned methods are among the first and common statistical methods used with texture. These co-occurrence methods estimate grey-level pixels by a displacement vector in specific directions. However, such methods do not produce sufficient information from different textures (He & Wang, 1990). For more effective extraction of information from the texture of the image, He & Wang used a method based on Texture Units (TUs), which is the smallest complete unit of image (He & Wang, 1990). In images, each pixel (central pixel) is surrounded by a number of pixels, called neighbour pixels. The relationship between any central pixel and its neighbours can be represented by eight directions surrounding the centre pixel (the horizontal, vertical, diagonal, and anti-diagonal pixels), where the smallest complete unit in the image is called the Texture Unit. The researchers applied texture spectrum (TS) on a set of produced TUs of features.

In TS, each element in the texture unit is replaced by one of three values (0, 1, 2), as explained in (2-5), which is the relationship between the surrounding 3×3 pixels with the pixel in the middle.

$$a_i = \begin{cases} 2, & if\ p_i > 0 \\ 1, & if\ p_i = 0 \\ 0, & if\ p_i < 0 \end{cases} \qquad where\ 1 \le i \le 8 \tag{2-5}$$

The texture spectrum is calculated using a histogram based on TUs, which is used as the features of the image. However, before using the histogram, every centre pixel in TUs is replaced by a summation of neighbour pixels from functions of logical operators and a product of their weighted, using equation (2-6). The result is 3^8= 6561 possible units of texture, which describe the spatial three-level patterns of neighbourhood pixels of TUs.

$$N_{TU} = \sum_{i=1}^{8} a_i \cdot 3^{i-1} \tag{2-6}$$

where N_{TU} represents the texture unit number, and a_i is the i^{th} element of texture unit.

The rest of this section introduces one of the most significant methods based on TUs for extracting features from texture (Local Binary Patterns - LBP), before subsequently introducing other extension methods based on this method.

(1) Local Binary Pattern

The histogram size from TS is unpractical, because the feature length is very long. To reduce the large value resulting from texture units by TS, Ojala et al. (1994) used TU for binary coding of texture patterns, which is called Local Binary Patterns (LBP). LBP replaced the quantization 0,1, and 2 in TS by only 0 and 1 (equation (2-7)), and as such, there are only 256 possible TUs (from 2^8, as explained in equation (2-8)).

$$a_i = \begin{cases} 1 & , p_i \ge 0 \\ 0 & , p_i < 0 \end{cases} \qquad where\ 1 \le i \le 8 \tag{2-7}$$

$$LBP_{P,R}(x,y) = \sum_{i=1}^{n} a(g_i - g_c)2^{i-1} \quad (2\text{-}8)$$

where g_i and g_c are the grey-level values at a neighbouring pixel and the centre pixel, respectively. The number of the pixels in the circular neighbourhood is denoted by n, and $a(.)$ is a binary quantisation of intensity value of patterns

Figure 2.10 explains in steps how the centre pixel in TUs is replaced by summation of neighbour values that result from the binary coding with corresponding weights.

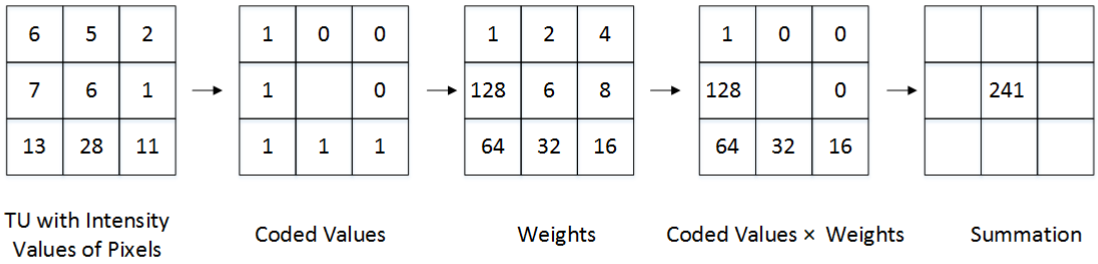


Fig. 2. 10 Computing LBP from sample of TU.

LBP has emerged as the most effective method used with textures classification, with a very low computational cost. In addition, LBP, with its invariance against luminance change, ensures that no effect takes place on signed differences between the middle pixel and the other surrounding pixels, which makes LBP invariant to grey-level shifting. We will utilise LBP later for developing new features for texture classification.

(2) Extensions of feature methods based on LBP

Conventional LBP descriptors define a small area in the image using a 3x3 window. However, this limited window is not effective for capturing enough structural information that can take place at wider scales. Ojala, Pietikainen, et al. (2002) modified the conventional LBP descriptor to be an invariant method to rotation and grey-scale. To accomplish different scale analysis, for the neighbourhood pixels surrounding the centre pixel, any number of neighbour samples (P) can be selected from the circular perimeter at any scale (R). However, a multi-scale analysis can be executed in a circular or square shape of neighborhood, which is appropriate in some applications that are not affected by rotation (Pietikäinen, Hadid, Zhao, & Ahonen, 2011). LBP with multiresolution analysis usually improves the discriminative capability of feature, and in turn, the accuracy of texture classification.

The feature length of LBP is 256, resulting from the number of patterns being equal to eight. When depending on multi scale analysis, the size of the histogram grows rapidly by 2^n. For example, with n= 16 neighbour samples, the size of the histogram would be 2^{16}. Increasing in the number of neighbour samples makes the LBP impractical for use in real applications.

The modified method compares pairs of pixels which are center-symmetric with each other. Such method produces 16 (2^4) rather than 256 (2^8) different binary patterns (Heikkilä, Pietikäinen, & Schmid, 2009).

In order to reduce the feature length resulting from the histogram with multi-scale, Ojala, Pietikainen, et al. (2002) utilised uniform patterns instead of complete neighbour patterns. In every scale, there is a limited number of transitions of the pattern.

One of the important methods extended from LBP is the Local Ternary Patterns (LTP) method. Tan and Triggs (2010) introduced LTP in order to process the sensitivity to noise in the conventional LBP, as LTP was intended for use in face recognition. LTP is less sensitive against noise in uniform regions, and it also usually provides more discriminative features. LTP has 3-valued codes, the ones above (±t) are equal to 1, those below (±t) are equal to -1, which are then replaced by 1 in a further process to reduce the size of resulting histogram, whereas those between (±t) are always equal to zero (2-9).

$$a_i = \begin{cases} 1, & if\ p_i > c + t \\ 0, & if\ \mathrm{c} - t \leq p_i \leq c + t \\ -1, & if\ p_i < c - t \end{cases} \qquad where\ 1 \leq i \leq 8 \qquad \text{(2-9)}$$

For analysing medical images and for more information from the texture, the author extended the LBP method into Local Quinary Patterns (LQP) encoding by adding two values of threshold (t1 and t2) for texture patterns to be divided into four binary patterns, as explained in equation (2-10) (Nanni, Lumini, & Brahnam, 2010).

$$a_i = \begin{cases} 2, & if\ p_i > c + t2 \\ 1, & if\ c + t1 \ \le p_i < c + t2 \\ 0, & if\ c - t1 \ \ge p_i < c + t1 \\ -1, & if\ c - t2 \le \ p_i < c - t1 \\ -2, & p_i < c + t2 \end{cases} \qquad where\ 1 \le i \le 8 \qquad (2\text{-}10)$$

It can be seen that, LBP improves the TS method by effective binary quantisation of intensity values of neighbour pixels, whereas LTP and LQP are used for extracting better features than LBP from TUs by considering more intensity values of the neighbourhood pixels.

2.2.2 Signal Processing Descriptors

Signal processing methods, also called filter methods, are applied for filtering images to capture relevant information. They have the ability to deal with different texture characteristics, such as developing a multiscale approach for the scale problem in the texture, which is the main property of these methods. Signal processing schemes are more modern methods than statistical methods.

2.2.2.1 Fourier Transform

Joseph Fourier introduced the Fourier transform (FT) method in 1807, which is a periodic function of an infinite sum of complex exponentials. Fourier analysis, as other spectrum methods, has the ability to study texture properties using the power spectrum, which provides information about texture properties, such as distinguishing between coarseness and fineness, or between directional and non-directional textures (William Henry Nailon, 2010).

Through signal processing, FT produces a global frequency without referring to the time, which is the main problem of FT. This is addressed by using the Short Time of Fourier transform (STFT), which processes the signal by a window function. Using a window of Gaussian function, and multiplying the input signal by the window produces the Gabor transform.

$$F(u,v) = \frac{1}{\omega^2} \sum_{x=1}^{M} \sum_{y=1}^{M} f(x,y) e^{\frac{-2\pi j}{\omega}(ux+vy)} \qquad (2\text{-}11)$$

The complex FFT represents magnitude (|F|), which is the absolute value or power spectrum density, and phase (θ) information of the signal in the frequency domain. As previously mentioned, the power spectrum density (PSD) represent the contents of an image using global frequency, which is directional and symmetric, where FFT is used to obtain high performance textural features. Zhou, Feng, and Shi (2001) studied texture classification and texture retrieval by local Fourier analysis. He proposed features extraction based on a histogram that resulted from describing Fourier coefficients of the local similarity of texture.

2.2.2.2 Gabor Filter

According to studies on the Human Visual System (HVS), textures depend on three primary characteristics, which are: frequency, orientation, and complexity. A Gabor filter (GF) mimics this system, because it can localise frequency and orientation characteristics of images by decomposing the images into different spatial frequencies and directions, and as such, GF is categorised under Multi Scales Multi Directions (MSMD) methods (Clausi & Jernigan, 2000).

The Gabor function was introduced by Dennis Gabor, who later won the Nobel-prize (Gabor, 1946). The main purpose of his study was the analysis of information and its transmission to speech. The signal was analysed symmetrically in the time and frequency domains. The development of GF was made by Daugman, who applied GF for two-dimensional (2D) signals, and found that the 2D Gabor filter gave good description for cells in an animal visual cortex (Daugman, 1985).

The extracted features from the texture by Gabor filters shows high discrimination. Since introduced, GF has seen widespread use in many applications, such as texture segmentation, image retrieval, and image classification (Idrissa & Acheroy, 2002; A. K. Jain & Farrokhnia, 1991; Manjunath & Ma, 1996). It is an unsupervised texture classification method that uses different frequencies and orientations of the Gabor filter, performed together with fuzzy clustering algorithm. Gabor filters are used to extract features for texture classification. The weakness of this method is that the numbers of clusters need to be specified in advance (Idrissa & Acheroy, 2002). A similar study (Kamarainen, Kyrki, & Kälviäinen, 2002) shows the invariant nature of Gabor filters to rotation and translation in image recognition. The fundamental frequencies of Gabor filters are used to represent the object shape for classification. Even symmetric Gabor filters are a robust method for rotation invariance. The classification rate on rotated textures of the Brodatz database with different directions reach just above eighty per cent (Manthalkar, Biswas, & Chatterji, 2003). Furthermore, Gabor filters are used in texture feature extraction, and are often compared with other techniques of texture analysis. This comparison takes into consideration both the filter's characterization and the feature extraction, based on the filter's outputs (Clausi & Jernigan, 2000).

Gabor filters are utilised in spatial and frequency domains. In the spatial domain (Equation (2.12)), they are a sinusoid wave modulated by a Gaussian envelope, where the bandwidth of the filter is determined by the standard deviation of the Gaussian envelope, and the direction and frequency of the sinusoid signal refer to the direction and frequency of the pass band of the filter.

$$\Psi(x, y; f_0, \theta\ ) = \frac{f_0^2}{\pi\gamma\eta}\ e^{-\frac{f_0^2}{\gamma^2}x'^2+\frac{f_0^2}{\eta^2}y'^2}\ e^{j2\pi\ f_0 x'} \tag{2-12}$$

$$x' = xcos\theta + ysin$$

$$y' = -xsin\theta + ycos\theta$$

where f_0 is the central frequency of the filter, θ is the angle between the sinusoidal wave direction and the x-axis, γ and η are Gaussian envelope's values in the wave direction of major axis and orthogonal to the wave direction (the minor axis), respectively.

In the frequency domain (Equation (2-13)), Gabor filters are Gaussian bell-shape filters in various orientations, and of different horizontal and vertical central frequency:

$$\psi(u, v; f_0, \theta\ ) = e^{-\pi^2\frac{u'-f_0}{\alpha^2}+\frac{v'}{\beta^2}} \tag{2-13}$$

$$u' = ucos\theta + vsin\theta$$

$$v' = -usin\theta + vcos\theta$$

For utilising GF in classification, it is necessary to apply a number of them (filter bank) on the image. To elaborate, a filter bank contains a set of filters with different parameters, and these parameters should be taken into consideration when designing the filter bank. To construct Gabor features, it is common to calculate Gabor filter response over the input image for a set of filters, called wavelet, which are tuned to various orientations and frequencies. This is to ensure that objects within an image can be categorised at different orientations, scales and translations (Kyrki, Kamarainen, & Kälviäinen, 2004).

Determining different frequencies is done via equation (2-14), which represents the scale invariance property of Gabor filters. The frequency values f_u are based on f_{max}, which describes the maximum central frequency, and the spacing factor between the different central frequencies λ, which is normally set to $\sqrt{2}$.

$$f_u = \frac{f_{max}}{\lambda^u}, u = \{0, \dots, u - 1\}, \quad (2\text{-}14)$$

where u is the scale size. The selection of discrete rotation angles is done via equation (2.15), which space the orientations uniformly.

$$\theta_v = \frac{\pi v}{V}, v = \{0, \dots, V - 1\} \quad (2\text{-}15)$$

These parameter values are used to cover the frequencies and orientations of interest for a Gabor wavelet, as shown in Fig 2.11, which illustrates five scales and eight orientations.

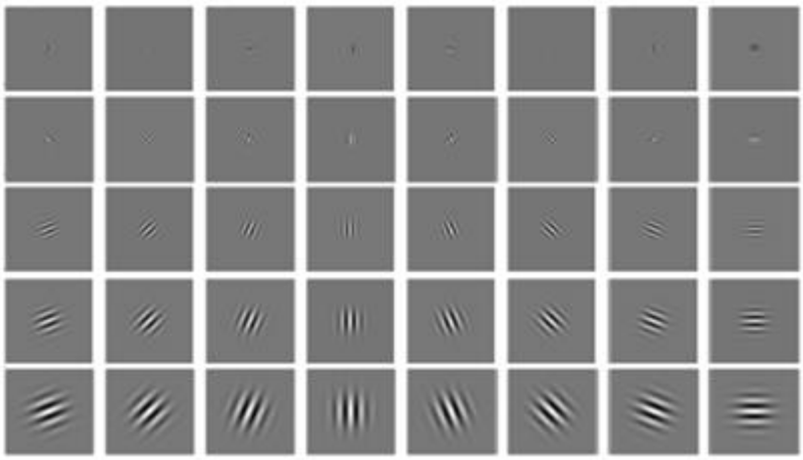


Fig. 2. 11 Gabor filter consisting of five frequencies and eight orientations.

The Gabor feature matrix is created by convolving the image with a particular Gabor filter. Here, a filter response that represents the amount of overlap between the filter and the texture in the image is obtained.

For multi-resolution methods, Gabor filters and wavelet transform are the most utilised methods of images decomposition. However, Gabor filters is the preferred approach for texture. The wavelet transform decomposes the image into three directions, namely 0◦, 45◦ and 90◦ , which are limited to the horizontal, diagonal and vertical sub-bands (Arivazhagan, Ganesan, & Priyal, 2006). Gabor filters, defined by equation (2-15), have the ability to decompose the image into multiple orientations.

Many comparison studies of wavelet transform methods of texture characteristics found that Gabor filters are more appropriate others. In (Ahmadian & Mostafa, 2003), the authors compared Gabor wavelets and dyadic wavelets as texture feature extraction methods. Dyadic wavelets are a wavelet transform, which applies two 1D transforms separately to reduce the complications of applying a 2D wavelet transform. In the context of texture classification, Gabor wavelets offer more discrimination of features from textures, and produce a higher accuracy. In another study (S. Li & Shawe-Taylor, 2005), authors compared multiscale several spectrum methods, namely dyadic wavelet, wavelet frame, Gabor wavelet, and steerable pyramid. They found that the steerable pyramid and Gabor wavelet achieved higher classification rate, whereas dyadic wavelet obtained the least accuracy, as it has a problem with translation invariance. The improved wavelet method obtained better results, but still lagged behind the steerable pyramid and Gabor wavelet in terms of performance. In another related work by Ma and Manjunath (1995), different wavelet transform methods were applied, such as the Orthogonal Wavelet Transforms (OWTs), Bi-orthogonal Wavelet Transforms (BWTs), and tree-structured decomposition (using orthogonal filter tree-structure decomposition and bi-orthogonal filter), and Gabor Wavelet Transforms (GWTs). The Gabor feature outperformed others, however, it was computationally complex, which is the main problem arising in its applications.

Supervised and unsupervised methods are the two sets of methods used to employ GF for texture applications (Clausi & Jernigan, 2000). Unsupervised methods are based on a set of filters with different frequencies and orientations in the bank, and are used without advance information about the texture in the image. Although they are more popular, the computational cost is expensive, especially with a large set of filters (Randen & Husoy, 1999). On the contrary, supervised methods are based on a particular filter or number of filters for a given problem. To achieve a more effective approach in texture applications, the objective of these supervised methods are to identify textural boundaries using only a minimum number of filters. Here, the parameters of filters are taken into consideration when finding the appropriate filters, which is applied as a solution to the high computational costs of GF (Bianconi & Fernández, 2007).

Parameters optimisation is the main challenge associated with GF. The optimum parameter values of a filter, where the filter has the highest sensitivity to the texture's patterns of the image, are different for each of the filters in the bank. Thus, exploring the influence of the parameters on the filter performance is important. In (L. Chen, Lu, & Zhang, 2004), the authors examined the influence of Gabor filter parameters, such as the number of scales, number of orientations and the filter mask size, on image retrieval. They found that an appropriate filter mask size and a suitable mixture of numbers of scales and orientations have a substantial impact on the performance and the computational cost of the process. In addition, Gabor filter parameter selection is influenced by the characteristics of the textures in the database. Another study showed the importance of the smoothing parameter (standard deviation of the Gaussian envelope) in Gabor filters, and that the relationship of frequencies and orientations of the filter had less impact on the classification results (Bianconi & Fernández, 2007).

In a bank of filters, not all created filters have the same significance, where only a number of the filters might be used, while the others may be redundant or not useful (W. Li, Mao, Zhang, & Chai, 2010). For features with discriminative power, the appropriate values of GF parameters should be set. However, based on experience, this becomes increasing challenging when faced with a large set of parameters.

Many optimisation methods have been employed for determining the optimum parameter values of GF. Genetic algorithm (GA) have been applied to GF parameters successfully in numerous optimisation tasks. The parameters were selected based on every image set in a database (Afshang, Helfroush, & Zahernia, 2009). A genetic method utilising a clustering algorithm was used for filter selection, where the clustering algorithm groups the filters to remove redundant information from similar filters (Sun, Bebis, & Miller, 2003). Another study was conducted using a genetic algorithm for the selection of a suitable value set of filter parameters, which included smooth parameters of a Gaussian envelope, as well as orientations and frequency values. The accuracy ranged from 97.5% to 96.9% for 16 and 6 filters, respectively (Pakdel & Tajeripour, 2011). In (Zavaschi, Britto Jr, Oliveira, & Koerich, 2013), the authors introduced a new method using a genetic algorithm that supports the vector machine used for recognition of facial expression, where Local Binary Patterns and a Gabor filter were used as an integrated method for feature extraction.

Particle Swarm Optimization (PSO) is a swarm method used for selecting a subset of features from the original features, where in (Kumar, Patidar, Khazanchi, & Saini, 2016), the features were extracted from a leaf image by Gabor filter for leaves classification. For iris recognition, the optimized Gabor filter can be used for decomposing iris images. Particle Swarm Optimization is also utilised for optimization of Gabor parameters values (Tsai, Taur, & Tao, 2009).

2.3 Texture Feature Fusion

2.3.1 Overview of Feature-level Fusion

Feature level fusion is one of the most common methods of integrating multiple features from different descriptors. It is based on the assumption that the existing features can be improved when combined with other features, which can be used as complementary features.

Images classification, especially by texture, is a major challenge, due to the wide diversity of different characteristics (refer to Section 2.1.1). The ability to classify different texture types is no longer sufficient using a single descriptor. This is because it is difficult to capture different characteristics of textures using any single descriptor (Bashar & Ohnishi, 2002). Most texture description methods in Section 2.2 are highly dependent on the particular type of texture. The applied descriptor is usually appropriate for a particular class of texture images, where its ability degrades and renders it unsuitable for other texture classes.

In order to improve the strength of feature extraction, the current trend involves combining the descriptors with each other (Solberg, 1996; Solberg & Jain, 1997). An appropriate combination of different features is required, which provides diversity of information for the problem. Successful fusion is based the complementary features of the combined methods. The complementary features are used for exploiting the diversity between features from the shared methods. Depending on the many resources used together for extracting information, the ability to recognise images from an environment can be enhanced. The integration of feature descriptors produces highly discriminative and effective features, which are robust to changes in imaging effects, such as random noise or blurring of the image (R. S. Blum & Liu, 2005).

The classification system has two main stages, which are the feature extraction stage and the classification stage. Therefore, fusion can be performed at the feature-level or at the decision-level of classification. Fusion at the feature-level is referred to as a pre-mapping, whereas fusion at the decision level is referred to a post-mapping fusion (Pohl & Van Genderen, 1998). The features are extracted by descriptors independently, and then combined in the feature level, where this is performed before the assessment of features by the classifier.

Feature level fusion is applied by directly concatenating several of the shared features, where the simplicity in implementation is main advantage of this method. Fusion between different features from different descriptors produces more discriminative features, and achieves robustness and accuracy (Bashar & Ohnishi, 2002). On the other hand, concatenation may increase the cost of computation from the resulting high feature dimensionality, which is the main drawback of combining different features. In concatenation of feature descriptors, if the number of descriptors is set by N and M-dimensions of extracted information from each descriptor, that dimensionality of the concatenated of participated descriptor will be M × N, which will increase the burden of computation.

2.3.2 Feature-level Fusion – Related Work

Because there are a wide variety of texture feature extraction methods, there have been real efforts by researchers to compare such methods to evaluate their performance separately, or when integrated together.

S. Li and Shawe-Taylor (2005) compared four spectrum methods, which are the dyadic wavelet, wavelet frame, Gabor wavelet, and steerable pyramid. In the experimental results, the steerable pyramid and Gabor wavelet achieved higher classification rates, and the combination of the two methods achieved better results than applying any one of them separately.

Barley and Town (2014) applied statistical and filter-based methods, which included the grey-level co-occurrence matrix, Gabor wavelets, and steerable pyramids. The results proved that fused features from two methods provide a higher classification accuracy.

In a similar study on a mix of statistical and spectrum-based methods, M. Singh and Singh (2002) examined the performances of the co-occurrence matrices, edge frequency, Laws' masks, run length, binary stack, texture operators, and texture spectrum methods. In general, they concluded that using combined methods followed by feature selection improved texture recognition.

In another related work (Ojala et al., 1996), the author compared a number of common feature methods that had been used before and new ones for texture classification. The compared methods included the grey-level difference method, Laws' texture measures, centre-symmetric covariance measures, and local binary patterns. The local binary patterns method was found to be more effective when combined with a contrast of texture measures for powerful feature detection. Local contrast was used as complementary approach to sign features of LBP. The patterns in the image and the greyscale image provide different information about texture. Integrating the conventional LBP with contrast measures from image texture was thus shown to enhance the accuracy of LBP significantly. LBP is an invariant to grey-scale, and can be integrated with other methods to be effective.

LBP produces powerful features with its simplicity, however, its application is often restricted to dealing with low discriminative types of texture, and there are efforts to make LBP a more discriminative texture extraction method (Ojala et al., 1996). LBP integrated with various other feature detection methods can be categorised into, fusion LBP with other similar features of LBP code, and fusion LBP with different descriptors features (L. Liu, Fieguth, Guo, Wang, & Pietikäinen, 2017).

In (Z. Guo, Zhang, & Zhang, 2010a), Completed LBP (CLBP) was the first attempt in integrating various features of LBP style. The integrated features included three different components. The first and second components are the results of the local differences between a centre pixel and its neighbours, which are the signs (CLBP_S) and the magnitudes (CLBP_M). These components work as complementary features to each other, where CLBP_M works as alternative features of contrast. The last component, CLBP_C, is used for global thresholding to obtain more discriminative information.

In (Ahmed, Hossain, Bari, & Shihavuddin, 2011), to obtain more discriminative features, the authors applied the same components of CLBP, with the exception of using two bits with neighbour pixels for both the sign and the magnitude components. The main reason for carrying out this approach was to obtain more discriminative information, and to avoid the long dimensionality of features the 16-bit pattern divided into two parts 8-bit pattern.

In (Y. Guo, Zhao, & PietikäInen, 2012), a learning model with CLBP components was applied for more discriminative features. The sign and magnitude components were selected to be invariant to rotation, and a threshold value was then defined from taking the average values of distance between the values of neighbours from the whole image to produce disCLBP.

In (L. Liu, Zhao, Long, Kuang, & Fieguth, 2012), features based on the radial-difference (RD) and angular-difference (AD) were extracted to be combined with two components from the intensity of the central pixel (CI) with its neighbours (NI).

In (L. Liu, Lao, et al., 2016), the authors proposed a Median Robust Extended LBP (MRELBP) method for extracting global information to deal with the noise in the image, which is main problem facing the conventional LBP method. In MRELBP, instead of depending on the intensity values of pixels, the method calculates medians from the regional image rather than the raw image intensities. This method applies the median approach efficiently on multi-scales with combined microstructure and macrostructure features.

For dealing with noise, the Adjacent Evaluation Local Binary Patterns (AELBP) method was introduced in (K. Song, Yan, Zhao, & Liu, 2015) for texture classification. It replaces the central pixel as the threshold pixel in LBP by an adjacent evaluation window. Furthermore, it sets an evaluation centre (ap) as the new threshold by a window that contains neighbours of the neighbourhood of centre (gc). This method has the ability to be combined with the Completed Local Binary Pattern (CLBP) or Local Ternary Pattern (LTP) methods.

LBP may classify different patterns to the same class as a result of the same LBP code. To obtain more discriminative features, the Local Structure Patterns (LSP) method was introduced in (Shrivastava & Tyagi, 2014). The patterns in the method are thresholded from a summation of centre pixel values and the average local differences. This provides more accurate classification results of textural structures by utilising local and global information. Furthermore, to improve the classification performance, LSP can be converted into Completed Local Structure Pattern (CLSP) by combining the conventional LBP method and centre pixel component.

LBP features can be combined with other descriptors that work as complementary features extractors, in order to improve the accuracy of the features detection. In (Z. Guo, Zhang, Zhang, & Zhang, 2010), LBP histogram was joined with an absolute difference of pixels method. The features used for directional information extracted from the texture were ignored from LBP. This made LBP invariant against rotation. The statistical information from absolute difference of pixels were extracted by the mean and standard deviation values.

In (Liao et al., 2009), to achieve an rotation-invariant LBP method, the Dominant Local Binary Patterns (DLBP) method was combined with a circularly symmetric Gabor filter. The DLBP method detected the frequently patterns occurring from the texture, whereas global directional features from the texture were obtained by Gabor filter.

A new approach proposed by (Xiaoyu Wang, Han, & Yan, 2009) involved combining LBP with the Histograms of Oriented Gradients (HOG) method for detection of human objects, whereas, in (Hussain & Triggs, 2010), LTP was added to LBP and HOG to detect more visual features.

In (Khellah, 2011), global image features were extracted by the Dominant Neighbourhood Similarity (DNS) method, and fused with local features detected by LBP for texture classification. Global dominant information is captured from calculating the average representation of the texture from specific windows, where every window calculates the similarity between an intensity pixel with its neighbours.

In (Satpathy, Jiang, & Eng, 2014), LBP was proposed as a solution for distinguishing between bright an object and its dark background. The LBP method and its complement were concatenated via their histogram.

A new approach introduced in (Z. Guo, Zhang, & Zhang, 2010b) involved combining contrast information with the LBP Variance (LBPV) method. The principal orientations of the texture were firstly extracted, then concatenated with the LBP histogram.

Integrating between different feature extraction method mostly increase the feature size. There are different approaches to overcome the dimensionality of LBP when it is integrated with other descriptors. In (Shan & Gritti, 2008), AdaBoost was used for improving the discrimination capability of LBP histogram by removing unnecessary LBP bins. The resulting features were used in facial expression recognition. The AdaBoost approach had also used for selecting Gabor wavelet features.

There are different methods used for projecting high-dimensional feature methods into a lower dimensional one. The Principal Component Analysis (PCA) is applied frequently with high-dimensional feature methods, and is one of the most common methods used with the LBP histogram. In (Tan & Triggs, 2007), PCA was applied with the combined features of LBP and GF in face recognition application. The features of LBP and GF are of high dimensionality, thus, they were reduced by PCA separately before integrating the reduced features together. On the other hand, in (Chan, Kittler, & Messer, 2007), the high dimensionality resulting from the Multi-Scale LBP features was dealt with via Linear Discriminant Analysis (LDA). LDA was adopted to project the features into lower size.

In (Hussain & Triggs, 2010), three different types of features, Histogram of Oriented Gradient HOD, LBP, and LTP, were proposed as complementary feature detection methods for improving object detection. The dimensionality problem from combining the aforementioned methods was addressed by the Partial Least Squares (PLS) approach. The aim was to extract fast and more discriminative features.

The next section is devoted to features background selection approaches, which are suggested for addressing the feature fusion problem. The purpose is to improve feature detection without suffering from the high dimensionality drawback.

2.4 Texture Feature Selection

2.4.1 Feature Selection – Overview

Dimensionality reduction in image classification is introduced as a solution for the high dimensionality problem ‘‘curse-of-dimensionality’’. Dimensionality reduction occurs by feature extraction methods and feature selection methods (A. K. Jain, Duin, & Mao, 2000; Solberg & Jain, 1997). Feature extraction methods refer to the methods that generate new features by a set of transformations. This happen by projecting the feature into a space having fewer dimensions (Solberg & Jain, 1997). However, feature transformation methods produce different representation of original data, by changing the semantic of features (Shang & Shen, 2008).

Feature selection is also employed for reducing the dimensionality of features by selecting part of the features, called relevant features, from the original features (A. L. Blum & Langley, 1997). The selected part of features should be adequate to describe the images with the same or better ability than the original features. In many applications, preserving the meaning of data is important when selecting part of data (Guyon & Elisseeff, 2003). The advantage of feature selection do not occur in features construction, where the selected features preserve the meaning of original features (Jensen & Shen, 2007).

Feature selection is usually essential to carry out after fusing between different features, as part of the relevant features become irrelevant, which can negatively affect other relevant features (A. Jain & Zongker, 1997). Removing irrelevant feature reduces the size of features, as well as usually improves the quality of features when compared with the original features (Deogun, Choubey, Raghavan, & Sever, 1998).

Supposing that the feature selection problem is defined by selecting a subset of features with size X from a given set of original features with size Y, such that $X \subseteq Y$. The classification accuracy resulting from the subset of features (X) is relevant if it is the same or better than depending on the original features. In such a case, X represents the optimal features, which are the most suitable features for classification that result in the highest possible accuracy.

In practice, finding the optimal features is a difficult and expensive task (Jensen & Shen, 2007). Feature selection is an interesting research area, and various strategies have been develop to deal with the associated challenges (Dash & Liu, 1997). The main problem encountered when selecting optimal features is that the relations between feature items are mostly complicated. It is generally not a good solution to rank the features items as individual items based on the best quality, then selecting the first subset of features that are sufficient for improving the classification accuracy. Feature items may individually produce reasonable results from a fitting function, whereas they produce less effect results when combined with other items. In other situations, items may produce better results when connected with other items than if applied individually, as they depend strongly on each other, which has positive interaction effects. The only solution for this problem is; instead of estimating each feature items individually, the search for optimal subsets is applied between different combinations of feature items. This means that if the number of feature items is (m), the possible number of combinations of candidate items for optimal features is equal to 2^m. As such, an increase in the number of items in the feature set makes the number of possible combinations grow exponentially. This make searching among all possible combinations of items for selecting optimal features is the main challenge of feature selection strategies (A. L. Blum & Langley, 1997). It is difficult to guarantee that the selected items are the optimal ones without an exhaustive search process of all different combinations of items. There are different selection approaches devoted for searching the optimal or near optimal features that produce acceptable results (Dash & Liu, 1997).

2.4.2 Feature Selection Approaches

Feature selection approaches have been divided into filter methods and wrapper methods (Dash & Liu, 1997). The wrapped feature selection methods conduct a search for optimal or near optimal feature parts. Such methods start by selecting the first parts of features from the original features, where the original features are supposed to consist of big sized of feature parts. The selected feature parts are evaluated in terms of performance by a learning algorithm, which is one of the common classifiers. The wrapper method depends on the results of learning algorithm for selecting feature parts as optimal features, where optimal features are feature parts with the highest score from the evaluation of the learning algorithm. The resulting score from the learning algorithm or classifier is either classification error or classification time (Kohavi & John, 1997). Fig 2.12 shows a wrapper feature selection method, where the learning algorithm and feature selection algorithm wrap together in the same block. The last step is devoted to validation, as the optimal features emerging from the search are tested independently via the learning algorithm.

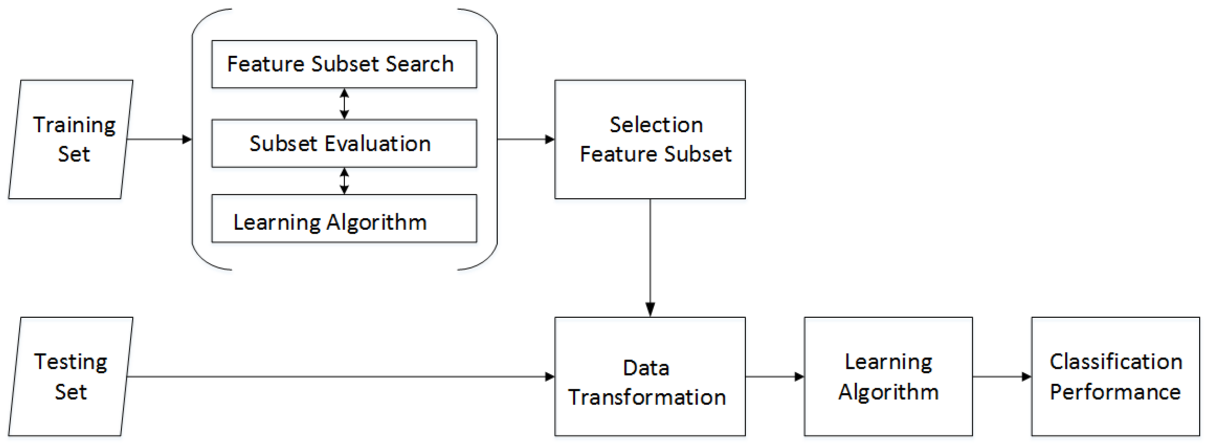


Fig. 2. 12 A wrapper feature selection algorithm.

Filter methods conduct a search for optimal feature parts independently from learning algorithms or classifiers, and without requiring additional information (R. Li & Wang, 2004). In filter methods, the evaluation of selected feature parts is based on measurements that utilise the features themselves. These measurements include distance, information, and consistency measurements (Dash & Liu, 1997). As Fig 2.13 depicts, the evaluation process is conducted on selected features without interfering with or waiting for the results from a learning algorithm. In fact, there is no learning algorithm enveloped within the feature search method in the block diagram. In the filter model, the optimal selected features are also tested independently via the learning algorithm for validation.

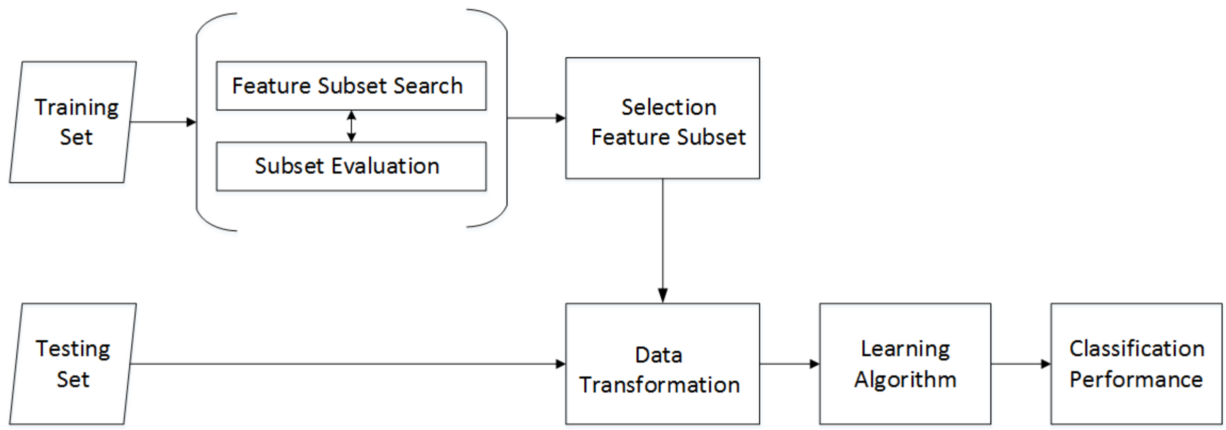


Fig. 2. 13 A filter feature selection algorithm.

For comparison between the wrapper methods and filter methods in terms of seeking optimal features, each technique has its own advantages with respect to the other. The advantage of wrapper selection methods is that the performance of optimal feature parts is usually better than the performance of optimal feature part resulting from filter methods. The reason is that; through the feature searching process, the selected optimal features are evaluated by the learning algorithm. On the other hand, using a learning algorithm in wrapper methods results in some disadvantage compared to filter methods. The wrapper methods are often computationally expensive, the extent of which depends on the learning algorithm used for selecting candidate feature parts (Dash & Liu, 1997; Langley, 1994). Therefore, this makes filter methods less computationally intensive in comparison.

The General Process of Feature Selection Algorithms

Generally, the selection methods from either the wrapper and filter approach follow same process for searching the optimal features parts, as explained in the following steps (A. L. Blum & Langley, 1997; Dash & Liu, 1997; H. Liu & Yu, 2005).

1. Search initialisation:

The start of the search can take one of three states: without features, with the entire features, or with random features part. These options affect the search direction of the algorithm. The first and second states are opposite to one another, as starting with no features leads the algorithm to collect or sequentially add feature parts, which is called forward searching, whereas starting with all features needs to remove parts of the features each time (backward searching). The last state, involves starting the search in the middle point of features and moving out.

2. Search organisation:

This stage involves a number of different search strategies, such as exhaustive search, which guarantees finding the optimal features, and is appropriate when the number of features is not too large. Random or Heuristic search strategies are applied when the number of features is large since they are more practical than exhaustive search, can be used to find the optimal or near optimal features.

3. Evaluation:

Evaluation is done via a function used to determine the importance of the selected parts of features. This stage of the search differs between the wrapper strategy and the filter strategy, where the former needs a classifier for the evaluation process, whereas the latter does not.

4. Stopping the search option:

This step is critical for avoiding exhaustive searching, which is especially important for a huge feature size. There are numerous options for stopping the continue for searching that can be used with the algorithm, such as determining the number of iterations, and the number of repeated features parts in evaluation (Dash & Liu, 1997). These mentioned options can be used for acquiring the optimal feature parts based on predefined evaluation, or indicate when there is no improvement from continuing further iterations, otherwise the algorithm will continue to find other feature parts and compare them with previous best ones.

A validation procedure is applied to the optimal feature parts in order to check of they are valid. The results of validation process are compared with the results of optimal features obtained from the search.

2.4.3 Wrapper Feature Selection

2.4.3.1 Metaheuristics Algorithms

Feature selection using a heuristic approach follows a blind search, which often fails to find the optimal reduction solution, especially in a high dimensionality problem (Brownlee, 2011). These types of methods are more appropriate to deal with a small features dimension (Deogun et al., 1998; K. Hu, Lu, & Shi, 2003).

Alternative methods, called metaheuristic methods, have been developed from natural phenomena, and are less complicated than previous solutions such as heuristic methods. Metaheuristic solutions have emerged as popular methods due to the fact that they result in acceptable solutions for addressing the complexity of different problems (Biswas, Mishra, Tiwari, & Misra, 2013). The metaheuristics algorithm includes a heuristic algorithm strategy, with repeated exploring of the search space, where the algorithm depends on two operations for discovering an optimal solution, which are exploring and exploiting (C. Blum & Roli, 2003).

These types of methods have replaced mathematical methods in solving parameter identification, because of the difficulties involved in dealing with real applications using mathematical methods. The metaheuristic optimization methods are used to identify the possible optimum values of parameters by minimising the difference value between real and numerical data (Talatahari, Mohaggeg, Najafi, & Manafzadeh, 2014). Furthermore, metaheuristic algorithms can be categorised into population-based search or single point search (C. Blum & Roli, 2003).

The population-based methods can offer a number of possible solutions to the problem, which are based on the information of their fitness (objective function) (Dervis Karaboga & Bahriye Akay, 2009). The population algorithms have been invented for optimisation problems, and can be divided into evolutionary algorithms and swarm intelligence algorithms (Karaboga & Basturk, 2007).

2.4.3.1.1 Evolutionary Algorithms

Evolutionary algorithms (EAs) are inspired by biological evolution. These algorithms perform some mechanisms such as reproduction, mutation, recombination, and selection, which are all derived and inspired from biological phenomena. One of the well-known evolutionary algorithms type is Genetic Algorithm (GA) (Holland, 1992). GA, introduced by John Holland, is inspired by natural selection of the Darwinian evolutionary theory (Rahmat-Samii, 2007).

GA solve the problem by starting with generating a population of chromosomes of a fixed-length binary strings, where these chromosomes refer to candidate solutions of the problem. To search for the optimal solution, evolution takes place on these population of chromosomes. During the generation process, the chromosomes with the highest fitness always continue for next stage of population. New generations are produced from performing crossover and mutation operations on chromosomes. These operations apply an exchange process between two parents to create substrings, and to increase the diversity of the offspring. The new solutions, which will continue with the next generation, will be selected by the selection operator (Digalakis & Margaritis, 2002).

2.4.3.1.2 Swarm Intelligence Algorithms

The first use of the expression ‘‘swarm intelligence - (SI)’’ was in the context of cellular robotic systems, which apply self-organising procedures of simple agents through nearest neighbour interactions. Subsequently, this meaning of swarm intelligence was extended to include solving any problem by an algorithm motivated by the behaviour of a collection of insects or animals (Bonabeau, Marco, Dorigo, Théraulaz, & Theraulaz, 1999). Referring to Bonabeau, swarm intelligence refers to ‘‘any attempt to design algorithms or distributed problem-solving devices inspired by the collective behaviour of social insect colonies and other animal societies” (Bonabeau, Dorigo, & Theraulaz, 1999).

One of the important observations of these groups is the ability to self-organisation without a centralised control system. Self-organisation is a set of guidelines that clearly determine communications between different components in the colony (Karaboga, 2005). Another important feature in SI is the division of labour, which guarantees the survival of the colony. The swarm consists of specialised individuals that perform their tasks in a simultaneous manner, which keeps the colony functioning efficiently (Bonabeau, Theraulaz, & Deneubourg, 1996). Collective knowledge among the individuals or agents occur by exchanging information, which determines if the agent has to continue a specific job, or change to another job (Bonabeau et al., 1996; Robinson, 1992).

A swarm is a population of interacting individuals or agents, which aim to optimise global objectives. Such as agents include ants, which inspired the Ant Colony Optimization (ACO) algorithm, birds, which inspired the Particle Swarm Optimization (PSO) algorithm, and Bees, which inspired the Artificial Bee Colony (ABC) algorithm.

(1) Ant Colony Optimisation

The Ant Colony Optimisation **(**ACO) algorithm is inspired by ants' behaviour when they seek food sources, where they attempt to find shortest path from the nest to the food source (Dorigo & Birattari, 2010). During travel, the ants deposit pheromone on the ground which is done in order to facilitate the navigation from the nest to the food source. In future, other ants of the colony follow the same paths based the released pheromone. Over time, the best path becomes the most followed path, as it will have the strongest pheromone due to the larger number of ants travelling through it over time.

(2) Particle Swarm Optimisation

The Particle Swarm Optimisation **(**PSO) algorithm is inspired by the social behaviour of birds flocking (Eberhart & Kennedy, 1995). The algorithm consists of particles, which search for the optimal solution. During the search process, each particle has memory of the best discovered place as of yet, and the best global place. The best global place is obtained through an exchange (or update) of information with other neighbours.

(3) Artificial Bee Colony Optimisation

The Artificial Bee Colony **(**ABC) algorithm was introduced by Karaboga in 2005, and is more recent algorithm than the particle swarm method (Karaboga, 2005). ABC is inspired from the daily behaviour of swarms of honey bees when collecting their food. During the search process, the food sources are candidate solutions. The bee swarm consists of three types of bees, which are employed bees, onlooker bees, and the scout bees. These bees perform efficient division of labour between them through self-organisation when collecting nectar from fields. The colony is divided equally between employed bees and onlooker bees. Every employed bee follows one food source, such that the number of food sources is equal to half the colony. The employed bees exchange information about the quality of food sources, based on the nectar amount, with onlooker bees, through a special dance called a waggle. The waggle gives onlooker bees information about the available food sources, including the direction, distance, and quantity. It is probable that the largest number of onlooker bees will visit the food source with the richest nectar. If the food source is exhausted, the employed bees are converted into scout bees, where they memorise the visited food sources with the highest nectar. In the colony, the employed and the onlooker bees have advanced information about the food sources, whereas the scout bees continue to look for other food sources randomly (Karaboga, 2005; Karaboga & Basturk, 2008). Further details about ABC algorithm is dedicated in next subsection.

2.4.3.2 The Artificial Bee Colony Algorithm

The ABC algorithm utilises a population of artificial bees. The model is adapted based on honey bees searching for food (forage selection). The bees in the model are employed bees, onlooker bees and scout bees, where the algorithm steps are based on these types of bees (for more information refer to (Bansal, Sharma, & Jadon, 2013; Karaboga & Basturk, 2008; Ozkan, Ozturk, Sunar, & Karaboga, 2011)).

In the ABC algorithm, the colony is divided equally between the employed bees and the onlooker bees, where the number of employed bees is equal to number of onlooker bees, which are equal to the number of food sources or solutions ($p = 1,2, \dots \dots, n_{Eb}$), where p is the population size, and n_{Eb} is number of employed bees.

Specific number of food sources are selected using equation (2-16). The food sources are considered as the population, and are selected randomly based on a numbers of parameters to be optimized ($d = 1,2, \dots \dots n_p$), where d is the dimension vector, which contains the number of parameters (n_p).

$$x_i^{jselected} = x_i^{jmin} + rand(u)\left(x_i^{jmax} - x_i^{jmin}\right) \tag{2-16}$$

where $x_i^{j\,selected}$ is the selected solution from j parameters out of i solutions, $x_i^{jmax} and\, x_i^{jmin}$ are the maximum and minimum value of the parameters (j) of solution number (i), respectively, and (u) here is [0,1].

The initial food sources are changed depending on the quantity of nectar existing at every source. The employed bees start a local search in adjoining areas of discovered initial places of solutions. This process is described by equation (2-17). The places in neighbourhood will be memorised instead of previous places if they contain a higher amount of nectar.

$$x_{ij}^{modified} = x_{ij}^{previous} + rand(v)(x_{ij}^{previous} - x_{kj}^{neighboring}) \qquad (2\text{-}17)$$

where, $x_{ij}^{modified}$ is the ith selected modified solution of the jth parameter, $x_{kj}^{neighbor}$ is the neighbouring parameter value, with a space equal to k, from the previous selected parameter value $\left(x_{ij}^{previous}\right)$, where $k \neq i$ $and\ both \in \{1,2, \ldots\ldots, n_{Eb}\}$, and v is $[-1,1]$.

When the employed bees complete the search, they share the information about the discovered places of nectar with onlooker bees, which wait in in the hive area, through a special dancing routine. The onlookers' role is to evaluate the quantity of nectar in the discovered places to choose the place with highest probability for optimum solution (equation (2-18)).

$$\Pr(i) = \frac{\mathrm{f}(i)}{\sum_{i=1}^{n_{Eb}} \mathrm{f}(i)} \qquad (2\text{-}18)$$

where $\Pr(i)$ is the probability of fitness $\mathrm{f}(i)$ of solution i, and n_{Eb} is the total number of employed bees or food source places.

Subsequently, the onlookers apply the same employed bees' procedure on elected places by probability to deduce the place with the highest nectar amount. Onlookers search in neighbourhood of such place, which will be replaced if a new place containing a higher amount of nectar than the previous place is found.

This sequence is repeated until one of search places is exhausted. The employed bees in the searched places become scouts, where they repeat exploring other places randomly (following equation (2-16)), which may possibly be better than previously known places.

This procedure can be employed for feature selection, where the bees select a number of feature parts randomly, and calculate the fitness for these parts to find the best feature part in each iteration.

This previous process can be controlled by the number of iterations, which is based on the size of features, and guarantees obtaining the optimal feature parts.

2.4.3.3 ABC-based Wrapper Feature Selection

Numerous evolutionary and swarm algorithms have been applied for different applications. Since the wrapper approach for feature selection is involved in this work through the ABC algorithm, a review of related work to the ABC method is presented in this subsection. This section justifies the preference of ABC to other wrapper optimization algorithms. It starts by comparing ABC to other population optimization algorithms, and justifies the selection of the ABC method as the proposed method for feature selection. The section then reviews a number of different applications, where ABC algorithm is used as an optimization method.

(1) ABC Algorithm Compared to Other Optimization Algorithms

Karaboga and Basturk (2007) compared ABC with the most well-known population evolutionary and population swarm optimization algorithms, which are the Genetic Algorithm (GA), the Particle Swarm Algorithm (PSO), and the hybrid Particle Swarm Inspired Evolutionary Algorithm (PS-EA). The aforementioned algorithms were tested using a benchmark based on high dimensional numerical functions. The main purpose was to evaluate the performance of ABC against these algorithms. In general testing, the ABC algorithm outperformed others by obtaining a local minimum, with the ability to deal with multivariable, multimodal function optimisations.

In another study, Karaboga and Basturk (2008) evaluated the performance of the ABC algorithm against other previous optimisation algorithms. The ABC algorithm outperformed other algorithms, which are the differential evolution (DE) particle swarm optimization (PSO) algorithm, and the evolutionary algorithm (EA). The testing was conducted on multi-dimensional numerical problems. The comparison was applied on different control parameter values, such as the population size (colony size) and limit values. The evaluation results showed that the performance of ABC was better than the aforementioned algorithms, and that the ABC algorithm has the ability to be used with high dimensional engineering problems.

A. Singh (2009) applied ABC as new optimization method for Leaf-Constrained Minimum Spanning Tree (LCMST), and compared it with other population approaches such as the genetic algorithm, the ant-colony optimization approach, and the Tabu search (TS) approach. The average solution of ABC reported superior results to others, except in one case, where the Tabu search approach provided average solutions of better quality. In addition, the execution time of ABC was faster than others. Thus ABC-LCMST achieved a better performance than others in terms of both, results quality and execution time.

Tahooneh and Ziarati (2011) adapted ABC to deal with the Resource Constrained Project Scheduling (RCPSP) problem. In the literature, other optimisation methods such as the genetic algorithm have been used for this problem to reduce the project duration, where choice is based randomly, as it is difficult to know in advance. ABC demonstrated its ability to provide better diversity and improve the quality of solutions, thus proving that it was more efficient than other algorithms for addressing this problem.

Gozde, Taplamacioglu, and Kocaarslan (2012) proposed ABC as an optimisation algorithm for tuning parameters of PI and PID controllers that are used in thermal power system for Automatic Generation Control (AGC). Furthermore, the authors compared the performance of ABC with the Particle Swarm Optimization (PSO) approach. The results showed that ABC was more effective than PSO for AGC problems.

Z. Deng, Gu, Feng, and Shu (2011) compared the performance of ABC with the genetic algorithm and the Max-Min Ant System (MMAS) in energy-aware mapping optimisation in Network-on-Chip (NoC) designs. The results showed that ABC performed better than GA and MMAS. Furthermore, the ABC results recorded lower energy consumption with a higher convergence rate.

Atasever, Özkan, and Sunar (2011) applied a number of optimization methods with unsupervised classification for remote sensing of images. The applied optimization methods were Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) and Artificial Bee Colony (ABC). Previous optimization methods relied on manually determining the number of clusters centres of K-means (KM) and Fuzzy C-means (FCM). In comparison, the ABC method, which is most modern optimization method, is more stable with initial conditions and is not influenced by changes in the parameters values.

(2) ABC Algorithm for Optimized Applications

ABC has recently been utilized in different optimization applications, where some of these applications were based on feature selection problems. Generally, in these applications, the ABC demonstrated effective performance.

Y. Zhang, Wu, and Wang (2011) applied hybrid forward neural network (FNN) and modified ABC for brain image classification into as normal or abnormal classes, where the images had been produced by magnetic resonance (MR). The ABC method was modified by fitness scaling and chaotic theory and was used for optimizing the parameters of FNN. Discrete wavelet transform was used to extract the features from the brain images before such features were reduced by the principle component analysis (PCA). The modified ABC was also shown to outperform other methods in optimizing FNN, such as the genetic algorithm, simulated annealing, and ABC itself.

Sathya and Geetha (2013) used the ABC algorithm to optimize a three-layer neural network for breast cancer classification. The main purpose was to improve the classification of the images based on the disease into malignant and benign lesions at the early stage of disease. The optimized neural network classifier based on ABC resulted in an improved detection accuracy of seven features, which were extracted from the target images of benign or malignant lesions.

Uzer, Yilmaz, and Inan (2013) used ABC for feature selection and to avoid redundant features from liver images. The selected features were sent into SVM for diagnosing diabetes in the liver. The results of testing based on UCI database yielded a classification accuracy of 94.92% for hepatitis, 74.81% for disorders, and 79.29% for diabetes.

Banerjee, Bharadwaj, Gupta, and Panchal (2012) used the most recent optimization method of ABC with Remote Sensing classification. The purpose of using ABC was to improve the image classification rate from satellite data of the earth land. The results from this application were compared with other techniques such as the biogeography-based optimization (BBO), maximum likelihood classifier (MLC), minimum distance classifier (MDC), and Fuzzy classifier. The results proved the suitability of ABC as new optimization method for this type of application.

Jayanth, Koliwad, and Kumar (2015) also applied ABC in a new method based on combining spectral variance with spatial distribution of the pixels in satellite images of earth land. Previous methods were, in comparison, based only on spectral variance. ABC was applied for extracting the features and to avoid the issues related to band correlation. The results of multi-class classification was improved by different classifiers such as Artificial Neural Network and Support Vector Machine.

Schiezaro and Pedrini (2013) proposed a selection algorithm based on wrapper ABC and SVM for feature selection based on UCI datasets. The results, based on Weka programming, showed the effectiveness of this method in removing irrelevant features by returning the lowest number of features. Furthermore, the method was compared against other relevant approaches, where the experimental results showed that the algorithm produced better performance than ACO, PSO and

GA. The ABC algorithm outperformed others in the context of least selected feature size, but with a slightly reduced accuracy.

Shanthi and Bhaskaran (2014) also proposed a wrapper algorithm for feature selection, which used a modified ABC algorithm and a Self-adaptive Resource Allocation Network (SRAN). The method was used for medical diagnosis to discover the prior signs of breast cancer. The selection was based on different features methods, which participated in detecting a set of features from a mammography image. The features were extracted by Gabor filters, fractal analysis, directional analysis, and the multiscale surrounding region dependence method. Detecting abnormality of mammogram images using a huge number of features was difficult, so a Modified Artificial Bee Colony based Feature Selection (MABCFS) was proposed to select the relevant features. The results of classification were improved by MABCFS when compared with previous selection methods (GA and PSO), as it recorded the best overall results.

Mohammadi and Abadeh (2014) proposed ABC as a wrapper selection method, which was applied with classifiers such as the Naïve Bayes (NB) classifier and the k-Nearest Neighbour (KNN) classifier, to evaluate feature subsets for steganalysis. The method was used for collecting features to experiment with hidden messages, which consisted of images, videos and audios files. The selected features were used for the steganalysis problem through detecting stego images used as secret messages. The applied method got superior results over GA, which had been used before for the same problem.

Duan, Deng, Wang, and Xu (2013) applied ABC for filtered feature selection as a way to remove noise. The method was used to reduce calculations when the selected regions contained salient objects. The results of this method were compared with PSO results, demonstrating that ABC achieved a superior performance to PSO in terms of the classification accuracy.

The previous studies highlight and justify the predilection for the ABC algorithm to be adopted in this research as a means for optimizing feature extraction methods in texture-based image classification applications.

2.4.4 Wrapper-based Filter Feature Selection

In Section 2.4.2, feature selection methods were categorised into two broad techniques. Generally, wrapper methods produce better results in feature selection applications than filter methods. However, wrapper methods are more demanding in terms of computation than filter methods. This is because evaluation is carried out using a learning algorithm for each time the method selects a candidate part of the relevant feature, which increases the computational cost. Recently, numerous hybrid approaches have been proposed by researchers, which look for advantages in filter and wrapper methods. Rough Set (RS) is one of the common filter methods that has been applied effectively for feature selection. This section presents a review of the work involving combining RS with other wrapper methods. The concepts of RS theory and Neighbourhood Rough Set are then introduced, where the latter is employed in continuous data.

2.4.4.1 Rough Set Theory-based Approaches

Many previous researchers have introduced several approaches of feature selection by RS algorithms, either in single operation, or combined with swarm algorithms.

Q. Wang, Li, and Liu (2008) investigated the effectiveness of the RS theory in identifying important features of texture for classification applications. The RS theory was used to select the features resulting from applying a wavelet packet. Empirical results were based on thirteen datasets from Brodatz textures. The results indicated that removing the redundancy of features by the RS had an effect on improving the classification performance due to the fewer number of features used.

H. Lin, Wang, and Liu (2010) presented a novel RS feature selection method for classification. The method was used to find the relevant features from texture, which were extracted by the Gray level Co-occurrence Matrix (GLCM). The reduced features were used for classification of images related to mine rules. The testing applied to images containing a decorative stone.

In (T.-S. Li, 2009), the author proposed a feature selection method based on rough sets, wavelet transform, and the SVM classifier, where the method was used for selecting the optimal parts of features. The RS was used for selecting the features. The extracted features, which were based on a number of wavelet decomposition levels, were obtained from smooth sub-image of homogeneous Copper Clad Laminate (CCL) surfaces. Experimental results demonstrated the efficiency of the proposed method in improving classification quality, especially when the results were compared with the BPNN classifier.

There are a variety of hybrid methods that been developed based on swarm algorithms with rough sets for feature selection applications. In (Xiangyang Wang, Yang, Teng, Xia, & Jensen, 2007), the author presented a hybrid filter with wrapper feature selection method using PSO and the RS algorithm. The RS strategy was based on completing a search for an optimal solution, which is only appropriate for small-sized datasets, as such an approach does not otherwise guarantee finding the optimal solution. PSO is a promising selection method for RS reduction because it works within the subset space. The experiment was carried out on number of datasets from UCI, and the results were compared with GA. The results of the hybrid method showed that the computation was less expensive than GA, and that the results were better than the RS reduction algorithm.

For the same purpose, Ke, Feng, and Ren (2008) used the Ant Colony Optimization (ACO) algorithm with RS theory (resulting in the ACORS algorithm) to avoid the NP-hard problem. The experiment used to prove the effectiveness of the ACORS algorithm was based on different sized numerical datasets with other gene expression datasets. The accuracy of the proposed method was reasonable, where it achieved slightly inferior performance than previous methods.

Y. Chen, Miao, and Wang (2010) also proposed a hybrid approach between the ACO algorithm and RS. ACO has the ability to quickly converge, which justified its use with RS for finding the optimal solution for the same problem. However, in this hybrid method, the ACO algorithm was based on resulted feature from RS as random feature, then continued searching for optimal features through the rest of feature dimensionality. UCI datasets were used in the experiment, where the results proved that the method has the ability find a minimal subset of the features when compared with using the RS method alone.

In (Bae, Yeh, Chung, & Liu, 2010), for a large number of attributes, the authors proposed the Intelligent Dynamic Swarm (IDS) approach, which was used to convert a problem of discrete variables to continuous variables. IDS was combined with RS to produce the IDSRSFS algorithm, which was used for improving the performance of feature selection applications. In the experiment, the algorithm was tested on different UCI datasets, and compared with the hybrid PSO and RS (PSORSFS) approaches. The results showed that IDS was faster than PSO in finding the minimal reductions of features, due to its low computation cost.

2.4.4.2 Rough Set Theory

The Rough Set Theory (RST) is one of the filter methods introduced by Pawlak (Pawlak, 1982). The RS approach has been proven to be an efficient feature selection tool. The general characteristics of the data are used to evaluate the selected subset in filter methods based on certain statistical criteria. The RS method processes the uncertainty and vagueness of certain data mathematically, without additional knowledge. It achieves this by extracting the dependency rules directly from the data itself, to obtain further information about the data.

The data in information systems is distributed as a table which is categorised into universe and attributes, such that $IS = (U, A)$. The rows represent the universe (U), which consists of a set of objects, whereas the columns represent the attributes (A). The decision table is generated by dividing the attributes in the information table into condition attributes and decision attributes, to become $IS = (U, C\ U\ D)$ (M. Zhang & Yao, 2004).

Feature reduction is one of the basic tasks of the rough sets theory beside classification. It is effectively exploited in selecting the number of attributes contained in a dataset through the dependencies of data. Whilst reducing the number of attributes, it is adequate for retaining the unique properties of the decision table (Yao & Zhao, 2008).

The reduction in the RS method is based on the granularity structure of the data as a result of indiscernibility, which is the main concept in RS theory. Indiscernibility occurs between objects if a number of them have the same information, which is referred to as equivalence or an elementary set.

If the B subset is an attribute of A, that means $B \subseteq A$ and $a \in A$, and the equivalence relation (R-indiscernibility) can be expressed as follows:

$$IND(B) = \{(i, j) \in U^2, \forall_a \in B, a(i) = a(j)\} \qquad (2\text{-}19)$$

where $a(i)$ is the attribute value a of object i, and $a(j)$ is the attribute value a of object j.

When applying the indiscernibility on the universe, the objects will be split into groups called elementary sets. If $(i, j) \in IND(B)$, i and j are said to be indiscernible with respect to B. The indiscernibility splits the objects into a family of equivalence classes, and if all objects in the group are equivalence classes of $IND(B)$, they are denoted by $U/\ IND(B)$.

As a result of indiscernibility, the attributes are approximated based on their relevance, and divided into a pair of sets, called lower approximation and upper approximation.

The lower approximation $B_{-}(t)$ is defined as follows:

$$B_{-}(t) = \{t \in U : B(t) \subseteq T\} \qquad (2\text{-}20)$$

The lower approximation of T is the set of elements of U that are absolutely classified as elements of T.

The upper approximation $B^{-}(t)$ is defined as follows:

$$B^{-}(t) = \{t \in U: B(t) \cap T \neq \emptyset\} \tag{2-21}$$

The upper approximation of T is the set of elements of U that are probably classified as belonging to the set T.

If $P, Q \subseteq A$ are equivalence relations over U, the method defines three regions, which are positive, boundary, and negative regions as follows.

The positive region $POS_p(Q)$, includes the objects of U which definitely belong to the relevant set of features.

$$POS_p(Q) = \cup_{x\epsilon \mathbf{U|Q}} B_{-}(t) \tag{2-22}$$

The negative region $NEG_p(Q)$, is the set of all objects of U that cannot belong into a relevant set of features.

$$NEG_p(Q) = U - \cup_{x\epsilon \mathbf{U|Q}} B^{-}(t) \tag{2-23}$$

The boundary region ($BN_s\ (t)$) is the distance between the upper and lower approximations, which holds the objects that are probably relevant features.

$$BN_s\ (t) = B^{-}(t) - B_{-}(t) \tag{2-24}$$

Based on the boundary region, the crisp state occurs when this region is empty ($B^{-}(t) = B_{-}(t)$), whereas the opposite case, which is the rough state, occurs when the region is not empty ($BN_s\ (t) \neq \emptyset$) (M. Zhang & Yao, 2004).

The degree of dependency (k) is used for discovering dependencies between attributes, which are either totally, partially or not dependent.

For $\mathrm{P}, \mathrm{Q} \subset \mathrm{A}$, the degree of dependency is calculate by following relation:

$$\mathrm{k} = \gamma_p(\mathrm{Q}) = \frac{|POS_p(Q)|}{|U|} \tag{2-25}$$

The dependence is partial when some value from attributes P depends on other attribute values of Q $(\mathrm{P} \rightarrow \mathrm{Q})$ in a degree $(0 \leq k \leq 1)$. The dependence is total between two attributes

(P → Q), when all values of attribute Q are uniquely determined by values of attributes P, which means ($k = 1$). Otherwise, ($k = 0$) means that Q does not depend on P .

The reduction is applied on the set of attributes after comparison based on equivalence relations is conducted between them. The quality of classification is expected to be the same before and after removing attributes as a result of equivalence.

From the decision table, which consists of set of condition attribute C and a set of decision attribute D, the equality of approximation of classification γ_R(D) determines the degree of dependency between condition and decision features, which result from the set of decision features (Pawlak, 1997).

A reduction is defined as a subset R of the conditional attribute set C, such that ($R \subseteq C$) if γ_R(D) = γ_C(D):

$$Red = \{R \subseteq C \mid \gamma_R(\mathrm{D}) = \gamma_C(\mathrm{D})\} \tag{2-26}$$

The RS model has proved its benefits in different applications with a wide range of techniques. RS is appropriate with discrete data, whereas Neighbourhood RS was developed based on RS for continuous data.

2.4.4.3 Neighbourhood Rough Set Theory

Depending on the equivalence relation of attributes, RS may not be suitable for continuous datasets. The Neighbourhood Rough Set (NRS) is used to deal with continuous data by distance function, which replace the equivalence approximation of RS. RS is only appropriate with for reduce discrete data. NRS substitutes equivalence spaces with topological spaces, which is more suitable for continuous data of real-world applications.

Neighbourhood Decision System (NDS)

In RS, the decision system is DS=$(U, C \cup D)$, where U is the universe and $A = C \cup D$, where C represents the sets of conditional attributes, and D represents the decision attributes. On the other hand, in a neighbourhood decision system based on $\theta(\theta > 0)$, the decision attributes will generate θ-neighborhood, and the result is NDS =$(U, C \cup D, \theta)$ (Q. Hu, Yu, Liu, & Wu, 2008).

Assuming $x_i \in U$ and $B \subseteq C$, the neighbourhood $\theta_B(x_i)$ of x_i in the subspace B is defined as:

$$\theta_B(x_i) = \{x_i \in U, x | f(x_i, x) \leq \theta\} \tag{2-27}$$

where f is a metric function, and $x_1, x_2, x_3 \in U$. f satisfies:

$\text{Non} - \text{negative}: f(x_1, x_2) \geq 0;$

$f(x_1, x_2) = 0; if\ and\ only\ if\ x_1 = x_2;$

$Symmetry: f(x_1, x_2) = f(x_2, x_1);$

$Triangle\ inequality: f(x_1, x_3) \leq f(x_1, x_2) + f(x_2, x_3).$

The metric distance function can be applied by Euclidean distance equation (2-24) for a θ-neighborhood relation between samples (xi), which replaces the equivalence approximation in RS, and makes NRS capable of dealing with continuous data.

$$f(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_i - x_j)} \tag{2-28}$$

where $\theta_{(xi)}$ is the neighborhood information granule cantered around sample xi, and the size of the neighbourhood depends on the threshold θ. Larger values of threshold (θ) mean more samples exist in the neighbourhood region, whereas $(\theta = 0)$ refers to a case that is applicable to discrete data.

With the same threshold θ, the sizes of neighbourhoods with different norms are different, and we thus have (Q. Hu, Yu, & Xie, 2008):

$$\theta_1(x) \subseteq \theta_2(x) \subseteq \theta_\infty(x) \tag{2-29}$$

Neighbourhood Approximation Regions

In $NDS = (U, C \cup D, \theta)$, B is a subset of C $(B \subseteq C)$ for an arbitrary $X \subseteq U$ construct, where the granules of lower approximation and upper approximation of X in terms of the relation of N with respect to B are as follows:

$$N_{B\_}(X) = \{x_i | \theta_B(x_i) \subseteq X, x_i \in U\} \tag{2-30}$$

$$N_{\mathrm{B}}^{-}(X) = \{x_{\mathrm{i}} | \theta_{\mathrm{B}}(x_{\mathrm{i}}) \cap \mathrm{X} \neq \emptyset, x_{\mathrm{i}} \in U\} \quad (2\text{-}31)$$

The lower approximation is also called the positive region of the decision, denoted by $POS_B(D)$, and the granules in this region consistently belong to one of the decision classes (Q. Hu, Yu, & Xie, 2008).

The boundary region of D with respect to attributes B is defined as:

$$BNR\ (D) = N_{\mathrm{B}}^{-}(X) - N_{\mathrm{B}\_}(X) \quad (2\text{-}32)$$

For the set x in the approximation space, the degree of roughness is represented by the size of the boundary region. It includes the set of samples which belong to different decision classes (uncertainty belong to any class). The boundary region controls the uncertainty decision by reducing it as little as possible. These set of samples cause classification confusion, because they cannot be determinately classified. The reduction aims to reduce the set of samples in this region in the decision-making process (Q. Hu, Yu, & Xie, 2008).

2.5 Challenges Identification and Proposed Solutions

This section summarises the main challenges facing this research and texture-based classification in general. This research studies texture features using feature descriptors, feature-level fusion, and feature selection.

2.5.1 Effective Texture Descriptors

Feature descriptors based on TUs proved to be more efficient than depending on a pair of pixels using co-occurrence methods. TS was the first method to be based on TUs. However, in addition to its long feature histogram, it has an unbiased quantisation function, as it assigns a binary one for only the intensity values of neighbour pixels that equal to the centre pixel. LBP addresses that by applying binary quantisation to reduce the feature length resulting from the histogram. LBP demonstrated its simplicity and power in comparison to previous statistical methods, as LBP features have been applied successfully in different texture applications. However, the LBP method may lose significant information from the intensity values of neighbourhood pixels. We think that despite the discriminative features of LBP, LBP is used more for its simplicity than for its capability to collect information with binary quantisation of intensity values of neighbour pixels. Other applications needed to extend LBP into LTPs, and subsequently into LQPs, where these extension methods capture better features from the texture. The aim always is to collect as much as information from the intensity values of neighbour pixels as possible. Here, we target the development of new descriptors for extracting local features from TUs to improve local texture features, which affect the performance of textures classification. The proposed strategies concentrate on exploiting grey-level intensity values of texture patterns. This can be done by well-designed quantisation functions, which can be utilised for discriminating between different texture patterns.

2.5.2 Curse of Dimensionality

There is a wide diversity of texture characteristics, which make depending on the single method, are usually not enough for dealing with texture. In order to obtain more distinctive and powerful features, features from different descriptors can be combined. These features should be complementary to capture good information from the texture. This was explained in Section 2.3, where a discussion was presented on how different descriptors techniques can be combined to represent features. LBP is one of the most common texture descriptors, which can be improved when combined with other complementary feature descriptors such as contrast measurements and GF. The most common problem with combining feature descriptors is the resulting large feature space, or what is referred to as the curse of dimensionality. This problem can be addressed by feature selection, which reduces the feature length.

2.5.3 Efficient Selection of the Relevant Features

The unified feature vector from different descriptors usually contains irrelevant or redundant features, where part of the relevant features of any descriptor may become irrelevant when combined with other features. In addition, the curse of dimensionality usually happens after combining different feature descriptors, as this often results in a large feature space. These problems have negative effect on the overall classification performance in terms of computation and classification accuracy.

Referring to Subsection (2.4.1), the problem of dimensionality reduction can be addressed by feature selection or feature transformation. One of the most common methods used for reducing the feature length of LBP is through the principle component analysis (PCA) (Tan & Triggs, 2007). However, PCA is a transformation method which reduces the feature length by changing the semantics of features. This is considered a drawback, as many applications need to save the meaning of features throughout the reduction operation (Shang & Shen, 2008). The rough set method (Subsection 2.4.4.2) is also utilised for feature selection, but offers to retain the meaning of features (Pereira & Sassi, 2012).

GA is also used for feature selection, and is one of the most common wrapper methods. The method has been successful in many optimisation applications that involved with GF, as explained in Subsection 2.2.2.2. However, the Artificial Bee Colony (ABC), introduced in Subsection 2.4.3.3, is an alternative and more recent wrapper method than GA (Karaboga & Basturk, 2008). The ABC method is favoured over other optimization algorithms for performing feature selection (Dervis Karaboga & Bahriye Akay, 2009; Krishnanand, Nayak, Panigrahi, & Rout, 2009). One important advantage that can be obtained from applying the ABC algorithm is its flexibility, where the algorithm can be easily adapted according to the particular problem, which makes it suitable for various optimization problems.

Furthermore, while GA can be used to optimize GF parameters in many applications, GA depends on complex evolution processes by crossover and mutation operations (Gheyas & Smith, 2010). On the other hand, the proposed optimisation method (ABC) depends on simple mathematical operations (Dervis Karaboga & Bahriye Akay, 2009).

Finally, ABC, with its simplicity, follows an effective strategy that makes it converge more quickly to the optimal solution. ABC conducts two local search processes before converting into a global search when local searching is exhausted. All of these processes are executed in one cycle with few parameters, and provide good results even with high dimensional operations (D Karaboga & B Akay, 2009).

The literature clearly shows that wrapper methods like ABC are more accurate for features selection applications. However, these methods require long computation time for evaluating the selected features by a learning algorithm. On the other hand, complete or exhaustive searching using RS is not effective with a large feature space (Ke et al., 2008; Xiangyang Wang et al., 2007). To obtain better results, it is thus recommended to utilise hybrid methods based on different techniques, as they are more efficient for feature selection applications. Exploiting the advantages of ABC and NRS for a proposed feature application may help in selecting the optimum features.

The main aim of this research is to tackle the aforementioned problems and challenges by obtaining optimum features that can improve texture-based classification. Our proposed approach accomplishes this by developing descriptors that are integrated with complementary features. Furthermore, adaptive hybrid selection methods are used to avoid the dimensionality problem that can result from integrating feature descriptors.

2.6 Summary

This chapter studied in detail the background of texture as an important source of feature for images classification, and provided a review of the most important research studies related to this topic. The chapter was organised as follows.

Section 2.1 presented an overview of texture characteristics, analysis methods of texture, and texture applications. Any procedure used for improving feature extraction can directly impact the overall efficiency and accuracy of images classification by texture. The following sections then concentrated on the important issues associated with texture features, which included features descriptors, features fusion and feature selection.

Section 2.2 introduced the common feature extraction methods, which were categorised into different groups. For random texture, which mostly exists in natural images, statistical and signal processing methods are more appropriate approaches. Among the various relevant feature extraction methods, LBP proved its simplicity and efficiency in extracting features from the texture units of an image. However, LBP does not always extract important information from different texture characteristics. This must be taken into consideration when attempting to realise better texture descriptors.

In Section 2.3, an overview of feature-level fusion and its importance for improving extracted feature was provided. Work related to feature-level fusion was also reviewed. The feature-level fusion stage is usually required because single feature descriptors are not adequate due to the diversity of texture characteristics. Thus, fusion between different feature descriptors can improve the feature.

In Section 2.4, the importance and major problems of feature selection were clarified. The approaches used for feature selection were investigated, the work associated with different feature selection algorithms was reviewed. Feature length reduction techniques are typically applied to reduce the high feature dimensions. Thus, the efficient selection of the relevant features is an important stage after fusing the different features.

Finally, Section 2.5 highlighted the major problems and challenges that face the research in improving texture-based feature for images classification. The challenges included texture descriptors, feature-level fusion, and feature selection methods. Based on previous studies, some of interesting points that have been established include:

- Common texture descriptors do not usually effectively exploit information from texture patterns.
- Most strategies are based on finding complementary features that can improve the performance of a single feature descriptor. However, such strategies usually suffer from the dimensionality problem.
- The curse of dimensionality is a critical problem, and different strategies have been applied to reduce the feature length. However, current strategies usually deal with this problem inefficiently.

Chapter 3

Research Methodology

Extracting powerful features from texture is a significant challenge for images classification. The last chapter summarised the limitations and challenges associated with texture feature extraction for images classification. It also discussed the proposed possible solutions to improve texture features. This chapter introduces the research methodology of this study, including the main stages of the classification system, and the employed methods to accomplish the research objectives.

The proposed methods for improving the texture features consist of new descriptors, termed LBZP and LMP. These descriptors are developed based on the LBP concept to extract distinctive features from TUs. As textures classification gets more challenging with the diversity of texture characteristics, the proposed feature descriptors are recommended to be combined with other complementary features descriptors. However, the common problem that faces feature-level fusion is either irrelevant features, or the resulting huge feature space, where the last call ‘‘curse of dimensionality’’. In our methodology, the feature space is reduced through a new hybrid selection method of ABC and NRS.

This Chapter 3 is organised as follows. Section 3.1 introduces the classification system methods required in our methodology, while Section 3.2 discusses the pre-processing stage of feature extraction, which includes preparing texture images of datasets. Section 3.3 provides full details about the proposed methods for new improved features, while Section 3.4 discusses the post-processing methods performed on the resulting improved feature. Finally, Section 3.5 concludes the chapter by providing a summary of the proposed methods for texture features.

3.1 Employed Methods

Here, an introduction to the employed methods used to improve features extraction in classification systems is presented. The complete classification system process is divided into pre-feature processing, feature processing and post-feature processing (see Fig 3.1).

Pre-processing of features includes preparing image classes from the selected databases to be used as a benchmark for the proposed method. On the other hand, feature processing involves the use of the employed methods for improving the feature. The methods used in this process are either LZBP- or LMP-based methods. The new hybrid feature selection method based ABC and NRS are then employed as a way to fuse feature descriptors. The feature level fusion is done between the proposed local features descriptors (LZBP and LMP) and other complementary feature descriptors (contrast measure and GF). Finally, the post-processing of features includes the classification of images by supervision classifiers, and the evaluation of classification results.

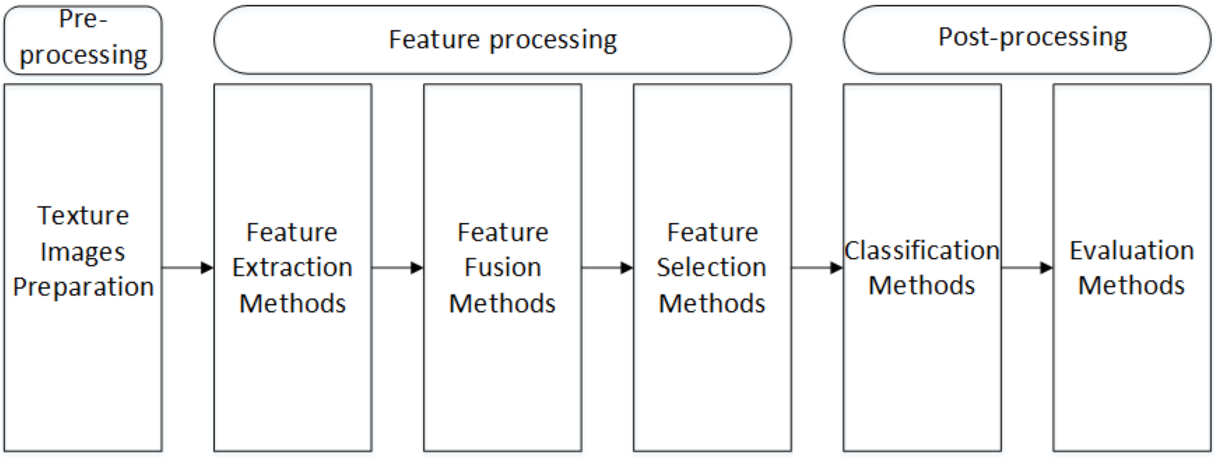


Fig. 3. 1 The role of the employed methods used for improved feature detection in the overall classification system.

3.2 Feature Pre-processing

The datasets, which reflect the characteristics of textures, are processed by texture descriptors in order to extract the features to be classified by the classification algorithms. The utilised datasets in this work should be designed to be suitable for the classification task. These databases consist of a stationary texture image, where the image contains only a single type of texture. The texture covers the entire image, and the extracted feature by the descriptors are supposed to stem from the complete texture characteristics. The image samples used in this stage are different to the images used for segmentation task, where there are two or more types of texture in the same image, which is called a non-stationary texture image (Petrou & Sevilla, 2006).

Texture databases are usually available from various websites (Lazebnik, Schmid, & Ponce, 2005; Mallikarjuna et al., 2006; Xu, Ji, & Fermüller, 2009). In multiclass classification, the involved databases consist of a number of classes, and each particular class has number of images. The database should have sufficient number of samples for supervised classifier, so that the results can be assessed and evaluated (Ojala, Maenpaa, et al., 2002). In the pre-processing stage, a sufficient number of images can be obtained through dividing the images into sub images. Most of the databases that currently exist do not have the demanded number of images for classification. For example, the Brodatz database has one image of high resolution for each class (Brodatz, 1966). The procedure used with these datasets thus involves dividing the target image into number of (n) images, before randomly selecting the required number of images to be used by the feature extraction method. Random selection guarantees an unbiased process in the testing and evaluation stage of the feature extraction methods. The choice of the format of sub-images is also made. Instead of applying this procedure manually with target databases, a method has been developed to perform this task efficiently. In the end, the processed images are saved in specific folders, and can called by referring to the path that indicates their locations.

3.3 Feature Processing

In the texture classification system, it is crucial to extract features that can be used to represent the texture efficiently. This makes the feature processing stage a crucial one, as it involves improving the discriminative characteristics of the features extracted. In first stage of feature processing, the LZBP and LMP descriptors, which are developed based on the basic LBP method, are used. The second stage involves integrating the new feature descriptors with other complementary feature descriptors in order to improve the diversity of features extracted. To avoid the dimensionality problem arising from integrating complementary feature descriptors, a new hybrid feature selection approach is introduced based on the artificial bee colony and the neighbourhood rough set algorithms.

3.3.1 Local Features Extraction from Texture Patterns

3.3.1.1 TUs mapped for Integer Values

One of the most effective and representative methods of extracting textural materials is the local binary pattern (LBP) method (Ojala et al., 1994; Ojala et al., 1996). LBP has achieved extensive success and widespread utilisation among other popular texture feature descriptors (Brahnam, Jain, Nanni, & Lumini, 2014; Pietikäinen et al., 2011).

As explained in Section (2.2.1.5), the LBP-based procedure starts with using binary quantisation of grey-level intensity values of the texture pattern in each TU, based on equation (2-7). Then, Equation (2-8) is utilised for weighting the results from the binary quantisation of the texture pattern. The sum of weighted (0, 1) of texture pattern is subsequently uniquely mapped to an integer value. This procedure guarantees to associate each unique texture pattern of different samples of TUs with an integer value.

TS is the first method utilised the uniquely map the texture pattern of TU to an integer value (He & Wang, 1990). TS is carried out by ternary quantisation of grey-level intensity values for texture patterns in each TU, following Equation (2-5). Equation (2-6) is then used for weighting the results from ternary quantisation of the texture pattern (0, 1, 2). However, TS fails to be an efficient quantisation function of texture patterns due to its dependency on ternary coding.

Compared to TS, the LBP method achieves better results in the context of texture classification through binary quantization using Equation (2-7), followed by Equation (2-8) for weighting the results from the binary quantization of TUs (0, 1). Furthermore, using Equation (2-9) and (2-10), LBP can be extended into the LTP and LQP methods respectively to obtain more information from texture patterns. However, due to its utilisation of simple binary quantisation, the LBP usually fails to recognise important texture patterns that may produce significant information from TUs.

Applying binary quantisation for texture patterns in TUs usually results in a difficulty in distinguishing between different intensity values of texture patterns. For example, in a TU where neighbourhood intensity values are larger than the threshold value, majorly different values are both converted into code 1 as a result of binary quantisation of LBP.

3.3.1.2 Discriminative Texture Descriptors Strategies

Different strategies have been introduced for extracting features from TUs. The main purpose of this process is to gather as much as information from the texture patterns as possible, by distinctive recognition of the intensity values of the texture patterns.

Motivated by LBP and TS, the proposed descriptors in this research are based on a weighing sequence progression of the quantised grey-level texture patterns to obtain an integer value that is unique to each texture pattern. If the weighing sequence is progressed regularly to produce integer values for unique texture patterns, the collected information from these texture patterns should be more discriminative, and be able to capture subtle differences between different texture patterns.

To improve the discrimination of the features, two different descriptors are proposed, termed Local Zones Binary Pattern (LZBP) and Local Multiple Pattern (LMP). To effectively exploit the intensity values of neighbour pixels and collect more information from texture patterns, every neighbour's place is divided into a number of zones. This is applied to the neighbour places with intensity values larger than the threshold value (centre pixel), resulting in neighbour places with a set of zones in every place.

In the proposed descriptors, the process starts by dividing the range or space from the centre pixel value to largest value among intensity values of neighbour pixels in TUs into zones. Given a set of intensity values of neighbourhood pixel $\mathrm{g_1, g_2, \ldots\ldots g_p}$, a centre pixel intensity value of $\mathrm{g_c}$, $\mathrm{T_{max}} = \max(\mathrm{g_c},\ \mathrm{g_1, g_2, \ldots\ldots g_p})$, $\mathrm{T_{min}} = \mathrm{g_c}$, the number of required intervals or zones (n) can be expressed as:

$$\mathrm{v_n} = \frac{\mathrm{T_{max}} - \mathrm{T_{min}}}{\mathrm{n}} \tag{3-1}$$

where v_n is the interval or range of zones, $\mathrm{T_{min}}$ is the minimum value and is equal to the value of centre pixel, and $\mathrm{T_{max}}$ is largest value among the intensity values of the TU.

The zone number and the corresponding intervals or range of each zone are explained by Equation (3-2), where the zones are (1,2,3 …., n).

$$v(x) = \begin{cases} n & T_{min} + \dfrac{(n-1) \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{n \times (T_{max} - T_{min})}{n} \\ & \ldots\ldots. \\ & \ldots\ldots. \\ 3 & T_{min} + \dfrac{2 \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{3 \times (T_{max} - T_{min})}{n} \\ 2 & T_{min} + \dfrac{1 \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{2 \times (T_{max} - T_{min})}{n} \\ 1 & T_{min} + \dfrac{0 \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{1 \times (T_{max} - T_{min})}{n} \end{cases} \tag{3-2}$$

(1) Local Zones Binary Pattern (LZBP)

After dividing the neighbourhood places into a number of zones via Equation (3.1), the resulting number of zones in the neighbour places can be mathematically expressed as in Equation (3-2). LZBP codes the intensity values of neighbour pixels using Equation (3-3), where values that exist in the zone or interval are converted into a binary 1, otherwise take a binary 0 value.

$$v(x_{p,z})\begin{cases}1 & if \quad T_{min}+\dfrac{z-1\times\ (T_{max}-T_{min})}{n}<x_z<T_{min}+\dfrac{z\times(T_{max}-T_{min})}{n} \\ 0 & otherwise\end{cases} \tag{3-3}$$

where n is the number of zones, such that that $z = 1 \dots \dots n$, and m is the number of neighbour places, such that that $p = 1 \dots \dots m$.

After thresholding the intensity values of neighbours, an OR logic operation is applied on the coded values of neighbours that are in the same zone. This procedure guarantees a result of 0 or 1 from each zone.

In LZBP, following the coding of the local image pattern, a unique local image pattern is replaced by an integer value of different samples (TUs), which is calculate by Equation (3-4). The calculated code replaces the centre pixel in the LZBP matrix. This is done by following a clockwise direction process, where every coded binary value is multiplied with its corresponding weight, before summing up the result. Here, the corresponding binomial weight is multiplied with resulting values from applying the OR logic operation.

$$\mathrm{LZBP}_{\boldsymbol{OR}}=\sum_{z=0}^{n}v(g_{p,z}-g_c)\times 2^z \tag{3-4}$$

Where $g_{p,z}$ and g_c are the grey-level values at zone and centre pixel respectively and n is the number of zones.

Other information can be obtained from coding the intensity values of neighbour pixels using Equation (3-3). This is done by utilising the number of existing coded 1s in the same zone, which range from 1 to 8 (number of neighbour places). This information is obtained via Equation (3-5), where a unique local image pattern is replaced by an integer value of different samples (TUs).

$$\mathrm{LZBP}_{\boldsymbol{sum}} = \sum_{z=0}^{n} 2^{\sum_{\boldsymbol{p}=\mathbf{0}}^{\mathbf{m}}\left(v\left(g_{p,z}-g_c\right)\right)} \tag{3-5}$$

where n is the number of zones, and m is the number of neighbour places.

The complete set of features is obtained from the histogram of LZBP, which is acquired by concatenating the resulting $\mathrm{LZBP}_{\mathrm{OR}}$ histogram from Equation (3-4) and the resulted $\mathrm{LZBP}_{\mathrm{sum}}$ histogram from Equation (3-5).

Example

Given the set of intensity values of neighbourhood pixels, and the intensity value of the centre pixel shown below, then $\mathrm{T}_{\mathrm{max}} = 40$, $\mathrm{T}_{\mathrm{min}} = 8$, and the number of required intervals or zones is (8).

11	5	14
4	6	15
5	10	8

Based on Equation (3-1), the range of intervals in each zone (interval space) $= \frac{40-8}{8} = 4$

Zone	Thresholding Equation	Thresholding of TUs	Applying OR	Applying Sum
1	$v(x_{p,1})\begin{cases}1 & if \quad 8 \leq x_1 \leq 12 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 1 & c & 0 \\ \hline 0 & 0 & 1 \\ \hline\end{array}$	u1=1	v1=2
2	$v(x_{p,2})\begin{cases}1 & if \quad 12 < x_2 \leq 16 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 1 \\ \hline 0 & c & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$	u2=1	v2=1
3	$v(x_{p,3})\begin{cases}1 & if \quad 16 < x_3 \leq 20 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 1 & 0 & 0 \\ \hline 0 & c & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$	u3=1	v3=1
4	$v(x_{p,4})\begin{cases}1 & if \quad 20 < x_4 \leq 24 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 0 & c & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$	u4=0	v4=0
5	$v(x_{p,5})\begin{cases}1 & if \quad 24 < x_5 \leq 28 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 0 & c & 0 \\ \hline 1 & 0 & 0 \\ \hline\end{array}$	u5=1	v5=1
6	$v(x_{p,6})\begin{cases}1 & if \quad 28 < x_6 \leq 32 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 0 & c & 0 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$	u6=0	v6=0
7	$v(x_{p,7})\begin{cases}1 & if \quad 32 < x_7 \leq 36 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 0 & c & 0 \\ \hline 0 & 1 & 0 \\ \hline\end{array}$	u7=1	v7=1
8	$v(x_{p,8})\begin{cases}1 & if \quad 36 < x_8 \leq 40 \\ 0 & otherwise\end{cases}$	$\begin{array}{\|c\|c\|c\|}\hline 0 & 0 & 0 \\ \hline 0 & c & 1 \\ \hline 0 & 0 & 0 \\ \hline\end{array}$	u8=1	v8=1

Fig. 3. 2 Coded neighbour value before used by equation (3-4) and (3-5) for integer value of TUs.

Based on the resulted from OR logic operation for calculating the $\mathrm{LZBP}_{\boldsymbol{OR}}$ by equation (3-4):

$$u_1 \times 2^0 + u_2 \times 2^1 + u_3 \times 2^2 + u_4 \times 2^3 + u_5 \times 2^4 + u_6 \times 2^5 + u_7 \times 2^6 + u_8 \times 2^7$$

$$1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 0 \times 2^5 + 1 \times 2^6 + 1 \times 2^7$$

Based on the resulted from sum operation for calculating the $\mathrm{LZBP}_{\boldsymbol{sum}}$ by equation (3-5):

$$1 \times 2^{v_1} + 1 \times 2^{v_2} + 1 \times 2^{v_3} + 1 \times 2^{v_4} + 1 \times 2^{v_5} + 1 \times 2^{v_6} + 1 \times 2^{v_7} + 1 \times 2^{v_8}$$

$$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^1$$

(2) Local Multiple Pattern (LMP)

In LMP, the same procedure described by Equation (3-1) for dividing the neighbourhood places into number of zones is adopted. The LMP coding intensity values of neighbour pixels is performed based on Equation (3-6), where values that exist within the interval threshold are coded into corresponding the values of zones number (1,2 … n), otherwise they are coded a value of 0.

$$v(x) = \begin{cases} n \quad if \quad T_{min} + \dfrac{(n-1) \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{n \times (T_{max} - T_{min})}{n} \\ \qquad \qquad \qquad \dots \dots . \\ \qquad \qquad \qquad \dots \dots . \\ 2 \quad if \quad T_{min} + \dfrac{1 \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{2 \times (T_{max} - T_{min})}{n} \\ 1 \quad if \quad T_{min} + \dfrac{0 \times (T_{max} - T_{min})}{n} < x < T_{min} + \dfrac{1 \times (T_{max} - T_{min})}{n} \\ 0 \qquad \qquad \qquad \qquad \qquad \qquad otherwise \end{cases} \tag{3-6}$$

Followed the coding of the local texture pattern, a unique local image pattern is replaced by an integer value of different samples (TUs), which is calculated using Equation (3-7). The calculated integer number replaces the centre pixel of the LMP matrix. This is achieved by following clockwise direction process, where every thresholded value is multiplied with its corresponding weight, before summing up the result.

$$\text{LMP} = \sum_{p=0}^{n} v(g_p - g_c) \times 2^{l*p} \tag{3-7}$$

where l is the number of levels, and its values range from 1 to m, resulting in a $z = 2^m - 1$ zones, and n is the number of neighbour places.

Example

Given the set of intensity values of neighbourhood pixels, and the intensity value of the centre pixel shown below, then $\text{T}_{\text{max}} = 15$ and $\text{T}_{\text{min}} = 6$.

11	5	14
4	6	15
5	10	8

For two LMP levels ($l = 2$), the weights that take places of the neighbours are as follows:

$$2^{2*\text{p}}, \quad \text{where p} = 0{,}1{,}2 \ldots\ldots 7$$

The sequence wights are therefore $2^0, 2^2, 2^4, 2^6, 2^8, 2^{10}, 2^{12}, 2^{14}$

Every neighbour place is divided into n zones, which are then converted or coded into intensity values of neighbour pixels by thresholding, as follows:

$$\text{n (number of zones)} = 2^2 - 1 = 4 - 1 = 3$$

$$2^0, 2^1, (2^0 + 2^1) = 1{,}2{,}3$$

The sequence weights $2^0, 2^2, 2^4, 2^6, 2^8, 2^{10}, 2^{12}, 2^{14}$ will be distributed among the neighbour places $(p_1, p_2, \ldots\ldots p_8)$, whereas the values 1,2,3 will be distributed among the zones (z_1, z_2, z_3) as follows.

The difference $v(x)$ is encoded by the three values according to the corresponding thresholds as in (3-8):

$$v(x) = \begin{cases} 3 & if \quad T_{min} + \frac{2 \times (T_{max} - T_{min})}{3} < x < T_{min} + \frac{3 \times (T_{max} - T_{min})}{3} \\ 2 & if \quad T_{min} + \frac{1 \times (T_{max} - T_{min})}{3} < x < T_{min} + \frac{2 \times (T_{max} - T_{min})}{3} \\ 1 & if \quad T_{min} + \frac{0 \times (T_{max} - T_{min})}{3} < x < T_{min} + \frac{1 \times (T_{max} - T_{min})}{3} \\ 0 & otherwise \end{cases} \qquad (3\text{-}8)$$

This arrangement of sequence weights of neighbour places and zones will produce a different integer for the texture pattern via Equation (3-9), which is linked to where the neighbour value is related to the place and zone.

The integer value from the texture pattern is calculated as follows:

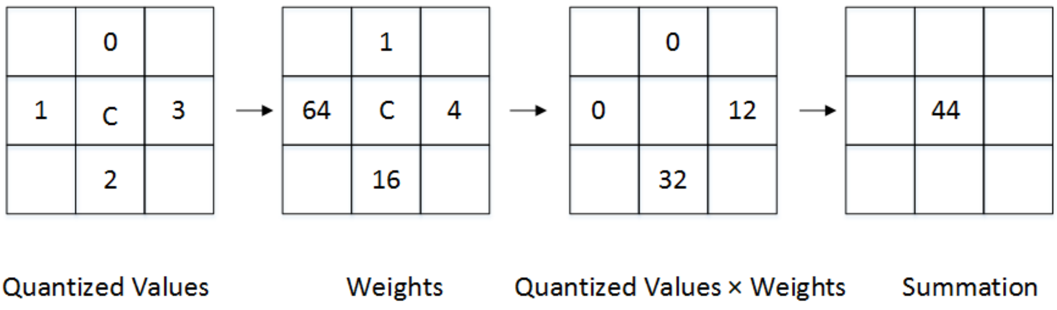
$$\mathrm{LMP} = \sum_{\boldsymbol{p=0}}^{\boldsymbol{7}} v(g_p - g_c) \times 2^{2*p} \qquad (3\text{-}9)$$

From the aforementioned explanation, the TU is supposed to be divided into three zones in each neighbour place. As such, based on Equation (3-1), the range of intervals in each zone (interval space) $= \frac{15-6}{3} = 3$. Thus, the thresholding values of resulted zones are explained in (3-10).

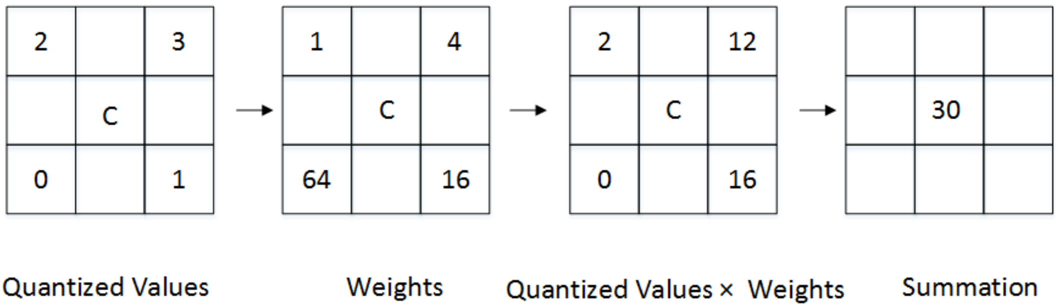
$$v(x) = \begin{cases} 3 & if \quad 12 < x < 15 \\ 2 & if \quad 9 < x < 12 \\ 1 & if \quad 6 < x < 9 \\ 0 & otherwise \end{cases} \qquad (3\text{-}10)$$

Based on dividing the TU into diagonal (plus) pattern and non-diagonal (cross) pattern the result is as follow:

Results of diagonal pattern of TU:



Results of non-diagonal pattern of TU:



3.3.1.3 The Effectiveness of LZBP and LMP

The weight sequence should be in a progression that guarantees a unique integer value for texture patterns of different texture samples. The LBP method is based on binary quantisation of intensity values of neighbour pixels. It consists of a centre pixel surrounded by eight other pixels for each TU in the image. The weight sequence $(2^0, 2^1, \dots \dots 2^p)$ takes place around the centre pixel. Therefore, the total number of the distinct LBP representations is 2^p , where p is the number of neighbour pixels.

In LZBP the weights sequence $(2^0, 2^1, \dots \dots 2^{n-1})$, is assigned to zones $(z_1, z_2, \dots \dots z_n)$, where every neighbour place is divided into number of zones, such that 2^0 is assigned to zone 1, 2^1 to zone 2 , 2^2 to zone 3, ….. , and 2^{n-1} to zone n. This will be repeated for m neighbour place. The final LZBP form then involves concatenation between the histogram resulted from Equation (3-4) and the histogram resulted from Equation (3-5). This makes the total number of the distinct representations of this descriptor $2^n + 2^p$, where n is the number of zones in each neighbour place, and (p) is the number of neighbour places.

In LMP, the weight sequence progresses in such a way as to take in consideration both the neighbour places and constructed zones. The weights are arranged between m neighbourhood places $(p_1, p_2, \ldots\ldots p_m)$, and n of zones in each place $(z_1, z_2, \ldots\ldots z_n)$, as explained in Fig 3.3. Furthermore, in LMP representation, the total number of the distinct representations is $2^{n+1} \times 2^{(l\times(p+1))}$, where n is the number of zones, p is the number of neighbour places, and l is number of levels.

For extracting the local features from texture by the LZBP, LMP, which is similar to LBP, the following procedure is adopted:

1. Convert the image into local texture patterns by thresholding the intensity values of pixels in each TU of the image. In thresholding, the intensity values of neighbourhood pixels are compared to the centre pixel.
2. Compute the integer values of the descriptor that correspond to the local patterns from the previous stage. The statistics of resulting local textures take into account the thresholded intensity values of the corresponding weighing coefficients, then sum the resulting values in the neighbourhood places.
3. Produce the histogram of resulting integer values of the descriptor, where each bin in the histogram records the number of occurrences of a unique local pattern of the descriptor in the image.

Stage	LBP	LZBP	LMP
(1)	Binary threshold (0, 1) the upper and lower zones to quantize the intensity values of neighbours.	Binary threshold the quantized intensity values based on the number (m) of zones, which are produced in each upper neighbour place.	Multiple threshold (n) of quantized intensity values based on the number of zones in each upper neighbour place, where (n) corresponds to the number of zones.
(2)	Weight sequence $2^0, 2^1, \ldots 2^{n-1}$ progresses to take place in the corresponding neighbour places $(p_1, p_2, \ldots p_n)$.	Weight sequence $2^0, 2^1, \ldots\ 2^{m-1}$ progresses to take place in the corresponding number of zones $(z_1, z_2, \ldots\ldots z_m)$.	Weight sequence progresses to take place in both the in the corresponding neighbour places $(p_1, p_2, \ldots p_n)$ and zones $(z_1, z_2, \ldots\ldots z_m)$.
(3)	Feature length of the histogram depends on the number of neighbour places (2^n)	Feature length of the histogram depends on the number of zones (m) in neighbour places (p) via the relation $2^m + 2^p$	Feature length of the histogram depends on the number of zones n in each neighbour place p via the relation $2^{n+1} \times 2^{(l\times(p+1))}$

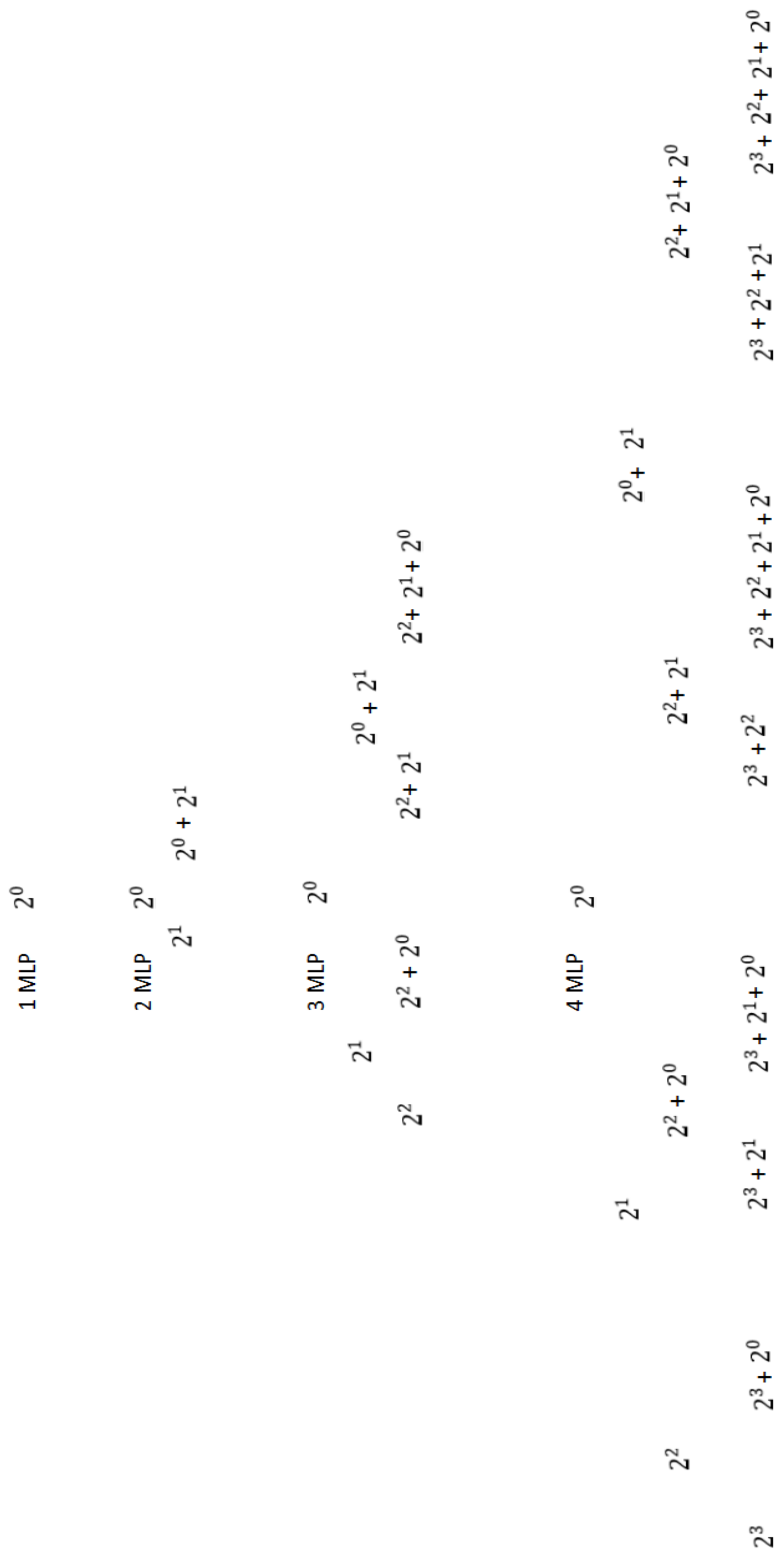


Fig. 3. 3 An explanation of how the values are assigned to zones for LMP.

3.3.2 Hybrid Selection Approach for Feature-level Fusion

The challenge of classifying diverse texture characteristics stems from employing a single feature extraction method. A single feature extraction method is usually not efficient, and faces practical limitations when used with diverse texture characteristics (Bashar & Ohnishi, 2002). This makes utilising more than one feature extraction method a practical solution, as it provides better information about the spatial structure of textures, and produces better discriminative features (Ojala et al., 1994).

To improve texture classification systems, integration between informative features from number of different texture description methods has been proposed. In the previous subsection, new texture descriptors were developed based on information that had not been utilised by the classical LBP method. In Section (2.3.2) of the literatures review, a discussion was presented of how the LBP method could be complemented with other feature descriptors to improve the discriminative capability of feature.

In this research, different feature fusion strategies are proposed. The first feature fusion method is based on combining the developed LZBP and LMP descriptors with the contrast of texture patterns as a complementary feature descriptor. It is worth noting that the use of the contrast of the local image texture as complementary feature descriptor has already been demonstrated to improve the performance of LBP (Ojala et al., 1996). The second feature fusion approach involves combining the local features extracted by the LZBP and LMP descriptors with the global features extracted by a GF, which is the most common complementary feature descriptor used with LBP (Liao et al., 2009). We expect that the accuracy of feature fusion between the proposed descriptors and the suggested complementary feature descriptors will be better than if any of feature descriptors was employed separately.

In this work, feature selection is an important part of the proposed feature descriptors. LZBP and LMP are utilised in a multiscale analysis, which is recommended before proceeding with LBP for extracting richer information from textures (Ojala, Pietikainen, et al., 2002). The overall features extracted from the multiscale descriptors are accomplished by concatenating histograms of different scales, which increases the feature space. In (Topi, Timo, Matti, & Maricor, 2000), the authors explained that the full LBP histogram may not be relevant or necessary, and that selecting part of the patterns encoded in LBP is more effective in producing higher classification rates compared to depending on a complete LBP histogram. The proposed complementary feature descriptor based on a GF also results in effective feature selection. Referring into Subsection 2.2.2.2, GF-based methods require long computation times when depending on complete filters without optimisation (Randen & Husoy, 1999).

Feature selection is also a crucial stage for improving the features extracted by the proposed feature fusion methods. In feature fusion, part of the extracted features usually becomes irrelevant after the fusion operation (A. Jain & Zongker, 1997). In addition, different shared features usually produce a large feature vector, which results in what is referred to as the "curse of dimensionality". In this case, the feature selection method selects the optimal set of features and removes the irrelevant features. The main task of feature selection methods is to decrease the feature length while retaining classification accuracy.

Reducing the feature space and avoiding high dimensionality by finding an optimal set of features is not an easy task, as explained in Subsection 2.4.1. For achieve this task and realise effective feature-level fusion, a new wrapper feature selection approach is proposed by utilising a hybrid method based on the ABC and NRS algorithms (see Subsection 2.4.4).

ABC is a recent and effective wrapper method, which is adopted in this research to improve feature extraction (see Subsection 2.4.3.3). ABC is proposed as a means of selecting the relevant features from feature-level fusion, where LZBP and LMP are combined with either contrast features or GF. The aforementioned methods are used as complementary feature descriptors to improve the texture classification rate. However, wrapper methods, of which ABC is one, are expensive since they consume significant computation time when evaluating the selected feature parts.

In this research, improving feature extraction by multiscale LZBP and LMP increases the search space for optimal feature parts. While the filter method is a heuristic method (like RS) which has the ability to deal with specific feature lengths, it becomes inappropriate with increasing the feature space, resulting in increasing the difficulty to find the optimum features. RS is proposed in this research to be applied on limited size features that result from the histogram of the multiscale LZBP and LMP descriptors. In the proposed procedure, the feature space of multiscale LZBP and LMP can be reduced by NRS by only selecting the relevant features before the feature-level fusion stage, where feature fusion takes place with other complementary features of contrast or GF.

Furthermore, in this research, an improved hybrid selection method is adopted as a more suitable means for selecting optimum features from the proposed feature descriptors. The proposed models for improving texture feature extraction for image classification are as follows:

Hybrid ABC-NRS method for feature fusion of LZBP and LMP with Contrast feature

Figure 3.4 illustrates the proposed wrapper method of ABC based on NRS is employed for extracting optimum features from combining the proposed feature LZBP and LMP descriptors with the image contrast measurement. NRS is used for selecting the relevant features from the multiscale descriptors. The ABC is subsequently utilised to select appropriate features from the reduced features of the multiscale LZBP or LMP descriptors and the reduced features of contrast, when combined together for the final improved features.

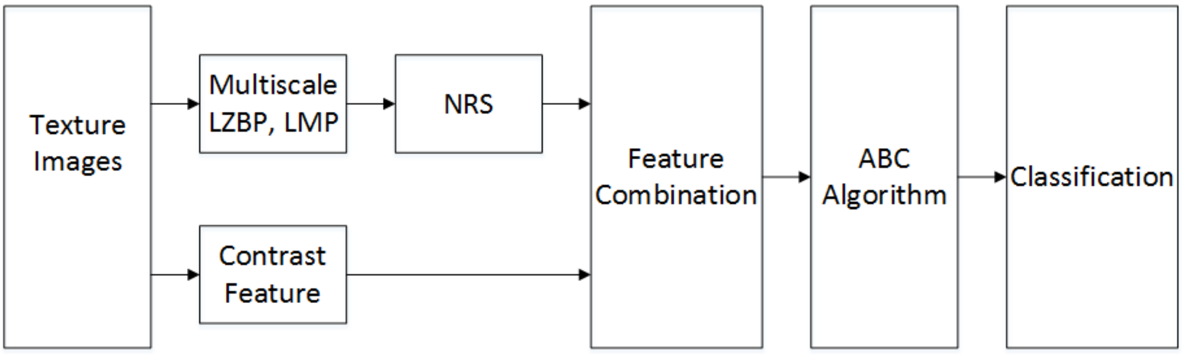


Fig. 3. 4 The proposed feature reduction using ABC-NRS for feature fusion between Contrast feature and LBP, LZBP and LMP descriptors.

Hybrid ABC-NRS method for feature fusion of LZBP and LMP with GF

Figure 3.5 illustrates the proposed wrapper method of ABC based on NRS is employed for extracting optimum features from combining the proposed feature LZBP and LMP descriptors with the GF. Utilises NRS to reduce the size of multiscale LZBP and LMP features. In addition, GF bank consisting of a number of filters, which produce a set of features, is utilised. The most relevant features are selected using the ABC algorithm. Subsequently, further reduction of the combined feature dimension between the features of optimum filters and the reduced features of LZBP or LMP is achieved using the ABC algorithm to yield the final improved feature set.

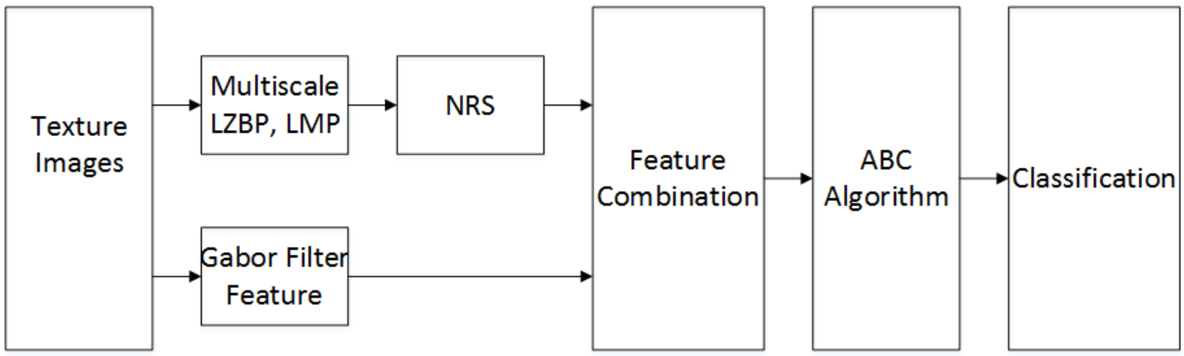


Fig. 3. 5 The proposed feature selection method using ABC-NRS for feature fusion between GF and LBP, LZBP, and LMP descriptors.

3.4 Feature Post-Processing

3.4.1 Supervised Classification Methods

In texture classification systems, the process starts by feature extraction then classification using a learning algorithm (classifier). The learning algorithm is required in this work for the assessment features, which can result from single descriptors, combination of multiple texture descriptors, or for evaluating potential optimum features of wrapper method.

Classifiers can be divided into supervised and unsupervised methods (Svozil, Kvasnicka, & Pospichal, 1997). The difference between the two approaches is that in the unsupervised classification, there is no information in advance about the number of object classes (Kohonen, 1990). In the classification stage, the unsupervised classifiers are not relevant in this research. A successful categorisation of images by unsupervised classifiers, which require training samples, are more appropriate for enhancing the efficiency of classification (Olaode, Naghdy, & Todd, 2014). One example application can be found in the context of image clusters that are mostly used in large databases, where such classifiers can provide a good overview about the database (Vailaya, Jain, & Zhang, 1998).

In supervised classification methods, the objects are divided into classes based on labelling/assigning the objects into classes by values, where the value of each label matches with a class. Supervised classification starts by training samples (*xi*, *yi*), which contain a feature vector *xi*, and a corresponding class label *yi*. In classification methods (learning algorithm), the input is a feature extracted from the training samples of images by descriptors, whereas the output is the classification of these samples. The features are extracted by descriptors from the training samples of images, and the classifier is constructed by these features from descriptors, which are then (i.e. the features) evaluated using the classification system.

In texture classification based on improved features from different feature processing methods (Section 3.3), supervised classifiers are employed to classify the resulting features. Different classifiers can be used such as the Back Propagation Neural Network (BPNN) classifier, or the Support Vector Machines (SVM) classifier (refer to Subsection 2.1.3.2 for details). However, classifier, is not the major part of this research.

3.4.2 Evaluation of Classification Systems

The implementation of the designed prototype of the proposed feature methods in a classification system will follow an organised approach. Before carrying out the evaluation, it is necessary to determine suitable procedures and choose appropriate tools for evaluation. Previous studies found specific benchmarks to be restricted to certain application areas. However, our work will depend on different available choices that are utilised in evaluating the improved features for image classification.

Datasets Benchmarks

The pre-processing stage (Section 3.2) involves preparing different texture datasets. These datasets consist of different texture characteristics, which are used as challenge of images classification by texture. It is essential that the performance of the proposed approaches is evaluated on well-known datasets. Furthermore, these datasets should be useful for evaluating classification methods through different texture characteristics. This demands selecting databases that reflect a variety of textures surfaces to be used as a benchmark for different texture extraction methods.

This research is based on several datasets that contain a collection of texture surfaces. There are number of datasets of texture images that have been used to evaluate different methods that relate to enhancing texture feature extraction (Crosier & Griffin, 2008; Varma & Garg, 2007; Xu, Yang, Ling, & Ji, 2010; J. Zhang, Marszałek, Lazebnik, & Schmid, 2007). Recently, these datasets have been used to evaluate the developed methods that are based on LBP (L. Liu, Fieguth, Wang, Pietikäinen, & Hu, 2016). The targeted datasets provide the required diversity of texture characteristics and randomness required by texture feature analysis methods. The datasets consist of images that vary in scale, direction, viewpoint, and illumination, as well as images affected by noise and blurring problems. These real images are mostly used for comparison between different feature extraction methods.

Comparison Methods

In order to evaluate the developed methods, it is necessary to compare said methods with other related methods. A new method can only be claimed successful or beneficial if its produced results can be validated as superior to those of existing methods.

As we target improving the classification accuracy by utilising a hybrid feature selection method with least feature size (Subsection 3.3.2), the accuracy and feature length rates can be used to evaluate the results. For methods that are based improving texture features by feature selection, the comparison is mostly based on the accuracy and feature space of classification results (H. Liu & Motoda, 2007).

Basic Metrics

There are various measures to evaluate the performance of classification systems, such as the Confusion Matrix (CM), which is also referred to as the Error Matrix (Lu & Weng, 2007). CM is also appropriate for multi-class image classification. Measures or metrics such as accuracy, precision, recall, and F-score are calculated from CM, and are used to evaluate the classification system (Sokolova & Lapalme, 2009).

In the classification process, the information in the CM is related to actual class and predicted class. Figure 3.6 illustrates this point for multi-class classification. When testing the data produced by a classification model, correct and incorrect samples from classifications of every class are counted and displayed in this table, where the table includes True Positive (TP) for each class and Error (E) of sample classes, where they belong to one of the classes but were incorrectly classified to other classes.

		Prediction				
		Class 1	Class 2	Class 3	...	Class n
Actual	Class 1	TP (1)	E (12)	E (13)	...	E (1n)
	Class 2	E (21)	TP (2)	E (23)	...	E (2n)
	Class 3	E (31)	E (32)	TP (3)	...	E (3n)
	...	...	...	...	...	...
	Class n	E (n1)	E (n2)	E (n3)	...	TP (n)

Fig. 3. 6 The confusion matrix for multi-class classification.

3.5 Summary

This chapter proposed a methodology for improving feature extraction for images classification systems based on the characteristics of texture. Mathematical methods based on LZBP and LMP were introduced as means of texture description. These texture descriptors aim to extract distinctive TUs features, and are developed based on the conventional LBP descriptor. Furthermore, the new feature descriptors are combined with two other complementary feature descriptors, which are contrast and GF. In addition, the proposed methodology takes into account the fact that any reliable classification system working with feature fusion should have the ability to select only the relevant features, and avoid curse of dimensionality. As such, a new hybrid feature selection approach is proposed by wrapper method based on ABC and NRS. The method is well equipped to deal with a huge feature size, and is capable of dealing with feature space efficiently.

The developed approach is expected to result improved texture features for image classification. These methods will be experimentally evaluated based on a specific procedure that has been followed by previous classical methods. This will thus verify the effectiveness of this approach in the context of classification systems.

Chapter 4

Design and Implementation

This chapter introduces the proposed methods for achieving improved features extraction from textures, and which will be applied in a classification system. These methods are implemented through their algorithms, which was coded by a MATLAB program. MATLAB is utilised by many programmers because it provides a convenient environment for algorithm implementation. In addition, MATLAB has the capabilities required to implement the employed methods mentioned in previous methodology chapter.

In this chapter, the developed prototypes of the methods intended to improve feature extraction are introduced in Section 4.1. Section 4.2 presents the implementation of the algorithms based on the new LZBP and LMP texture descriptors. Section 4.3 explains the process applied for improving feature extraction from texture, whereas Section 4.4 details the process of the algorithms based on the methods used to improve feature extraction in classification systems. The final section, Section 4.5, provides a summary of this chapter.

4.1 Design of the Developed Prototype

The proposed classification system follows the basic workflow shown in Fig 1.2. In this work, the proposed design involves the components related to the feature process stage (see Fig 4.1). In classification, the process starts with the submission of a number of images to the system. The system then extracts the features of the image using the new improved feature extraction methods based on LZBP and LMP, whereas complementary features are extracted either through contrast measurements or a Gabor filter. Then the system integrates the features from involved texture descriptors into a unified vector. The next step involves applying feature selection for a subset of relevant features using the hybrid ABC-NRS algorithm. Finally, the classifier receives the results of the improved features to classify the images into a number of classes.

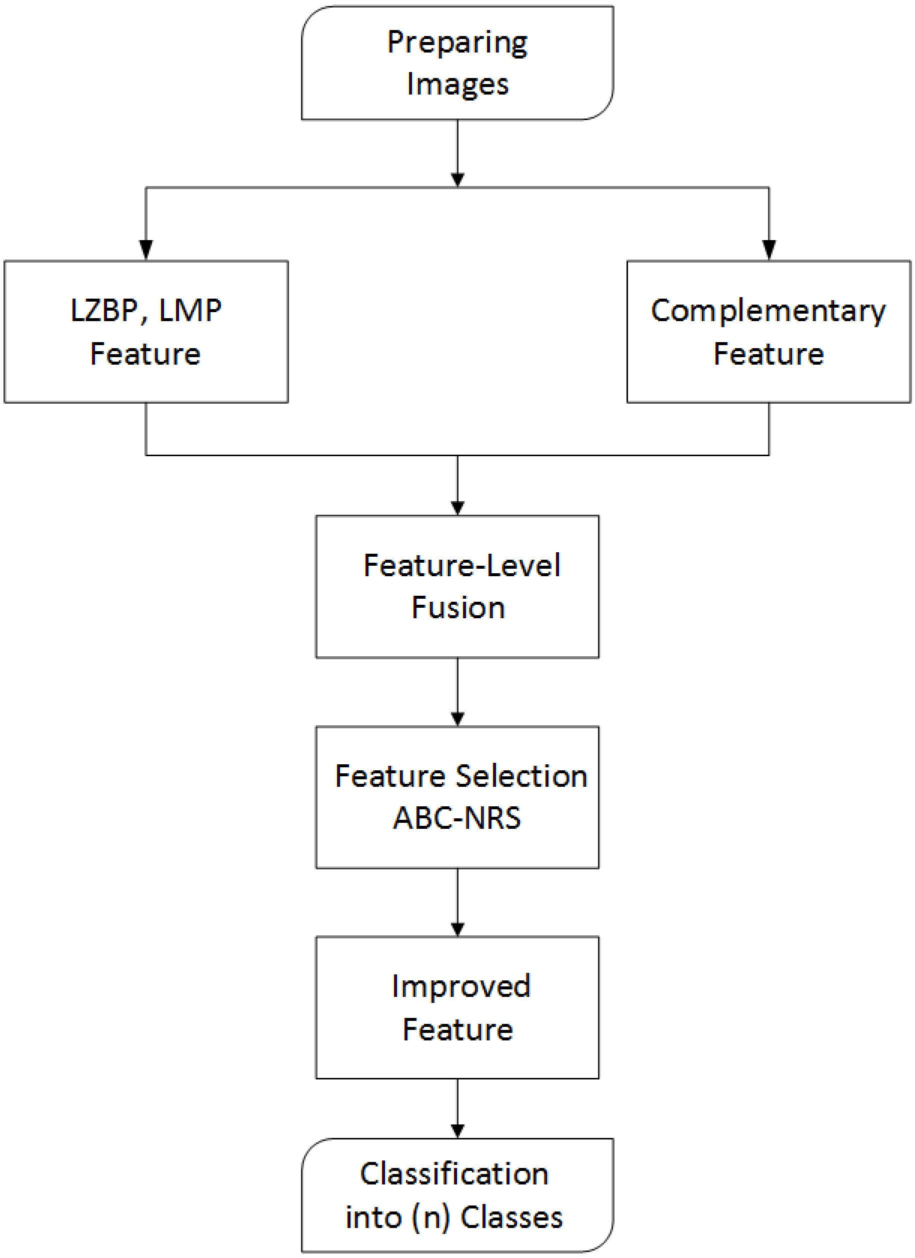


Fig. 4. 1 Prototype of the feature extraction process based on the methods used to improve feature extraction in a classification system.

4.2 Texture Feature Extraction by LZBP and LMP

In texture-based classification systems, a set of visual features is extracted from target images. This is preferred to involving the entire data or pixels of the images. The feasibility of this depends on the visual features, however, collecting the sufficient features from the textures that reflect their characteristics is always of great importance (Amato & Di Lecce, 2008).

As explained in the previous chapter, LZBP and LMP are based on the LBP method for collecting distinctive features from texture characteristics. LZBP and LMP employ more sophisticated quantisation functions for texture patterns to extract improved features compared to LBP. The procedure used by LZBP and LMP is suitable for texture patterns, especially for varying intensity values of patterns.

4.2.1 Algorithms of the LZBP and LMP

In developing new descriptors based on LZBP and LMP, the relationship between the central pixel value (threshold) with intensity values of neighbourhood pixels has been established. This procedure involves discriminating between patterns from different characteristics of texture, and collecting as much information as possible. In LZBP- and LMP-based texture descriptors, the process involves constructing a number of zones which quantise the intensity values of neighbourhood pixels (refer to Section 3.3.1).

For coding the patterns in TUs, the LZBP and LMP methods establish:

- A number of pixels in a neighbourhood (p), which are surrounded by a grey-level value of the centre pixel.
- A number of quantisation zones (z) in the neighbourhood places.

4.2.1.1 LZBP Algorithm

Listing 4.1 shows the developed LZBP algorithm. The extracted visual feature vector is done via the relation between the threshold value of the central pixel and the other intensity values of neighbourhood pixels. However, in LZBP, the constructed number of zones are, which serve to quantise the intensity values of neighbourhood pixels, as show in line 7 by the FOR loop. The extracted features depend on the number of quantised zones (z), as this impact the recognition capability of intensity values of neighbourhood pixel.

In line 10, the WHILE loop determines the intensity values of the neighbourhood that exists in the same zone. The LZBP algorithm, based on the condition displayed in line 12, the texture pattern in the zone is converted into (1) (set_pattern=1) as show in line 13. Subsequently, different results from all neighbourhood intensity values are saved by $Set\_\ Zonepattern$, as show in line 15.

The feature is calculated by the histogram, where an OR logic operation is applied on the resulting coded values saved by $Set\_\ Zonepattern$ following Equation (3-4), whereas the summation operation is applied on resulting coded values saved by $Set\_\ Zonepattern$ following Equation (3-5).

Listing 4.1: LZBP algorithm

1. $pattern\ \leftarrow 0$

2. $Center\_value \leftarrow Sample\_center$

3. $\boldsymbol{for}\ i = 1\ to\ p\ \boldsymbol{do}$

4. $Max\_value \leftarrow Max(i)$

5. $Min\_value \leftarrow Center\_value$

6. $\boldsymbol{end}$

7. $\boldsymbol{for}\ j = 1\ to\ z\ \boldsymbol{do} : \boldsymbol{z\ is\ number\ of\ zones}$

8. $term\_\max(j) \leftarrow Min\_value + j\ \times (Max\_value - \ Min\_value)\ /z$

9. $term\_\min(j) \leftarrow Min_{value} + (j-1)\ \times (Max\_value - \ Min\_value)\ /z$

10. $\boldsymbol{while}\ i = 1\ to\ p\ \boldsymbol{do} : \boldsymbol{p\ is\ number\ of\ neighbour\ patterns}$

11. $neighour\_value \leftarrow Sample\_value(i)$

12. $\boldsymbol{if}\ \left(neighour_{value} > Center_{value} \& \ term_{\max(j)} < neighour_{value} < term_{\min(j)}\right)\boldsymbol{then}$

13. $Set\_pattern(i) \leftarrow 1$

14. $\boldsymbol{end}$

15. $Set\_Zonepattern(i,j) \leftarrow Set\_pattern(i)$

16. $\boldsymbol{end}$

17. $\boldsymbol{end}$

4.2.1.2 LMP Algorithm

Listing 4.2 presents the developed LMP algorithm. The extracted feature vectors are still based on the relation between the threshold value of central pixel and the other intensity values neighbourhood pixels. The LMP algorithm use the same strategy for constructing a number of zones, which quantise the intensity values of neighbourhood pixels, as show in line 7 (the FOR loop). However, for resulting coded value, the LMP algorithm uses the condition in line 12, where the intensity values of neighbourhood that exist in the same zone are converted into the zone number (j) by (set_pattern=j), as shown in line 13. The result from set_pattern is then used by Equation (3-7), and is subsequently used by the histogram for the feature vector.

Listing 4.2: LMP algorithm

1. $pattern \leftarrow 0$

2. $Center\_value \leftarrow Sample\_center$

3. $\boldsymbol{for}\ i = 1\ to\ p\ \boldsymbol{do}$

4. $Max\_value \leftarrow Max(i)$

5. $Min\_value \leftarrow Center\_value$

6. $\boldsymbol{end}$

7. $\boldsymbol{for}\ j = 1\ to\ z\ \boldsymbol{do} : \boldsymbol{z\ is\ number\ of\ zones}$

8. $term\_\max(j) \leftarrow Min\_value + j \times (Max\_value - Min\_value)\ /z$

9. $term\_\min(j) \leftarrow Min\_value + (j-1) \times (Max\_value - Min\_value)\ /z$

10. $\boldsymbol{while}\ i = 1\ to\ p\ \boldsymbol{do} : \boldsymbol{p\ is\ number\ of\ neighbour\ patterns}$

11. $neighour\_value \leftarrow Sample\_value(i)$

12. $\boldsymbol{if}\ \left(neighour_{value} > Center_{value} \&\ term_{\max(j)} < neighour_{value} < term_{\min(j)}\right)\boldsymbol{then}$

13. $Set\_pattern(i)\ \leftarrow j$

14. $\boldsymbol{end}$

15. $\boldsymbol{end}$

16. $\boldsymbol{end}$

4.2.2 Implementation of the LZBP and LMP

In this section, an example of the implementation of the LZBP and LMP algorithms on sample TUs is presented and compared with the LBP algorithm. The TUs sample is from different classes of images, as shown in Fig 4.2 (a).

TUs coding

The first step involves the coding of the intensity values of neighbour pixels in TUs. Following Equation (2-7), LBP happens to produce the same binary code value from the relationship between threshold and intensity values of two different textures patterns. From Fig 4.2 (b), using LBP, both TUs are classified into the same class, when they in fact belong to different classes. LBP applied to TU Class 1 and TU Class 2 results in both being converted to '1 1 0 0 1 1 1 1'. The repeated existence of these patterns in textures from different classes results in the histogram of the two classes being similar, which increases the probability of the two textures belonging to the same class.

In the LZBP algorithm, the process starts by Equation (3-3), which is use for coding the intensity values of TUs. In the case of OR logic operation by (3-4) as in Fig 4.2 (c) (i), the intensity values of TU Class 1 are converted to '1 0 1 0 1 1 1 0', whereas the intensity values of TU Class 2 are coded into '1 0 1 0 0 0 1 0 1'. In the case of sum operation by (3-5) as in Fig 4.2 (c) (ii), when coding the same texture patterns, the coded TU Class 1 is equal to ‘1 0 2 0 1 1 1 0’, while the TU Class 2 is equal ‘2 0 2 0 0 1 0 1’. The coding of different samples of images results in different code numbers, and as such, it is expected to produce different features for Class 1 and Class 2.

In the LMP algorithm, Equation (3-6) is used to code the intensity values of TUs with a three-level result, as in Fig 4.2 (d). For the first quantized zone of level 1, the result of the TU Class 1 is ‘0 1 0 0 0 0 1 1’ and the result of the TU Class 2 is 1 1 0 0 1 0 0 0’, whereas for the second quantized zone of level 2, the result of the TU Class 1 is ‘0 0 0 0 2 0 0 0’ and the result of the TU Class 2 is ‘0 0 0 0 0 0 0 2’. Finally, for the third quantized zone of level 3, the TU Class 1 is converted into ‘3 0 0 0 0 3 0 0’, while the TU Class 2 is converted into ‘0 0 0 0 0 0 3 0’. The results of level 1, 2 and 3 of the algorithm thus produce different coding for Class 1 and Class 2.

TU of Class 1 and TU of Class 2 are from different texture images. However, despite the TUs being from different classes, the LBP algorithm produced the same code values. On the other hand, the developed LZBP and LMP algorithms produced different coded values for different TUs classes. As such, this highlights the capability of the LZBP and LMP algorithms in discriminating between texture patterns.

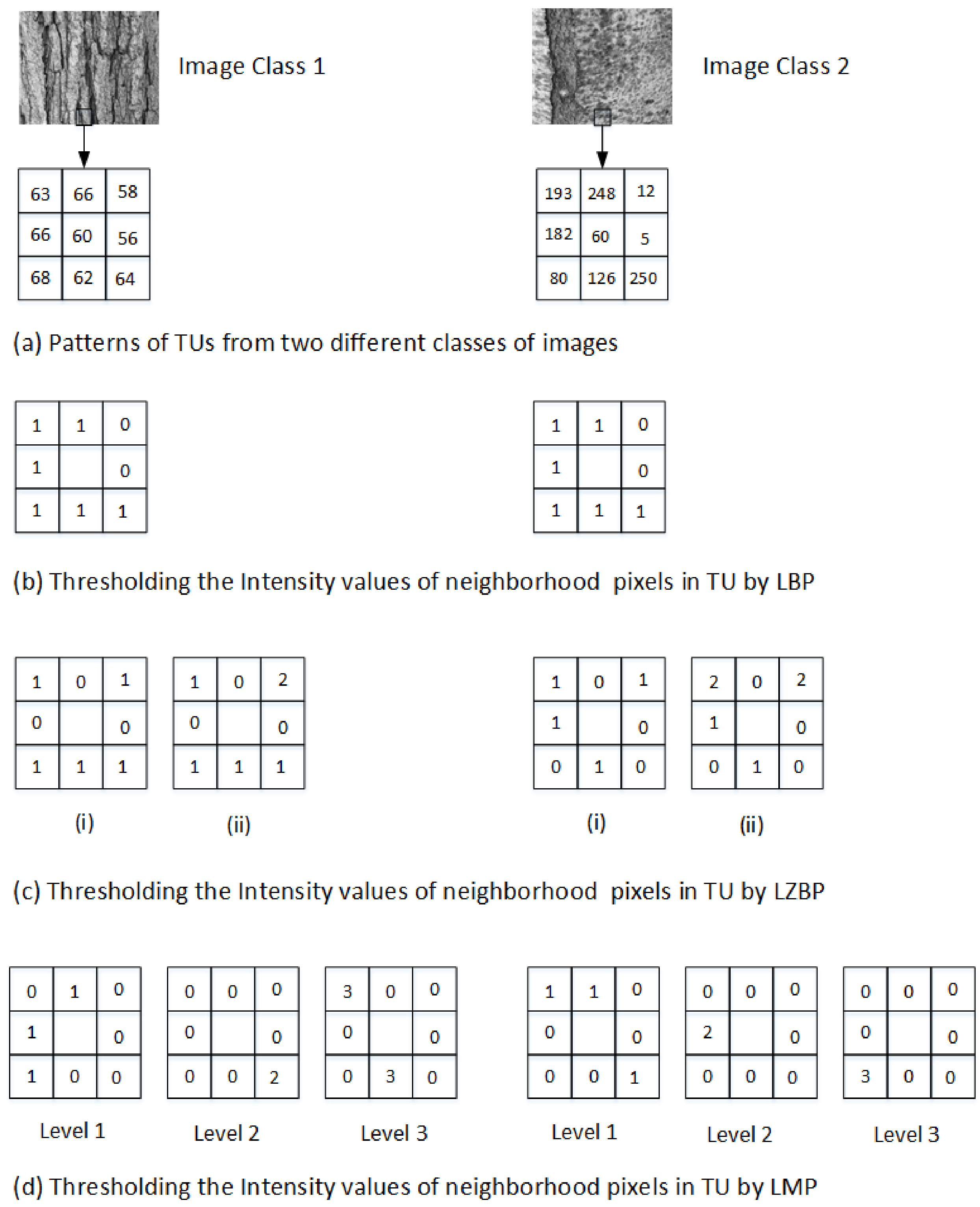


Fig. 4. 2 Results of coding TUs from two different classes of images by LBP, LZBP and LMP.

Mapping TUs into integer values

In this step, the two descriptors apply the sum of weighted coded values of neighbour places to uniquely map them to an integer value. Equation (3-4) and (3-5) of LZBP and Equation (3-7) of LMP are used to guarantee a texture pattern with a unique integer value for different TUs.

<u>Results of Class 1 TU:</u>

In LBP, the integer value is calculated by Equation (2-8):

$$LBP = 2^7 \times 1 + 2^6 \times 1 + 2^5 \times 0 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1$$

In LZBP, the integer value is calculated by Equations (3-4) and (3-5):

$$LZBP\_OR = 2^7 \times 1 + 2^6 \times 0 + 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 0$$

$$LZBP\_sum = 2^1 \times 1 + 2^0 \times 1 + 2^2 \times 1 + 2^0 \times 1 + 2^1 \times 1 + 2^1 \times 1 + 2^1 \times 1 + 2^0 \times 1$$

$$LZBP = Concatenate(LZBP\_OR, LZBP\_sum)$$

In LMP, the integer value is calculated by Equation (3-7):

$$\begin{aligned} LMP\_cross = {} & (2^6 \times 1) \times (1 \times 3) + (2^6 \times 1) \times (0 \times 2) + (2^6 \times 1) \times (0 \times 1) + \\ & (2^4 \times 1) \times (0 \times 3) + (2^4 \times 1) \times (0 \times 2) + (2^4 \times 1) \times (0 \times 1) + \\ & (2^2 \times 1) \times (0 \times 3) + (2^2 \times 1) \times (1 \times 2) + (2^2 \times 1) \times (0 \times 0) + \\ & (2^0 \times 1) \times (0 \times 3) + (2^0 \times 1) \times (0 \times 2) + (2^0 \times 1) \times (1 \times 1) + \end{aligned}$$

$$\begin{aligned} LMP\_plus = {} & (2^6 \times 1) \times (0 \times 3) + (2^6 \times 1) \times (0 \times 2) + (2^6 \times 1) \times (1 \times 1) + \\ & (2^4 \times 1) \times (0 \times 3) + (2^4 \times 1) \times (0 \times 2) + (2^4 \times 1) \times (0 \times 1) + \\ & (2^2 \times 1) \times (1 \times 3) + (2^2 \times 1) \times (0 \times 2) + (2^2 \times 1) \times (0 \times 1) + \\ & (2^0 \times 1) \times (0 \times 3) + (2^0 \times 1) \times (0 \times 2) + (2^0 \times 1) \times (1 \times 1) + \end{aligned}$$

$$LMP = Concatenate(LMP\_cross, LMP\_plus)$$

<u>Results of Class 2 TU:</u>

Results of coding TUs from two different classes of images by LBP, LZBP, and LMP.

In LBP, the integer value is calculated by Equation (2-8):

$$LBP = 2^7 \times 1 + 2^6 \times 1 + 2^5 \times 0 + 2^4 \times 0 + 2^3 \times 1 + 2^2 \times 1 + 2^1 \times 1 + 2^0 \times 1$$

In LZBP, the integer value is calculated by Equations (3-4) and (3-5):

$$LZBP\_OR = 2^7 \times 1 + 2^6 \times 0 + 2^5 \times 1 + 2^4 \times 0 + 2^3 \times 0 + 2^2 \times 1 + 2^1 \times 0 + 2^0 \times 1$$

$$LZBP\_sum = 2^2 \times 1 + 2^0 \times 1 + 2^2 \times 1 + 2^0 \times 1 + 2^0 \times 1 + 2^1 \times 1 + 2^0 \times 1 + 2^1 \times 1$$

$$LZBP = Concatenate(LZBP\_OR, LZBP\_sum)$$

In LMP, the integer value is calculated by Equation (3-7):

$$\begin{gathered} LMP\_cross = (2^6 \times 1) \times (0 \times 3) + (2^6 \times 1) \times (0 \times 2) + (2^6 \times 1) \times (1 \times 1) + \\ (2^4 \times 1) \times (0 \times 3) + (2^4 \times 1) \times (0 \times 2) + (2^4 \times 0) \times (0 \times 1) + \\ (2^2 \times 1) \times (0 \times 3) + (2^2 \times 1) \times (0 \times 2) + (2^2 \times 1) \times (1 \times 1) + \\ (2^0 \times 1) \times (1 \times 3) + (2^0 \times 1) \times (0 \times 2) + (2^0 \times 1) \times (0 \times 1) + \end{gathered}$$

$$\begin{gathered} LMP\_plus = (2^6 \times 1) \times (0 \times 3) + (2^6 \times 1) \times (0 \times 2) + (2^6 \times 1) \times (1 \times 1) + \\ (2^4 \times 1) \times (0 \times 3) + (2^4 \times 1) \times (0 \times 2) + (2^4 \times 1) \times (0 \times 1) + \\ (2^2 \times 1) \times (0 \times 3) + (2^2 \times 1) \times (0 \times 2) + (2^2 \times 1) \times (0 \times 1) + \\ (2^0 \times 1) \times (0 \times 3) + (2^0 \times 1) \times (1 \times 2) + (2^0 \times 1) \times (0 \times 1) + \\ LMP = Concatenate(LMP\_cross, LMP\_plus) \end{gathered}$$

Table. 4. 1 The resulted integer values for LBP, LZBP and LMP from TU sample of class 1 and class 2.

	TU Sample of Class 1	TU Sample of Class 2
LBP	207	207
LZBP	165/16	174/15
LMP	193/194	75/75

Feature histogram

The aforementioned step (1) and step (2) procedure is repeated for every TU of the two classes of the target images. The histogram is then obtained by representing the local features by the descriptors. Fig 4.3 and 4.4 show the resulting histograms of image Class 1 and image Class 2, respectively, where subfigures labelled (a) represent the LBP histogram, subfigures labelled (b1) and (b2) represent the LZBP histograms resulting from $LZBP\_OR$ and $LZBP\_sum$, respectively, and subfigures labelled (c1) and (c2) represent the LMP histograms resulting from LMP_cross and LMP_plus, respectively.

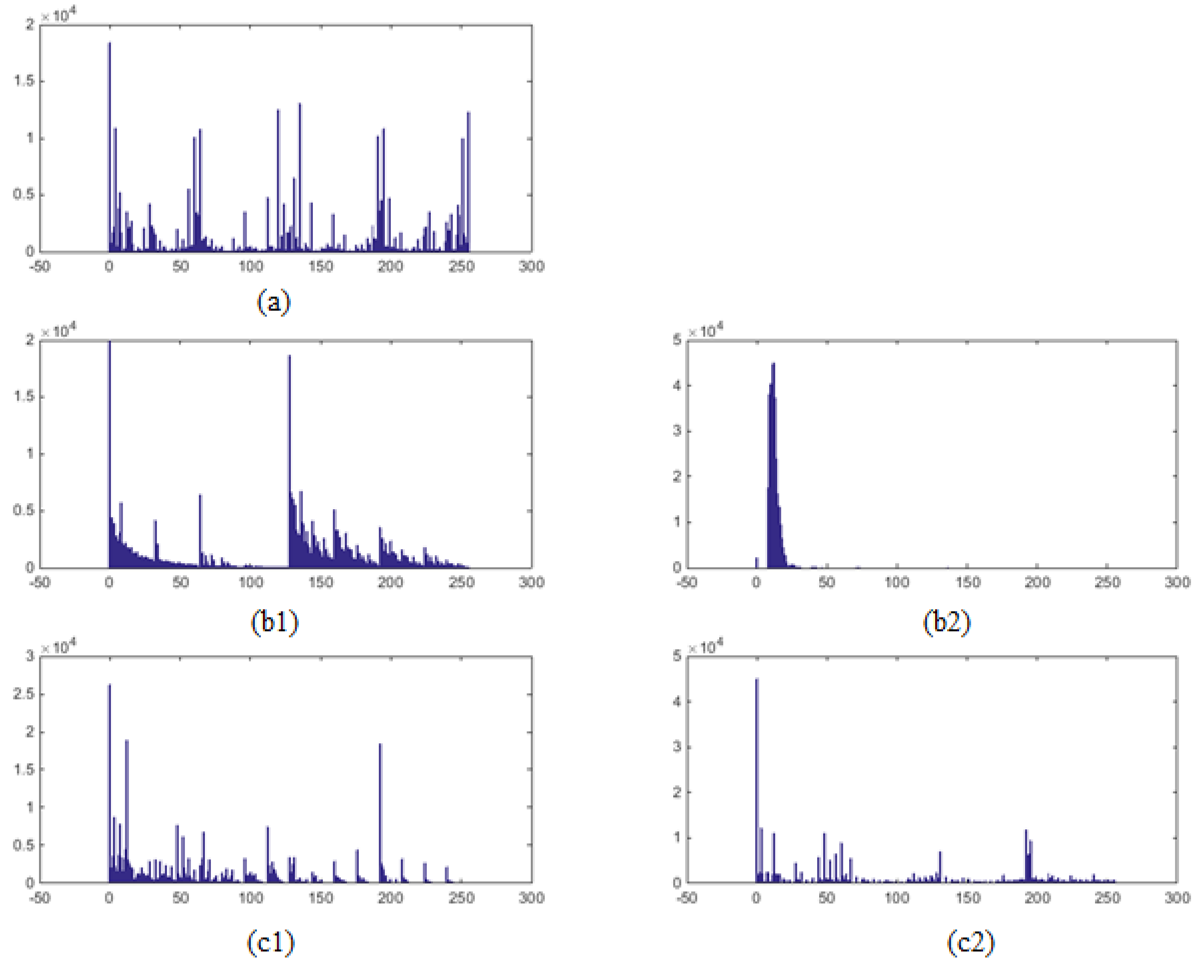


Fig. 4. 3 The LBP, LZBP and LMP histograms of texture Class 1.

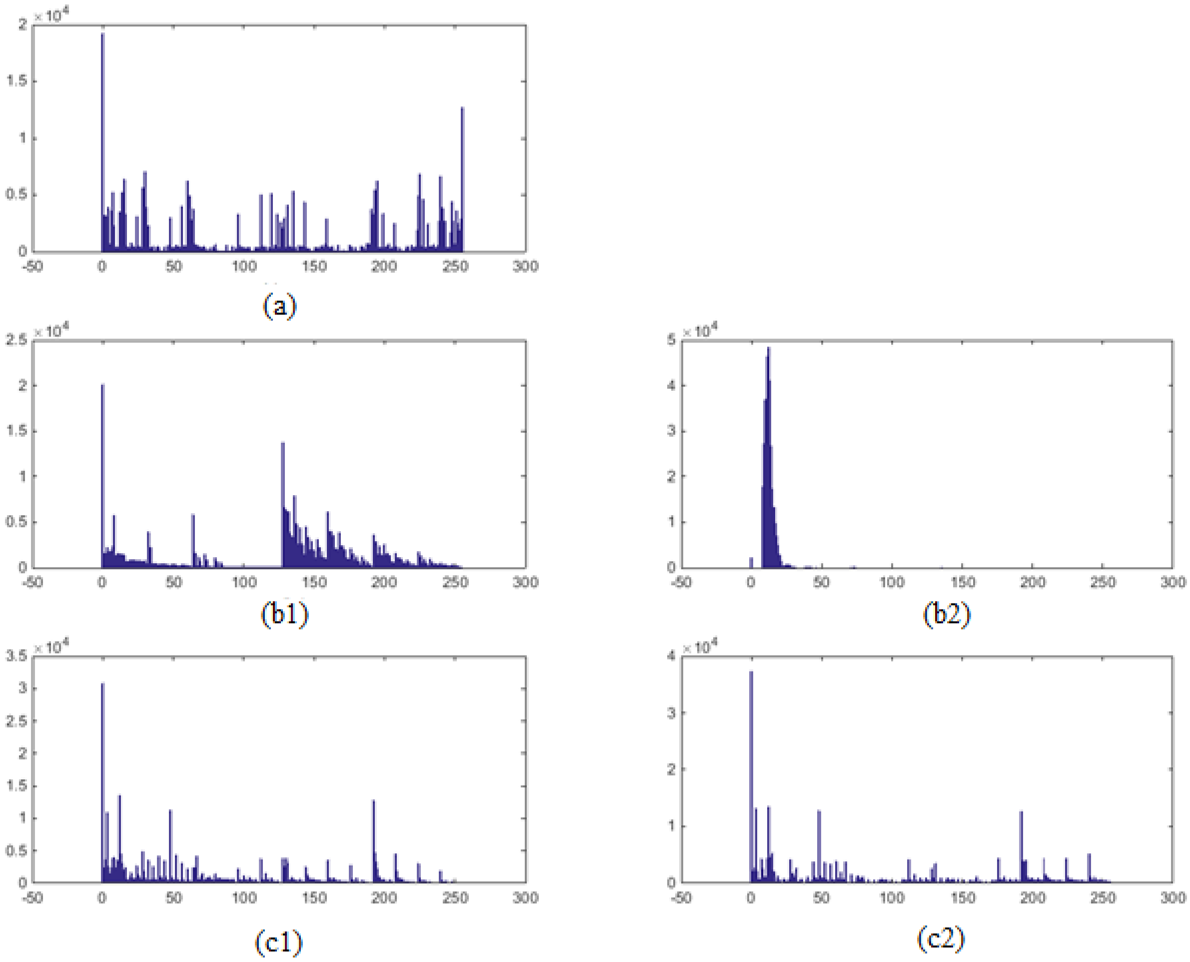


Fig. 4. 4 The LBP, LZBP and LMP histograms of texture Class 2.

4.3 The Improved Feature Extraction Process

In previous section, new feature descriptors based on the LZBP and LMP methods were implemented. As explained in Fig 4.1, these new features descriptors will be combined with other complementary features descriptors to improve feature extraction from different texture characteristics. Feature selection of feature-level fusion output will then be applied to result in optimum features.

4.3.1 Texture Descriptors

The goal of texture descriptors is to create feature vectors consisting of meaningful information about the texture characteristics of an image. The implemented descriptors for extracting features from texture in the developed prototype are:

Texture-based feature descriptors utilising the LZBP and LMP methods

LZBP and LMP extract visual features from target images, where their implementation was discussed in the previous section. As previously explained, LZBP produces the features from concatenation of two histograms. The first is obtained from the histogram resulting from the OR logic operation (Equation (3.4)), whereas the second is obtained from the sum operation (Equation (3.5)).

In LMP, the histogram is applied to the results of LMP equation (Equation (3.7)). Applying all 16-bits patterns results in a high feature dimension of 2^{16}. To reduce the huge feature vector size, LMP is applied by an 8-bit patterns using diagonal (plus) pattern and non-diagonal (cross) pattern (Heikkilä et al., 2009). As such, the LMP produces a couple of histograms, where concatenating LMP_cross and LMP_plus results in a feature dimension of 512 (2^8+ 2^8).

The feature vectors of the proposed LZBP and LMP feature descriptors are:

The feature vector representing the LZBP extracted features is $F_{LZBP} = [F_{1\ LZBP}, F_{2\ LZBP},\ldots F_{M\ LZBP}]$

The feature vector representing the LMP extracted features is $F_{LMP} = [F_{1\ LMP}, F_{2\ LMP},\ F_{M\ LMP}]$

where M is the size of the feature vector.

Texture-based feature descriptors utilising complementary features of contrast and GF

The contrast measure and GF are used throughout this project to extract complementary features to be combined with the extracted features from the LZBP and LMP methods. For further details on the contrast measure and GF descriptors, refer to Subsections 2.3.2 and 2.2.2.2.

The complementary feature vectors of Contrast Feature (CF) and Gabor Filter (GF) are:

The feature vector representing the CF extracted features is $F_{CF} = [F_{1CF}, F_{2CF}, \dots F_{MCF}]$

The feature vector representing the GF extracted features is $F_{GF} = [F_{1GF}, F_{2GF}, \dots F_{MGF}]$.

where M is the size of the feature vector.

4.3.2 Feature-Level Fusion

The extracted features from the texture image by the proposed LZBP and LMP feature descriptors and the complementary CF and GF are obtained separately. In the subsequent stage, the improved features are obtained by combining between these sets of features together. In feature-level fusion, the final feature vector is constructed by concatenating the involved features.

Local LZBP and LMP features combined with complementary CF

The LZBP and LMP methods, as well as CF, are applied to the target image to form a feature vector. Then, LZBP features are concatenated with CF to obtain the fused LZBP and CF vector (Equation (4-1)). LMP features are also concatenated with CF to obtain the fused LMP and CF vector (Equation (4-2)).

$$F_{LZBP/CF} = [F_{1\,LZBP}, F_{2\,LZBP}, \dots F_{MLZBP}, F_{1CF}, F_{2CF}, \dots, F_{MCF}] \quad (4\text{-}1)$$

$$F_{LMP/CF} = [F_{1\,LMP}, F_{2\,LMP}, \dots F_{M\,LMP}, F_{1CF}, F_{2CF}, \dots, F_{MCF}] \quad (4\text{-}2)$$

Local LZBP and LMP features combined with complementary GF

The LZBP and LMP methods and the GF are applied on the target image to form a feature vector. Subsequently, LZBP and GF features are concatenated to obtain the fused LZBP and GF feature vector (Equation (4-3)). LMP features are also concatenated with GF features to obtain the fused LMP and GF feature vector (Equation (4-4)):

$$F_{LZBP/GF} = [F_{1\,LZBP}, F_{2\,LZBP}, \dots F_{MLZBP}, F_{1GF}, F_{2GF}, \dots, F_{MGF}] \quad (4\text{-}3)$$

$$F_{LMP/GF} = [F_{1\,LMP}, F_{2\,LMP}, \dots F_{M\,LMP}, F_{1GF}, F_{2GF}, \dots, F_{MGF}] \quad (4\text{-}4)$$

The previous fused feature vectors of Equations (4-1), (4-2), (4-3) and (4-4) are for single image. The previous operations will be repeated for the (N) images of the dataset, in order to obtain the feature fusion of the dataset. The complete feature size is therefore ($M \times N$), where M is the feature size, and N is the number of images in the dataset.

4.3.3 Selection Approach Based on the Hybrid ABC-NRS Algorithm

Usually, in any texture-based classification system, selecting the optimum features produces better results compared to the complete set of feature. In this work, feature selection is essential for optimising the features, and is done by selecting a subset of features from the combined features of different descriptors. This involves removing irrelevant features while retaining the same rate of classification.

Wrapper selection methods, such as the ABC method, take a long time to evaluate the potential optimum parts of features. To reduce the computation burden involved in searching the optimum parts of features, a hybrid selection approach was proposed in Subsection 3.3.2 for the proposed feature fusion of Subsection 4.3.2. This approach is based on using the ABC wrapper selection method based on the NRS method. Hybrid selection methods mostly produce good results when utilised for large feature space (Ke et al., 2008; Xiangyang Wang et al., 2007).

Implementation Scheme

The hybrid ABC-NRS feature selection method is applied to select the optimum features from the proposed feature-level fusion discussed in the previous subsection. The ABC algorithm is employed to select the relevant features from the LZBP and LMP extracted features that are combined with complementary contrast or GF features. However, multiscale LZBP and LMP produce a large feature space for ABC algorithm. The NRS method is an efficient tool extracting for optimum features from specific lengths of multiscale LZBP and LMP. In the proposed algorithm, explained in Fig 4.5, NRS reduces the features of multi-scale LZBP and LMP before said features are combined with the complementary features of CF and GF. The ABC algorithm is then applied to select optimum or improved features.

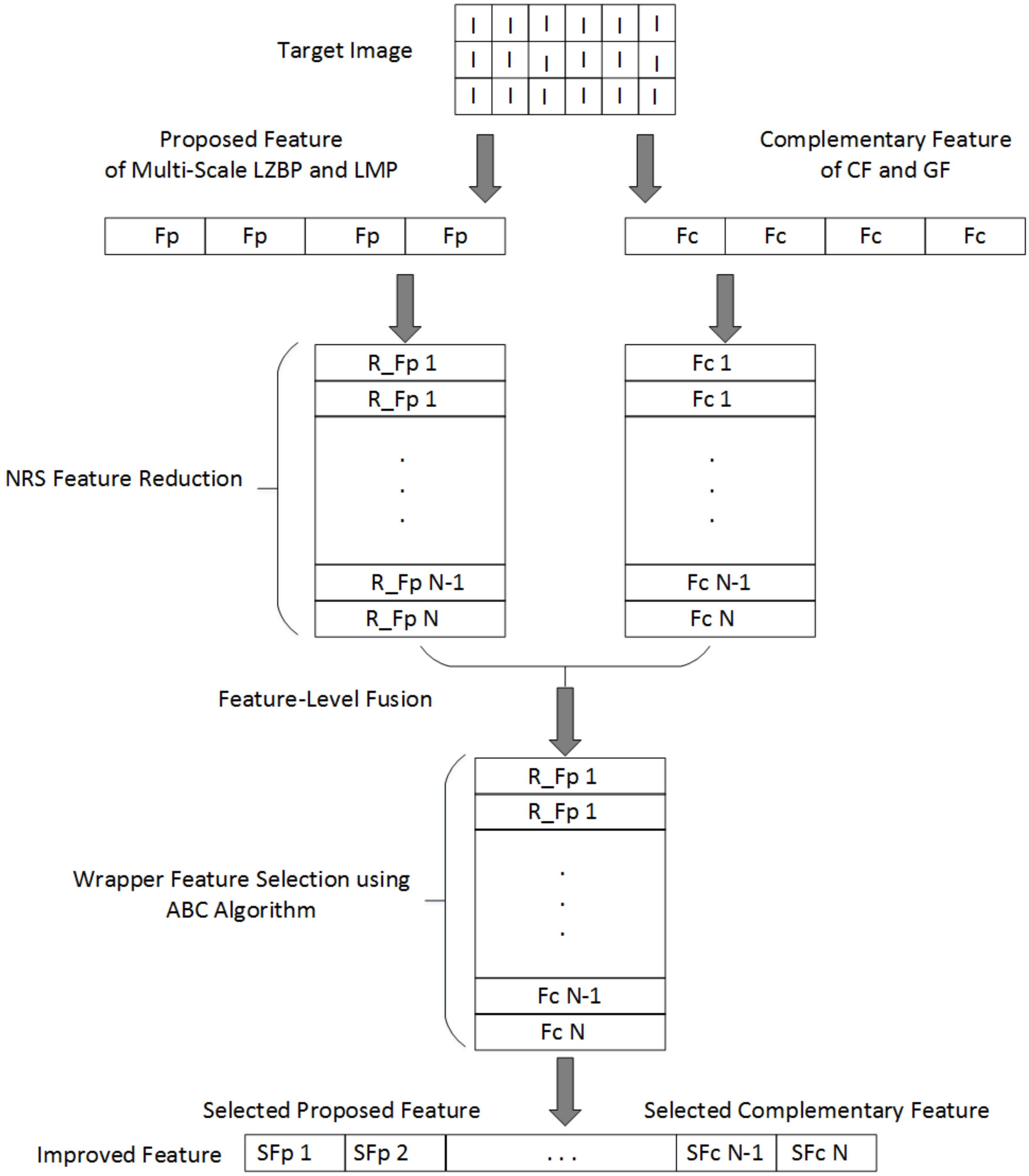


Fig. 4. 5 Hybrid feature selection method ABC-NRS for involved feature level fusion.

ABC-NRS Algorithm

Listing 4.3 shows the hybrid wrapper ABC-NRS algorithm used for the feature-level fusion of the proposed LZBP and LMP feature descriptors and the complementary feature descriptors (CF and GF).

Listing 4.3: Hybrid ABC-NRS algorithm

Set of control parameters of the ABC algorithm:

- Number of food sources
- Maximum Cycle Number (MCN)

Begine

$PF - Proposed\ Features\ \leftarrow NRS\ \big(FP\ optimal\ (s)\big): for\ LZBP\ and\ LMP\ with\ NRS\ algorithm$

$CF - Complementary\ Features\ \leftarrow CF\ optimal\ (s): for\ CF\ and\ GF$

Bounded parameters values:

$$FV_{feature\ vector\ min}=[V_{PF\ min}\ \ V_{CF\ min}]$$

$$FV_{feature\ vector\ max}=[V_{PF\ max}\ \ V_{CF\ max}]$$

Initial period

$\boldsymbol{for}\ s = 1\ to\ CS\ \boldsymbol{do}$

$P - intial\ (V_{PF}, \quad V_{CF}) \leftarrow Rand\ \big(V_{feature\ vector}\big) : Random\ solution\ consisting\ of\ feature\ vector$

$FV - fitness\ (\ intial) \leftarrow classifier\ \big(P - intial\ (V_{PF}, \quad V_{CF})\big) :\ Evaluate\ initial\ solution$

$\boldsymbol{end}$

$cycle = 1;$

$\boldsymbol{while}\ cycle < MCN\ \boldsymbol{do}$

Employed bees′ period

$\boldsymbol{for}\ s = 1\ to\ CS\ \boldsymbol{do}$

$P - employed\ (V_{PF}, \quad V_{CF}) \leftarrow neigh\ \big(P - intial\ (V_{PF}, \quad V_{CF})\big) : Exploring\ the\ neighbourhood$

$FV - fitness\ (\ employed\ ) \leftarrow classifier\ \big(P - employed\ (V_{PF}, \quad V_{CF})\big) : Evaluate\ new\ solution$

$$\boldsymbol{if}\ FV - fitness\ (\ employed\ ) < V - fitness\ (\ intial\ )\ \boldsymbol{then}$$

$$P - saved\ (V_{PF}, \quad V_{CF}) \leftarrow P - intial\ (V_{PF}, \quad V_{CF})$$

$$FV - fitness\ (\ saved) \leftarrow FV - fitness\ (\ intial\ )$$

$\boldsymbol{else}$

$$P - saved\ (V_{PF}, \quad V_{CF}) \leftarrow P - employed\ (V_{PF}, \quad V_{CF})$$
$$FV - fitness\ (\ saved) \leftarrow FV - fitness\ (\ employed\ )$$

$\boldsymbol{end}$

$\boldsymbol{end}$

Onlooker bees′ period

$FV - fitness\ (\ global\ ) \leftarrow FV - fitness\ (\ saved) :\ Calculate\ probability\ for\ onlooker\ period$

$P - global\ (V_{PF}, \quad V_{CF}) \leftarrow\ P - saved\ (V_{PF}, \quad V_{CF})$

$\boldsymbol{for}\ s = 1\ to\ CS\ \boldsymbol{do}$

$P - onlooker\ (V_{PF}, \quad V_{CF}) \leftarrow neigh\ \big(P - global\ (V_{PF}, \quad V_{CF})\big) : Exploring\ the\ neighbourhood$

$FV - fitness\ (\ onlooker\ ) \leftarrow classifier\ \big(P - onlooker\ (V_{PF}, \quad V_{CF})\big) : Evaluate\ new\ solution$

$\boldsymbol{if}\ FV - fitness\ (\ global\ ) <\ FV - fitness\ (\ onlooker\ )\ \boldsymbol{then}$

$$P - saved\ (V_{PF}, \quad V_{CF}) \leftarrow P - onlooker\ (V_{PF}, \quad V_{CF})$$
$$FV - fitness\ (\ saved) \leftarrow FV - fitness\ (\ onlooker\ )$$

$\boldsymbol{else}$

$$P - saved\ (V_{PF}, \quad V_{CF}) \leftarrow P - global\ (V_{PF}, \quad V_{CF})$$
$$FV - fitness\ (\ saved) \leftarrow FV - fitness\ (\ global\ )$$

$\boldsymbol{end}$

$$P - memorized\ (V_{PF}, \quad V_{CF}) \leftarrow max\ (P - saved\ (V_{PF}, \quad V_{CF}))$$

Scout bees′ period

$P - intial\ (V_{PF}, \quad V_{CF}) \leftarrow P - scout\ (V_{PF}, \quad V_{CF}) : Exploring\ other\ solutions$

$FV - fitness\ (\ initial\ ) \leftarrow classifier\ \big(P - scout\ (V_{PF}, \quad V_{CF})\big)$

$cycle + +;$

$\boldsymbol{end}$
$\boldsymbol{end}$

Explanation of algorithm:

For any selection algorithm applied for optimisation, all parameters have to be carefully considered since they have an impact on the performance of optimisation. ABC has few control parameters compared to other selection algorithms, which are the maximum number of cycles (MCN), and the colony size (or population size). The population size is determined by the number of food sources (or the number of bees). (Karaboga & Basturk, 2007).

To select the optimal features from the proposed LZBP and LMP feature descriptors combined with the complementary CF and GF feature descriptors, one can express the following:

$SN = \{Num\_PF, Num\_CF\}$

$D = \{Par\_PF, Par\_CF\}$

where Num_PF & Par_PF are the feature parameters of the proposed feature descriptors, Num_CF & Par_CF are the feature parameters of the complementary feature descriptors, SN denotes the size of the employed bees or onlooker bees, and D is the number of optimisation parameters.

The ABC algorithm consists of a number of periods (or stages), which are the initial period, employed bees period, onlooker bees period, and scout bees period.

(1) Initial period

In Step 1, after preparing the feature vectors, the number of features values are selected randomly as an initial solution, using Equation (2.16). The initial period process starts by selecting the features that will submitted to the classifier. In our method, the parameter values that exceed the limited band in the random selection take the limit values.

In Step 2, the objective function is used to evaluate the initial solutions of the parameter values. The algorithm iteratively generates the parameter values of the feature extraction methods for evaluation by the classifier as a possible optimum solution.

(2) Employed bees period

In Step 1, the employed bees start locally searching in the vicinity of the initial solutions, in order to obtain more important parameter values than the initial parameter values. These new parameter values are calculated by randomly changing the values of one parameter and keeping the reminders unchanged according to Equation (2.17).

In Step 2, the new possible solutions that result from the modified parameters values are also evaluated based on the classification accuracy results of classifier.

In Step 3, these parameter values, which are possible solutions, are updated by greedy selection between initial parameter values and the discovered vicinity parameter values, based on whichever achieved the better classification accuracy.

(3) Onlooker bees period

In Step 1, following the greedy selection and finding new sets of local solutions, the highest probability is calculated using the Roulette Wheel Selection (RWS) equation. The equation extracts the global solution from previous sets of local solutions, which result from the greedy selection in the employed bees period. Equation (2.18) computes the highest probability of parameters by their image classification accuracy, where the probability of a selected parameter value being a possible solution increases by increasing its repeated accuracy.

In Step 2, onlookers start searching in the vicinity of the parameter value with the highest probability, in order to find better optimum solutions (Equation 2.17).

In Step 3, the new possible solutions that result from the modified values of parameters are also evaluated based on the classification accuracy results of the classifier.

Finally, in Step 4, these parameter values, which are possible solutions, are updated by greedy selection between the initial parameter values and the discovered vicinity parameter values, based on whichever achieved the better classification accuracy.

(4) Scout bees period

In Step 1, the employed bees and onlookers continue exploiting the identified places, and if there is no further improvement, the final parameter values with the best classification accuracy can be saved (i.e. memorised).

In Step 2, the employed bees in the abandoned places, which consist of dimensional parameter values, are converted into scout bees in order to explore other places without guidance, by randomly selecting and not repeating other parameter values from Equation (2.16).

4.4 Algorithm for Integrating Feature Components

This section describes the general algorithm for integrating the components used in improving feature extraction in texture-based images classification (Fig 4.1). The discriminative feature extraction capability of LZBP and LMP is utilised and combined with complementary contrast measure and GF features, while only retaining the relevant features to provide powerful and descriptive features of texture.

INPUT

Group of images in grey-level**.**

OUTPUT

Classified images into a set of classes according to their similarity.

FEATURE PROCESS COMPONENTS

First component: feature extraction

Step 1.a: Applying the proposed multiscale descriptors for (N) images in (M) classes.

LZBP: Extracting features by Algorithm (4.1).

Calculating the features through an LZBP histogram by Equations (3-4) and (3-5)

LMP: Extracting features by Algorithm (4.2).

Calculating the features through an LMP histogram by Equation (3-7)

Step 1.b: Applying the complementary features of descriptors for (N) images in (M) classes.

CF: Extracting features through a histogram.

GF: Extracting features by (2.12), then apply mean and standard deviation.

Second component: feature-level fusion

Step 2.a. Applying the feature-level fusion for LZBP with CF using Equation (4-1).

Applying the feature-level fusion for LMP with CF using Equation (4-2).

Step 2.b. Applying the feature-level fusion for LZBP with GF using Equation (4-3).

Applying the feature-level fusion for LMP with GF using Equation (4-4).

Third component: feature selection

Step 3.a: Applying feature-level selection by Algorithm (4.3) for Step 2.a.

Step 3.b: Applying feature level selection by Algorithm (4.3) for Step 2.b.

4.5 Summary

This chapter presented the implementation of the designed system used to harvest improved features through a set of processes related to texture feature extraction methods. The implementation included the methods designed to improve feature extraction in texture classification. The applied LZBP and LMP algorithms were explained, where the purpose of these descriptors is to detect discriminative features before they are combined with other complementary features, where a unified vector is used to integrate between both feature sets. To avoid the high dimensionality problem resulting from the combined features, a new hybrid feature selection algorithm based on ABC-NRS was implemented. These methods are anticipated to work together to make the resulting features more powerful than those resulting from other classical methods. To verify such a hypothesis, the system will be tested and evaluated, which will be done in the next two chapters.

Chapter 5

Experimental Results and Discussion

This chapter reports the results of implementing the methods proposed in the previous chapter to improve features extraction. The robustness of the new LZBP and LMP feature descriptors are assessed through practical experiments. Furthermore, the new texture descriptors are also experimentally tested when integrated with other complementary feature descriptors, and the processed fused features are assessed using the new selection approaches. This chapter also discusses the improvement that the proposed methods contribute to texture feature extraction.

Section 5.1 starts the chapter by introducing the test classification framework of the improved feature extraction methods. The first set of experimental results and a discussion of the proposed methods are presented in Section 5.2. The testing starts with evaluating the new LZBP and LMP texture feature descriptors, before testing the approaches developed to improve the result of texture-based image classification when dealing with a variety of texture characteristics. These approaches include integrating LZBP and LMP with the contrast measure of the image, or with GF. These approaches apply feature-level fusion, before extracting only the relevant features using the new selection methods. Finally, Section 5.3 concludes the chapter by providing a summary of its findings.

5.1 Tests Framework

In order to examine the performance of the new descriptors, as well as the proposed feature fusion approaches and the new feature selection method, different texture images are available from databases and can be therefore used as a benchmark (L. Liu, Fieguth, et al., 2016). In this research, image databases of varying complexity of texture surfaces were applied in the experiments, and some of these databases were prepared to be used for classification testing (see Section 3.2). The images in the databases were intended to be used to measure the robustness and performance of the developed methods when compared to other feature extraction methods on different classification challenges such as changes in rotation, scale, illumination, and viewpoint, and the robustness to against noise and blur problems. The proposed methods focus on improving the accuracy and feature length resulting from the extraction methods in classification systems.

The tested methods in this work are used to extract features from these databases, where the databases consist of a number of classes, and each class has a number of sample images. Table 5.1 shows a summary of the parameter set of datasets used for texture classification, which are UIUC (Lazebnik et al., 2005), UMD (Xu et al., 2009), Brodatz (Laine & Fan, 1993), KTHTIP2b (Mallikarjuna et al., 2006) and Outex_TC11 (Ojala, Maenpaa, et al., 2002). They were prepared for conducting the experimental testing on the implemented feature components described in the previous chapter.

Table. 5. 1 A brief summary of databases benchmark parameters used in the experimental study.

Database Benchmark	UIUC	UMD	Brodatz	KTHTIP2b	Outex_TC11
Number of images per class	40	40	40	100	40
Number of classes	25	25	25	11	24
Image resolution	640 × 480	640 × 480	256 × 256	200 × 200	128 × 128

The classification is conducted by extracted features from the images. This methodology constitutes markedly less complexity and computation burden than depending on the original input data, where classification is conducted by measuring distances between features of the same class.

BPNN and SVM are supervision classifiers that are used to evaluate the performance of feature extraction algorithms. These classifiers are also needed for wrapper approaches like the ABC algorithm, where they are essential for evaluating the selected subset of features through an evolutionary training process. For classifier testing, MATLAB tools implementing the BPNN and SVM classifiers were used. In testing, BPNN had a single hidden layer, where the number of nodes in the hidden layer was chosen to be n = 80, which selected along with activation function after set of testing through designing the classifier. SVM is generally adopted to deal with multi-class texture classification problems through a one-class-against-others approach, and the parameters values of SVM are based on the best accuracy. The classification performance of BPNN and SVM is calculated through the Classification Accuracy (CA), where CA is the percentage of the correct classified samples divided by the total samples.

In a classifier, the images samples in the dataset are divided into a training set and a testing set. For classifier evaluation, k-fold cross validation is an unbiased estimator that is employed to assess the statistical relevance of classifiers (Bengio & Grandvalet, 2004). k-fold cross validation is also beneficial with a limited number of training sets and in avoiding overfitting problems (Tu, 1996). Feature selection also reduces the overfitting problem by reducing the features dimensionality (Guyon & Elisseeff, 2003) . In the k-fold cross validation, the features were chosen randomly. In the 5-fold cross validation, when a feature is extracted from n samples of texture images, (n×80%) of the samples are used for training, and (n×20%) of the samples are used for testing, whereas in the 10-fold cross validation, (n×90%) of the samples are used for training, and (n×10%) of the samples are used for testing. The 10-fold cross validation is more accurate, but takes a longer time. The average of accuracy of k-folds is used to assess the performance of the classifier.

5.2 Results and Discussions

5.2.1 LZBP and LMP Feature Descriptors

LZBP and LMP, along with other common feature extraction methods, were tested on a number of databases using BPNN and SVM for the 5-fold cross-validation and 10-fold cross-validation. According to results displayed in Table 5.2, it can be observed that, for the 5-fold cross-validation, the LZBP descriptor produced the highest accuracy of 87.4%, followed by LMP with 82.3%, and GF with 72.1%, for the UIUC dataset. When testing on the UMD database, LMP gave the highest accuracy of 94.06%, followed by both LZBP and LBP with 93.72 % and 93.3%, respectively, and finally GF with 87.8 %. LMP again showed the highest accuracy of 87.526% for the KTHTIP2b dataset, thus confirming the high efficiency of the proposed descriptors. Methods such as GLDM, GLDM, and TS recorded the lowest accuracies of no more than 65% using BPNN. For the Brodatz dataset, which is less challenging compared to the previous databases, these methods recorded higher accuracy (over 80 %), but were still behind the accuracy of LBP, LZBP and LMP (over 95 %).

Table. 5. 2 Classification accuracy results of feature extraction methods that tested on a number of databases using BPNN and SVM with 5-fold cross validation.

Methods	BPNN Classifier				SVM Classifier			
	UIUC	**UMD**	**KTH**	**Brodatz**	**UIUC**	**UMD**	**KTH**	**Brodatz**
Proposed TU:								
LZBP	87.4	93.72	74.145	95.52	89.58	94.38	74.363	95.86
LMP	82.3	94.06	87.526	96.62	85.48	95.4	91.1086	96.4
Co-occurrence:								
GLCM	63.54	68.94	63.126	85.94	70.4	75.52	64.2722	87.52
GLDM	44.84	62.28	55.217	90.86	64.76	80	73.654	91.38
Texture Unit:								
LBP	77.22	93.3	85.545	95.1	80.18	94.62	87.6722	94.92
TS	58.2	77.08	61.799	87.6	58.82	78.76	64.9994	90.22
Signal Processing:								
WT	64.5	71.3	56.727	82.3	68.3	74.6	64.0000	87.3
GF	70.7	87.8	83.636	97.1	77.1	91.4	87.0909	97.4

According to the results presents in Table 5.3, some improvement in the accuracy can be seen using the 10-fold cross-validation instead of the 5-fold cross-validation. However, the 10-fold cross-validation results in an increased computation cost. In the case of the 10-fold cross-validation, the performance of LBP and LZBP increased by 0.626% and 1.151%, respectively, using BPNN; and 0.903 % and 0.432 %, respectively, using SVM. The effect of the 5-fold cross-validation is thus slightly smaller than the 10 fold cross-validation. The results also show that the ranking of the methods stay the same, where LZBP, LMP, LBP and GF outperformed other methods such as GLCM. GLDM and TS.

Table. 5. 3 Classification accuracy results of feature extraction methods that tested on a number of databases using BPNN and SVM with 10-fold cross validation.

Methods	BPNN Classifier				SVM Classifier			
	UIUC	**UMD**	**KTH**	**Brodatz**	**UIUC**	**UMD**	**KTH**	**Brodatz**
Proposed TU:								
LZBP	88.5	93.98	76.690	96.22	90.28	94.88	74.672	96.08
LMP	84.94	95.7	88.890	97.26	87.18	96.4	91.363	96.96
Co-occurrence:								
GLCM	66.42	70.22	64.163	86.56	71	76.98	64.435	88.06
GLDM	47.2	66.36	55.345	91.48	66.48	81.32	73.781	92.34
Texture Unit:								
LBP	78.38	93.84	85.690	95.76	81.88	95.18	88.726	95.22
TS	59.4	78.18	62.799	88.76	59.64	79.48	65.363	90.68
Signal Processing:								
WT	64.4	73.1	59.909	83.9	67.9	74.5	64.545	86.5
GF	73.8	87.3	83.090	97.9	78.8	91.6	88.636	97.5

Results of Multiscale LZBP and Multiscale LMP

The proposed LZBP and LMP descriptors are based thresholding intensity values of neighbour pixels of the central pixel. In our experiment, the performance of multi-scale LZBP and LMP was compared with that of multi-scale LBP (Ojala, Pietikainen, et al., 2002). Multi-scale LBP is recommended in many applications for extracting more information from different scales of images.

The classification results of the LZBP, LMP, and LBP methods on the utilised databases are presented in Table 5.4 for UIUC, Table 5.5 for UMD, Table 5.6 for KTHTIP2b, and Table 5.7 for Brodatz. The results show that LZBP and LMP outperformed the LBP descriptor. LZBP, LMP is affected by different values of radii in the scale analysis, as is the case with LBP, especially from one radii (8,1) to two radii (8,1+8,2). On average, using the UIUC, UMD, Brodatz, and KTHTIP2b datasets, an improvement of about 2.175 % for LZBP, 3.075 % for LMP, 3.375 % for LBP was obtained. Changing from two radii (8,1+8,2) to three radii (8,1, + 8,2, + 8,3) results in a reduced level of improvement, particularly for the LZBP.

Table. 5. 4 Classification accuracy results for UIUC database of LBP/LZBP/LMP descriptors individually, when combined with complementary contrast features (c), and when all features are combined together.

Resolution	C	LBP	LZBP	LMP	LBP-C	LZBP-C	LMP-C	All-F
8,1,	61.7	76.6	88.6	83.2	85.1	91.8	89.6	89.5
8,2,	65.8	76.3	91.3	87.9	84.2	91.7	90.2	92.7
8,3,	65.0	77.0	92.0	84.6	83.8	92.9	90.1	91.4
8,1+8,2,	62.2	76.6	93.6	87	85.2	92.0	91.1	92.9
8,2,+ 8,3,	64.6	80.0	93.4	87.3	86.6	92.2	90.4	93.3
8,1,+8,3,	65.6	80.5	93.5	89.0	86.6	93.0	91.9	93.1
8,1,+8,2,+ 8,3,	66.1	80.8	91.5	88.3	87.7	93.1	90.4	92
Average	64.429	78.257	91.986	86.757	85.600	92.386	90.529	92.129

Table. 5. 5 Classification accuracy results for UMD database of LBP/LZBP/LMP descriptors individually, when combined with complementary contrast features (c), and when all features are combined together.

Resolution	C	LBP	LZBP	LMP	LBP-C	LZBP-C	LMP-C	All-F
8,1,	67.7	91.9	95.1	94.5	94.9	94.3	95.0	97.3
8,2,	71.2	93.0	94.8	94.3	95.4	95.4	95.9	98.0
8,3,	69.1	94.5	96.2	95.8	96.1	96.1	96.8	97.0
8,1+8,2,	68.6	95.5	96.3	96.8	97.4	96.4	96.3	97.9
8,2,+ 8,3,	70.0	96.0	96.8	95.5	95.1	96.7	97.1	98.1
8,1,+8,3,	70.3	96.8	96.9	97.1	96.6	96.0	97.6	98.1
8,1,+8,2,+ 8,3,	67.5	97.3	97.5	97.6	96.7	95.7	96.8	98.2
Average	69.2	95	96.23	95.943	96.03	95.8	96.5	97.8

Table. 5. 6 Classification accuracy results for KTHTIP2b database of LBP/LZBP/LMP descriptors individually, when combined with complementary contrast features (c), and when all features are combined together.

Resolution	C	LBP	LZBP	LMP	LBP-C	LZBP-C	LMP-C	All-F
8,1,	43.9	85.9	74.7	87.3	85.8	76.9	86.4	91.4
8,2,	45.4	82.3	71.5	85.1	85.6	75.2	71.2	91.1
8,3,	48.0	80.5	66.7	81.2	83.6	73.9	84.6	88.9
8,1+8,2,	46.4	88.2	79.9	92.6	89.2	78.7	90.7	91.0
8,2,+ 8,3,	47.1	86.0	72.9	84.9	82.0	76.9	88.3	91.1
8,1,+8,3,	47.8	86.2	79.2	88.9	87.6	79.0	87.5	92.2
8,1,+8,2,+ 8,3,	42.6	90.4	77.1	91.6	87.8	78.1	91.0	92.6
Average	45.895	85.642	74.57	87.371	85.94	76.957	85.67	**91.**19

Table. 5. 7 Classification accuracy results for Brodatz database of LBP/LZBP/LMP descriptors individually, when combined with complementary contrast features (c), and when all features are combined together.

Resolution	C	LBP	LZBP	LMP	LBP-C	LZBP-C	LMP-C	All-F
8,1,	81.90	94.50	96.50	96.400	97.40	97.3000	97.80	98.40
8,2,	80.10	97.80	97.80	97.300	98.70	97.8000	98.60	99
8,3,	81.70	96.10	97.60	97.100	98.70	98.8000	98.90	99.40
8,1+8,2,	81.60	97.30	97.40	98.500	97.90	98.1000	98.60	99.20
8,2,+ 8,3,	80.70	96.90	97	98.200	98.50	98.1000	98.90	99.10
8,1,+8,3,	80	97.30	98.10	98.600	99.30	97.8000	98.60	98.40
8,1,+8,2,+ 8,3,	80.10	98.20	97.60	98.400	98.90	98.2000	99.30	99.30
Average	80.871	96.871	97.429	97.786	98.486	98.014	98.671	98.971

Figure 5.1 shows the average classification accuracy of the three multiscale descriptors over all datasets. The use of three radii shows less improvements over all datasets compared to the original multi-scale approach. This could be due to the high features dimensionality. The accuracy increased from 88.422 % to 91.642 % when applying two radii instead of one radii, and from 91.642 % to 92.192 % when employing three radii.

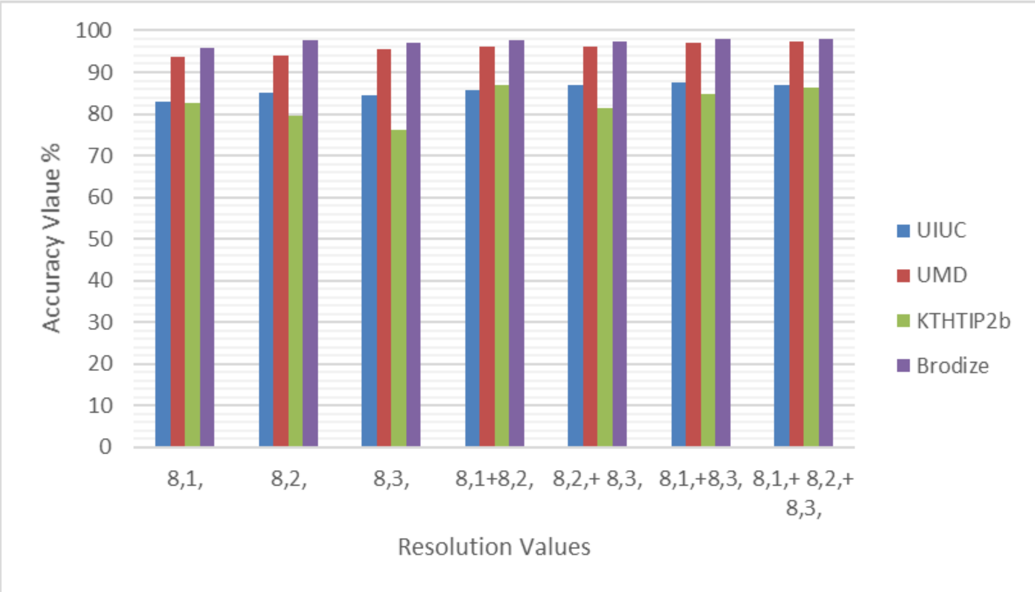


Fig. 5. 1 The average classification accuracy value of multi scales LBP, LZBP and LMP when tested on a number of databases.

Subsequently, the performance of these descriptors was evaluated on each dataset separately. Fig 5.2 shows the classification accuracy using the UIUC dataset. The data shows that LZBP outperformed the LBP and LMP descriptors in different multi-scales states. Fig 5.3 shows that the results of the UMD database are more affected by increasing the multi-scale levels. For instance, the highest possible accuracies obtained vary between LZBP and LMP in different multi-scales cases, with multi-scale LBP being consistently outperformed. Fig 5.4 shows the results of the KTHTIP2b database, where both LBP and LMP outperformed LZBP. Finally, Fig 5.5 show the accuracy results for the Brodatz database, where in single scale descriptors, LMP was markedly outperformed by the other two descriptors, whereas in multiscale, a significant improvement in the results of LMP was observed.

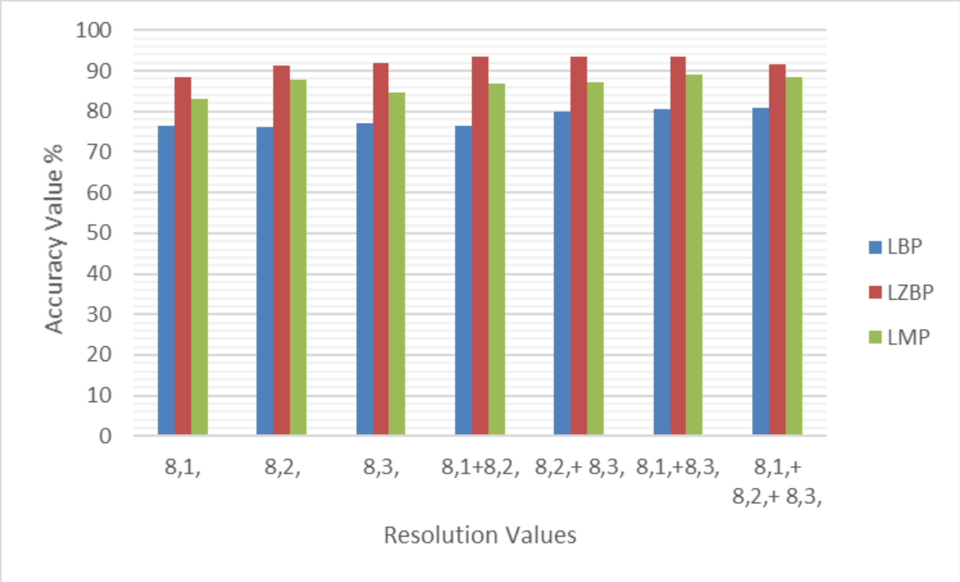


Fig. 5. 2 The classification accuracy value of multi scales LBP, LZBP, and LMP on the UIUC database.

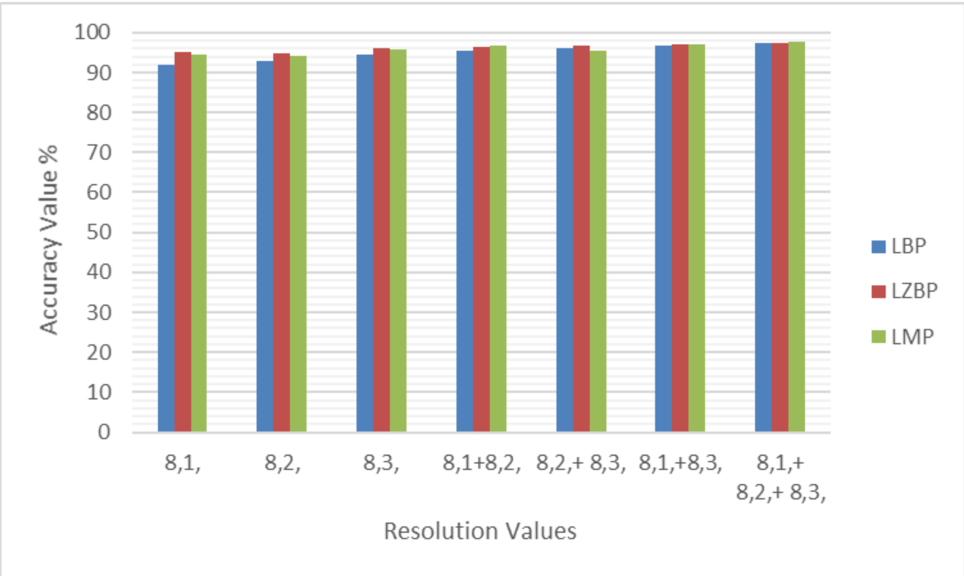


Fig. 5. 3 The classification accuracy value of multi scales LBP, LZBP, and LMP on the UMD database.

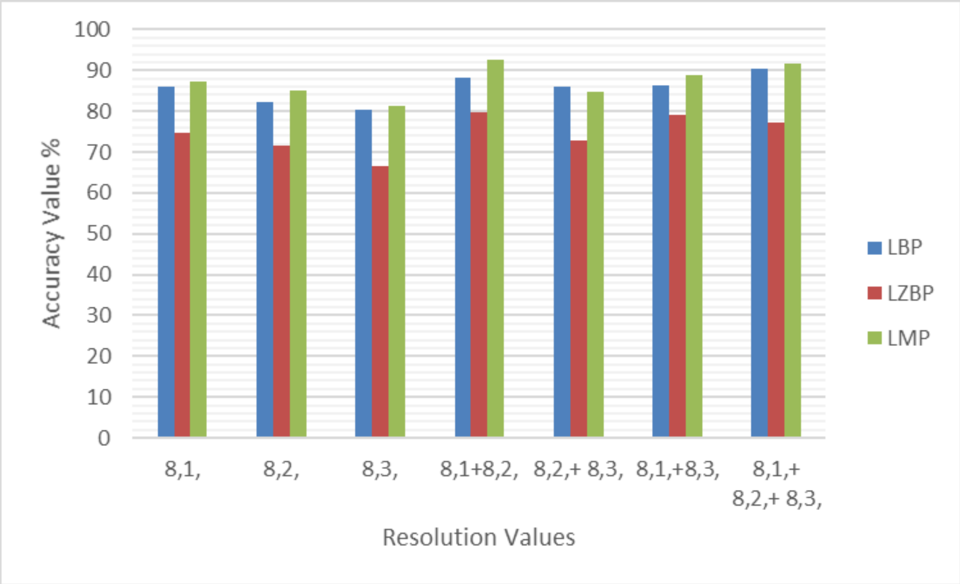


Fig. 5. 4 The classification accuracy value of multi scales LBP, LZBP, and LMP on the KTHTIP2b database.

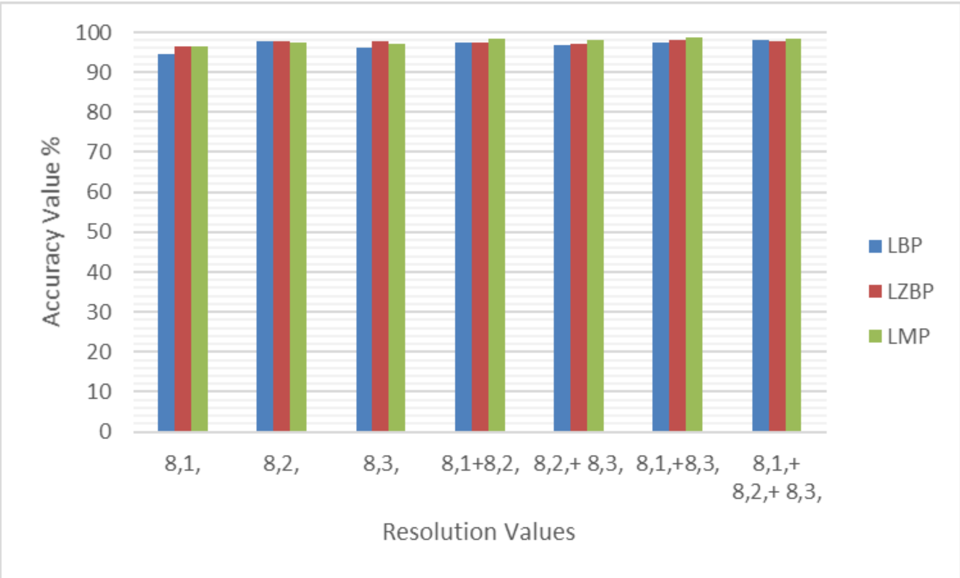


Fig. 5. 5 The classification accuracy value of multi scales LBP, LZBP, and LMP on the Brodatz database.

From testing, it can therefore be concluded that every texture database produces the highest possible accuracy for its own unique scale, and that no fixed radii can be uniformly and universally used to improve the accuracy. Applying these descriptors in multi scale analysis produced different results, where some descriptors improved their accuracy with multiscale more than others. Furthermore, the previous experimental results showed that every database matched with a specific descriptor can produce better image classification accuracies than others. This means that finding the appropriate descriptor along with a suitable scale is important for improving feature extraction, as this result in an effective classification performance when dealing with different texture characteristics.

5.2.2 LZBP and LMP Combined with Contrast Measure

This section starts by testing LZBP and LMP with a complementary feature descriptor based on contrast measure. This includes testing direct feature-level fusion, and the performance of the new feature selection approach, which is proposed for feature-level fusion and selecting only the relevant features from the extracted feature set. Subsequently, a discussion of the results is presented, based on a comparing the improvement that these features methods bring to texture-based image classification. In the experimental study, the accuracy and feature size are important when the feature length is reduced and compared to direct feature-level fusion. The tests also cover the performance of the methods when exposed to blurred and noisy images.

5.2.2.1 Test Results

Results of Features-level Fusion

LBP has already been combined with feature contrast measurements to improve the discriminative feature extraction capability of LBP in texture classification applications, as the classification accuracy results are generally improved following such an approach (Ojala et al., 1996). Here, the effect of combining contrast features with the LZBP and LMP descriptors is investigated. In other words, the LBP, LZBP, and LMP features are combined together with contrast measure by feature-level fusion to examine the possibility of achieving a classification rate improvement compared to applying each descriptor separately. In this approach, the total feature histogram is achieved by concatenating the different histograms resulting from different scales analyses.

Large scale analysis is used to distinguish the different texture patterns, whereas conventional LBP works in a small window setting (Ojala et al., 1996). Regarding multi-scale analysis, Ojala et al. explored three LBP scales based on 1,2 and 3 radii (Ojala, Pietikainen, et al., 2002). Applying more than eight pixels results in a significant increase in feature length (2^9=512), since the feature length grows exponentially with the number of pixels. This can be avoided by depending on constant neighbouring pixels, which do not have much effect on the discrimination of features (L. Liu et al., 2017). To reduce the computation cost in our testing, the same strategy was followed based on three scales with a constant number of neighbour pixels.

The previous Tables (5.4 to 5.7) also represent the experimental results after combining the features of these descriptors with contrast features. For the UIUC database, difference in accuracy between applying LBP, LZBP, and LMP separately and combining them with texture image contrast was 8.5%, 3.2%, and 6.4%, respectively. However, combing all features of LBP, LZBP, LMP and contrast resulted in 2.3% less accuracy than LZBP-C. For the UMD database, combining with contrast only improved the accuracy of LBP (by 3%) and LMP (by 0.5%), while reducing the accuracy of LZBP from 95.1 % to 94.3%. In the same database, combining different features of LBP, LZBP, LMP and contrast improved the accuracy by 2.4%, 3% and 2.3% in comparison to LBP-C, LZBP-C, and LMP-C, respectively. The results of the KTHTIP2b database demonstrate that combining with contrast only improved the LZBP descriptor from 74.7% to 76.9%, whereas combining all features also improved the accuracy by 5.6%, 14.5% and 5% on LBP-C, LZBP-C, and LMP-C, respectively.

Once can notice that, despite the multiscale results being different, depending on all multiscale features produced more important results when taking into consideration different databases. We expect that the improvement from combining different features will be better by depending on a diverse set of features. However, with this approach, a solution to the huge feature size must be found.

Results of Feature Selection

In order to improve the classification performance, we depended on the diversity of information resulting from feature-level fusion. However, to avoid the problem of dimensionality, the proposed feature selection approaches were utilised to retain only the relevant features for classification.

The combined features from multiscale descriptors were processed by ABC, which is the new wrapper selection method. ABC was applied for selecting the relevant features based on the appropriate descriptors and scales from the involved multiscale descriptors (see Subsection 5.2.1).

The involved descriptors were applied at different scales based on three radii (1,2,3) and a fixed number of neighbour pixels (n=8), where the ABC selected a combination of appropriate scales from the involved different descriptors. According to the results presented in Table 5.8, in all cases, the ABC method was successful in improving the classification rate compared with joining all scales directly. For the UIUC dataset, using BPNN, the accuracy by relevant features compared to complete feature of LBP-C, LZBP-C and LMP-C improved the accuracy by 1.6%, 1.8%, and 2.4%, respectively. The least improvement was observed in the Brodaze dataset, which was only 0.5 % for LBP-C and LMP-C, and 1% for LZBP-C. For the UMD dataset, higher improvement levels of 1.4% for LBP-C, 2.4% for LZBP-C, and 1.6% for LMP-C were obtained. Some of the highest overall improvement levels were seen using the KTHTIP2b dataset, where improvements of 5.62% for LZBP-C, 4.29% for LBP-C, and 1.18 % for LMP-C were achieved. The results of the SVM classifier were similar to those of BPNN.

Table. 5. 8 Classification accuracy and feature dimensionality results of applying ABC algorithm on multiscale descriptors for selecting suitable features scales.

Methods	Dataset	BPNN Classifier		SVM Classifier	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**
LBP-C	UIUC	89.3	1024	92.2	768
	UMD	98.1	1280	97.9	1024
	BRODAZE	99.4	1024	99	512
	KTHTIP2b	92.09	1024	94.36	1024
LZBP-C	UIUC	94.9	802	96	1058
	UMD	98.1	802	98	802
	BRODAZE	99.2	620	99.4	620
	KTHTIP2b	83.72	802	88.09	620
LMP-C	UIUC	92.8	1792	94.4	1792
	UMD	98.4	1792	98.5	1792
	BRODAZE	99.7	1280	99.1	1280
	KTHTIP2b	92.18	1280	94.27	1792
LBP-LZBP-LMP-C	UIUC	95.2	1570	95.5	1826
	UMD	99	1826	98.4	1826
	BRODAZE	99.8	1206	99.4	1826
	KTHTIP2b	94.4	1462	95.3	3106

From the results, the ABC method clearly succeeded in selecting suitable scales from different multiscale descriptors. ABC improved the accuracy of classification compared to depending on different scales of descriptors by utilising a smaller feature size. The new hybrid wrapper ABC algorithm based on the NRS filter method was subsequently tested, where NRS was used to select the relevant features from those selected by the ABC algorithm.

Table 5.9 lists the results obtained from using the hybrid ABC-NRS method with multi scales features. As can be seen from the table, in most cases, the hybrid selection method improved the classification rate compared to directly combining features. For example, for the UIUC database, using BPNN, an accuracy improvement of 1.6%, 0.4% and 0.4% was obtained for LBP-C, LZBP-C, and LMP-C, respectively. This improvement was reduced compared to using ABC only, as the hybrid selection method brought further improvement by reducing the feature length, as the results of Tables 5.5 and 5.6 testify. This occurred for almost all of the datasets tested.

Table. 5. 9 Classification accuracy and feature dimensionality results of applying ABC-NRS on multiscale descriptors for selecting suitable features scales.

Methods	Dataset	BPNN Classifier		SVM Classifier	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**
LBP-C	UIUC	89.3	432	91.5	428
	UMD	97.1	321	98	319
	BRODAZE	99	306	99.2	258
	KTHTIP2b	90.72	335	92.82	471
LZBP-C	UIUC	93.5	186	95.9	190
	UMD	97.9	208	97.6	208
	BRODAZE	98.7	191	99.4	212
	KTHTIP2b	83.27	182	87.72	217
LMP-C	UIUC	90.8	389	93.5	385
	UMD	97.5	344	98	344
	BRODAZE	99.4	307	99.3	303
	KTHTIP2b	89.09	498	92.73	502
LBP-LZBP-LMP-C	UIUC	94.2	429	95.1	375
	UMD	99	513	98.2	340
	BRODAZE	99.8	648	99.6	613
	KTHTIP2b	92.6	633	94.6	394

5.2.2.2 Discussion and Comparison of Results

Figures 5.6 and 5.7 show a comparison of classification accuracy result vs. feature length. The comparison is based on the different feature methods that were applied in testing by the BPNN classifier. Several image databases were used for the tests, and the comparison was based on the average of the results of all databases. Figures 5.6 and 5.7 illustrate the various accuracy and feature length results obtained from applying the different descriptors separately, as well as from combining the participating feature descriptors together.

When applying the LBP, LZBP, and LMP texture descriptors individually without combing with other features, LMP recorded the highest average accuracy on the tested datasets. However, although LMP was applied by 8-bit diagonal and non-diagonal patterns instead of the 16-bit patterns for a shorter feature length, it still had the longest feature length (see Fig 5.7). As such, comparatively, the descriptor with the longest feature length recorded the highest classification accuracy for the datasets used.

In general, the classification accuracy of these local texture descriptors achieved improved feature extraction for texture classification. Furthermore, there were examples of each local descriptor performing well with a specific dataset, such as LZBP being more effective with UIUC, whereas LBP achieved much better results than LZBP for KTH database. A further step to improve classification accuracy could be realised by combining the involved descriptors, where feature-level fusion might improve clarification accuracy.

Direct fusing between different feature descriptors improved the classification accuracy against individual application of LBP, LZBP, and LMP by 3.85%, 4.6%, and 1.55%, respectively, where all shared features descriptors were based on three scales. While it is clear that there is a slight improvement, one should not ignore the fact that a major drawback of combining feature descriptors is the resulting redundant features, as it is possible that part of the relevant features before fusion become irrelevant after fusion with other feature descriptors.

To address this, the proposed ABC-based feature selection method can be utilised to extract only the relevant features from the combined LBP, LZBP, and LMP feature descriptors. An improvement in the classification rate can be seen from Fig 5.6 when selecting parts of the features from the whole feature set using ABC. The difference in accuracy was 1.56591%. Furthermore, from Fig 5.7, the feature size was reduced from 3189 to 1516 as a result of applying ABC. The reduction in feature size resulted in an enhancement in classifier time. The feature length was reduced by ABC by selecting a number of the shared histograms instead of concatenating all shared histograms of multiscale descriptors. However, it is clear that the reduced feature length is still not short enough to match the feature length of any individual descriptor (i.e. LBP, LZBP, or LMP). The ABC-based feature selection followed a strategy based on utilising the complete histogram from every selected scale of shared descriptors. In other words, ABC did not employ only the relevant features from the selected histograms. However, depending on the complete histograms of the selected descriptors may not yield a completely relevant set of features (Ojala et al., 1994).

In the hybrid ABC-NRS method, depending on filter methods such as NRS results in a reduced computational cost for specific feature sizes. However, it must be noted that using filter methods alone with a long feature space is also not feasible (Xiangyang Wang et al., 2007). A hybrid method based on combining ABC with NRS is a more effective approach for reducing feature length, where NRS is applied to extract the relevant features from selected histograms. In this experiment, the feature length was indeed reduced by applying the hybrid selection method. The aim of the hybrid selection approach is to exploit the advantages of both ABC and NRS for feature selection to yield the best possible performance. In the hybrid method, NRS was applied to select the relevant features from the histograms of shared descriptors to produce pool of relevant features. The ABC as wrapper method was then applied to find the optimal subset of features from the relevant histogram features obtained by NRS. This makes feature selection faster since the filter method works under a specific feature length, which is the fixed histograms length resulting from the different combined multiscale descriptors.

As can be seen from Fig 5.7, compared to the direct feature fusion, the hybrid selection method achieved a significantly improved feature length of the combined descriptors by selecting the optimum features. In other words, depending on the hybrid ABC-NRS method reduced the feature size of direct feature fusion. The maximum difference between the average feature size resulting from using ABC and using ABC-NRS is around 960, which reduced the processing time of the classifier. This was the shortest feature length compared to applying LBP, LZBP, and LMP separately. However, one can notice that a slight reduction in classification accuracy of 0.96% was observed from applying the hybrid ABC-NRS instead of depending on ABC alone for feature selection.

From the comparison of the results, it is clear that utilising either ABC alone, or the hybrid ABC-NRS method as feature selection methods reduced the feature length by selecting relevant features from LBP, LZBP, and LMP. Fig 5.6 shows that the hybrid ABC-NRS was more effective with multi-scale feature length, at the expense of a slightly lower accuracy than the ABC performance. However, the proposed hybrid selection method (using ABC-NRS) still provided a better performance than direct fusion of different feature descriptors, as the former yielded a much smaller feature length.

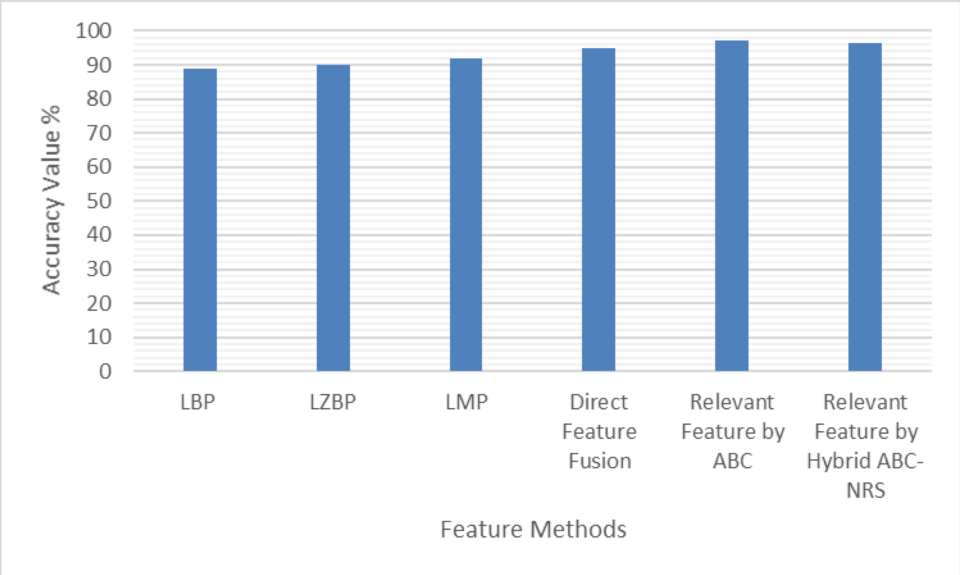


Fig. 5. 6 Comparison the classification accuracy value of LBP, LZBP, LMP, feature fusion of these descriptors, and relevant feature by ABC alone and hybrid ABC-NRS.

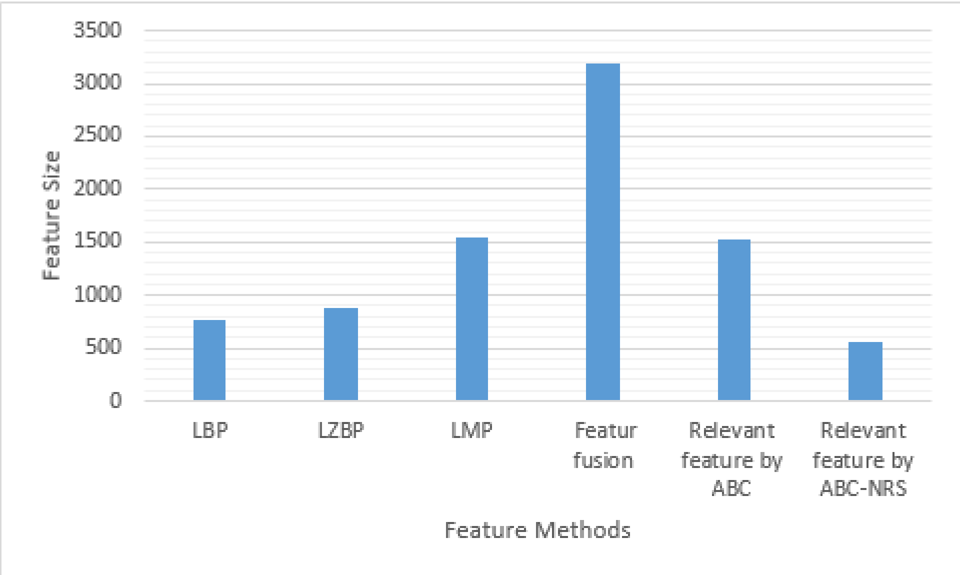


Fig. 5. 7 Comparison the classification feature size of LBP, LZBP, LMP, feature fusion of these descriptors, and relevant feature by ABC alone and hybrid ABC-NRS.

Here, further comparisons of the resulting relevant features by using ABC alone and in a hybrid fashion with NRS (ABC-NRS) are presented. The LZBP, LMP, and LBP descriptors combined with complementary contrast (C) features (yielding LBP\C, LZBP\C, and LMP\C), and combined all together (yielding LBP\LZBP\LMP\C). According to Fig 5.8, an improvement in the classification rate can be obtained by combining the different feature descriptors together rather than combining LBP, LZBP or LMP separately with complementary contrast features. From the figure, the average accuracy difference using ABC was 2.3682% 3.1091%, and 1.3205 % for LBP\C, LZBP\C, and LMP\C, respectively, whereas using the hybrid ABC-NRS method yielded an accuracy difference of 2.3682%, 3.0568%, and 2.2023% for LBP\C, LZBP\C, and LMP\C, respectively. Fig 5.9 shows that the feature size based on combining all different feature descriptors is somewhat larger than the feature sizes of the individual LBP\C, LZBP\C, and LMP\C methods, especially when using ABC alone as the selection method. However, utilising the hybrid ABC-NRS feature selection resulted in an improved feature size that was much closer to the feature sizes of the individual LBP\C, LZBP\C, and LMP\C methods.

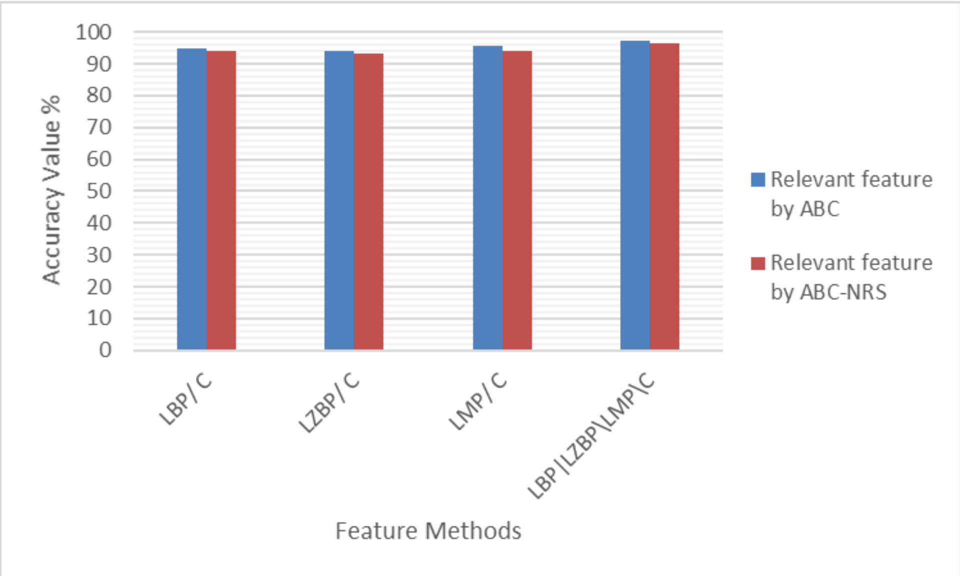


Fig. 5. 8 Comparison the classification accuracy value of LBP, LZBP, and LMP when combined with contrast feature separately, and when combining all features together.

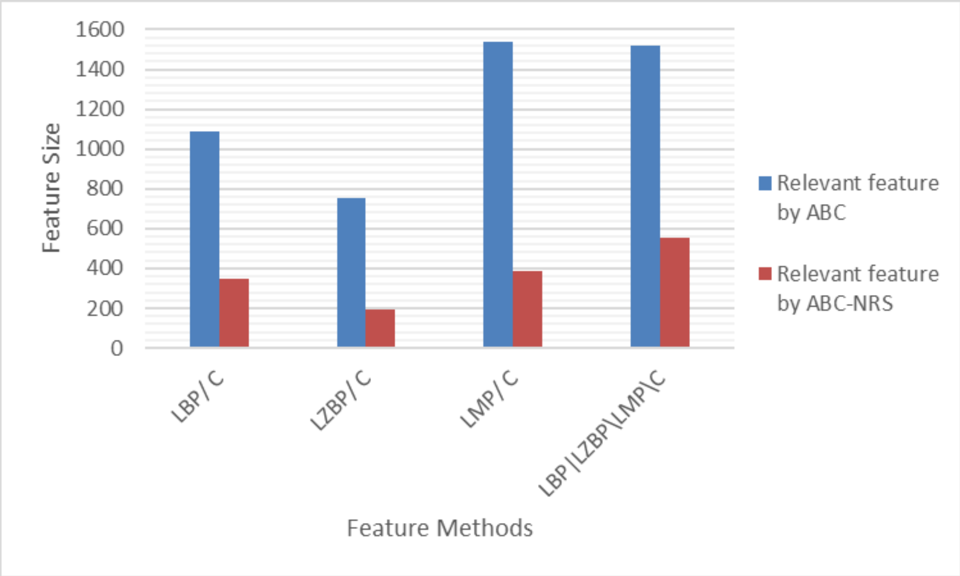


Fig. 5. 9 Comparison the classification feature size of LBP, LZBP, and LMP when combined with contrast feature separately, and when combining all features together.

5.2.2.3 Testing on Blurry and Noisy Images

Blur or noise in images occurs as a result of acquisition. The blur problem results from the motion of the scene or from an incorrect focus of the source, whereas noise in the images results from a random noise source. Referring to Subsection 2.3.1 and (R. S. Blum & Liu, 2005), integration of features from different descriptors produces highly discriminative and effective features from images affected by these problems.

A new approach for improving the features extracted from blurry and noisy images is proposed. LZBP and LMP are similar to LBP in that they depend on TUs. However, they are different in the strategy that adopt to extract features. To evaluate the noise tolerance properties of the proposed method, salt-and-pepper noise and image blurring was added to Outex_TC11n dataset, which has 24 texture classes. Fig 5.10 shows these effect on the images.

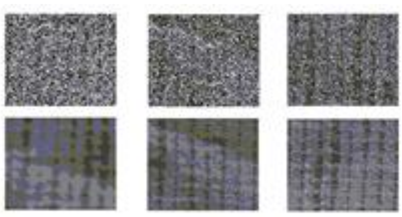


Fig. 5. 10 Based on the Outex-11dataset, the upper row texture images have had noise applied to them, with noise command values of 0.4, 0.3, 0.15 (from right to left), whereas the bottom row texture images have had blurring applied to them, with values of 1, 0.75, 0.5 (from right to left).

Results of the Blur Problem

Most classification methods are negatively affected by blur problems. Fig 5.11 shows a comparison of the classification accuracy of LBP, LZBP, and LMP separately and when combined together to target blurry images. LBP was affected by increasing the blurring values in the image from 1.0 to 5.0. LZBP, on the other hand, improved its accuracy with blurring values from 0.75 to 2.0, before dropping again with increasing blurring in the images. Of the various descriptors tested, LMP responded best to the blurring of images. Moreover, fusion between different feature descriptors did not result in a better accuracy than using LMP alone, which appears to be caused by the high dimensionality of the fused descriptors. Applying the feature selection method on the combined feature descriptors produced the best results, with only a slight degradation in accuracy for blurring values above 4.0.

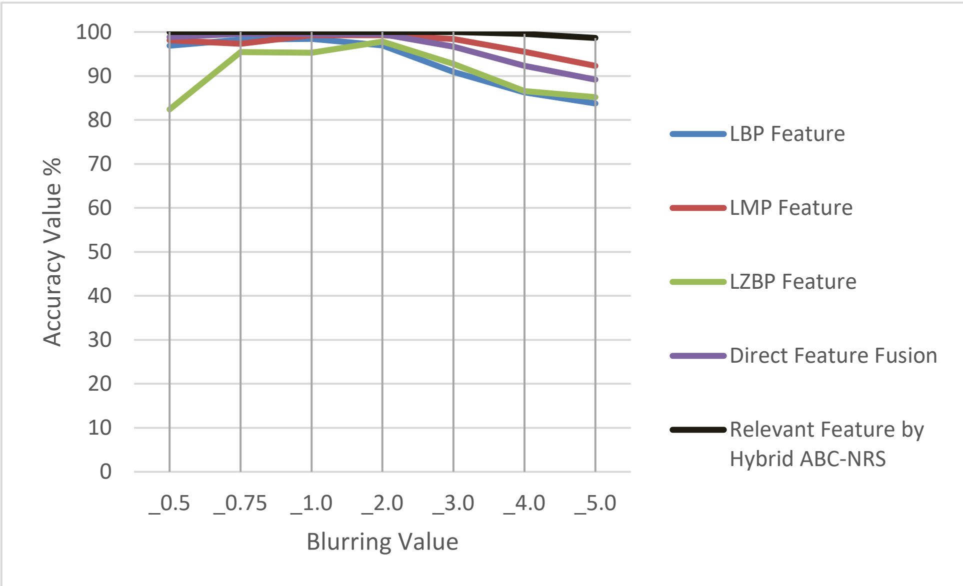


Fig. 5. 11 Comparison the classification accuracy value of involved descriptors when applied on blurry images.

Results of the Noise Problem

LTP was developed as an improvement to LBP for extracting features from noisy images (Tan & Triggs, 2010), as it provides more resistance to noise in the image. LZBP and LMP can be changed from binary patterns into ternary patterns by adding a specific value to the threshold.

Our aim is combining these local ternary feature descriptors to test whether doing so provides an improvement against the noise problem. Enhancing the robustness of feature descriptors against noise is done by combining the different descriptors to achieve a diverse set of features. More discriminative features can be obtained from fusing the novel local ternary descriptors with LTP to tackle the noise problem. Furthermore, the proposed hybrid selection method can be used for combining the involved descriptors in order to select only the relevant features.

Figure 5.12 shows that the LTP descriptor did not produce good results compared to the new LZBP and LMP descriptors. LMP produced more resistance to noise, and its effect from noise was small. The accuracy of LZBP was lower than that of LTP, however, LZBP was more stable against noise after value 0.3, where it outperformed LTP.

From the figure, feature fusion of the combined descriptors provided an improvement compared when the feature descriptors were applied separately, except for a noise value of 0.5, where LMP performed better. In addition, the figure shows that after applying our feature selection approach on the combined descriptors, a significant improvement in the accuracy was obtained across all noise values from 0 to 0.5. It is therefore clear that combining different feature descriptors and removing irrelevant features by the selection method provides more resistance to noise.

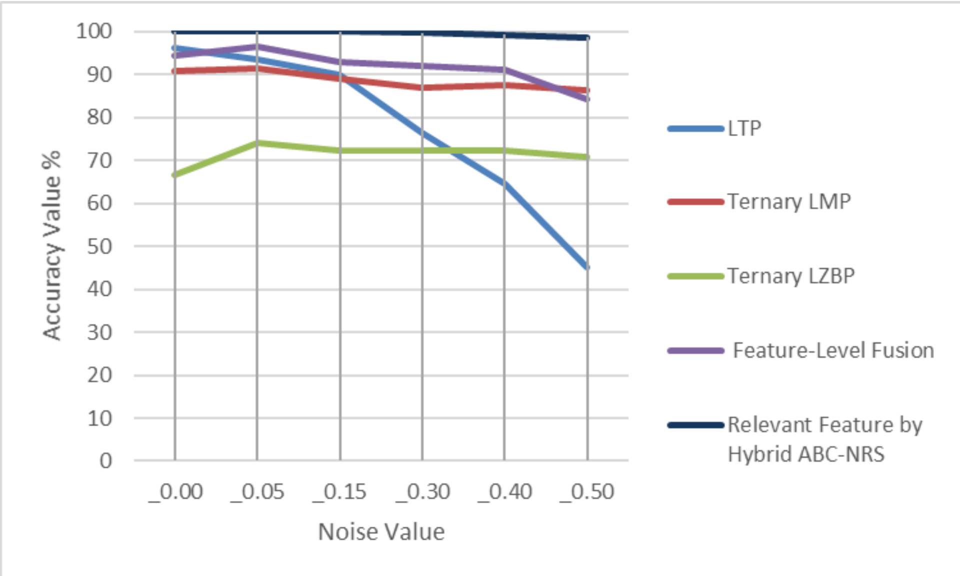


Fig. 5. 12 Comparison the classification accuracy value of involved ternary descriptors when applied on noisy images.

5.2.3 LZBP and LMP Combined with GF

This subsection follows the same procedure followed by the previous one, where tests of the performance of the LZBP and LMP descriptors combined with complementary features extracted by GF are conducted.

5.2.3.1 Tests Results

Results of Features-level Fusion

The test results shown in the tables below correspond to the classification rates of LBP, LZBP, and LMP and GF when they are applied individually, and when the former three descriptors are combined directly with GF using the BPNN (Table 5.10) and SVM (Table 5.11) classifiers. In the experiments, the methods were tested on four datasets, which are UIUC, UMD, KTHTIP2b and Brodaze.

According to the Table 5.10, the performance of GF when combined with any local feature descriptors (LBP, LZBP or LMP) outperforms any of them separately. Accuracy improvements of 5.419 %, 6.975 %, and 3.494 % where obtained from combing GF with LBP, LZBP, and LMP, respectively, using the BPNN classifier. Table 5.11 depicts similar levels of improvement when the SVM classifier was utilised, where accuracy improvements of 4.563 % 6.419 %, and 2.319 % were obtained after combining GF with LBP, LZBP, and LMP, respectively.

Table. 5. 10 Classification accuracy results using local LBP, LZBP, and LMP descriptor, GF, and feature fusion of local descriptors with GF using the BPNN classifier.

Datasets	LBP	LZBP	LMP	GF	LBP-GF	LZBP-GF	LMP-GF
UIUC	75.8	86.7	82.8	71.6	83.6	88.1	85.8
UMD	93.7	92.9	94.2	85.1	96	96.4	96.5
KTHTIP2b	86.2	73.3	85.8	79.5	90.5	87.8	90.9
Brodaze	82.975	82.625	85.725	90.3	90.25	91.125	89.3
Average	84.66875	83.88125	87.13125	81.625	90.0875	90.85625	90.625

Table. 5. 11 Classification accuracy results using local LBP, LZBP, and LMP descriptor, GF, and feature fusion of local descriptors with GF using the SVM classifier.

Datasets	LBP	LZBP	LMP	GF	LBP-GF	LZBP-GF	LMP-GF
UIUC	81.1	90.1	85.8	77.4	85.2	90.9	87.9
UMD	94.9	94.6	96.2	91	96.8	95.8	96.8
KTHTIP2b	87.9	74.7	91.2	87.5	93.55	91	93.65
Brodaze	85.175	84.875	87.875	91.65	91.775	92.25	92
Average	87.26875	86.06875	90.26875	86.8875	91.83125	92.4875	92.5875

Another set of experimental results is shown in Table 5.12 and Table 5.13. The results correspond to combining GF with the same features descriptors (i.e. LBP, LZBP, and LMP) in a multiscale analysis, where R= {1,2,3} and P=8. In this scenario, the increase in accuracy for LBP, LZBP, and LMP when combined with GF was 1.729 %, 4.171 %, and 1.425 %, respectively when the BPNN classifier was used; whereas the improvement was 2.136 %, 3.051 %, and 0.944 %, respectively when the SVM classifier was used.

From these results, although there is an improvement from combining GF with other descriptors in multiscale over combining in a single scale, the improvement is not significant. In comparison, the difference in classification rate between combining a set of Gabor filters with LBP, LZBP, and LMP using single scales (1,8) and combining a set of Gabor filters with multiscale LBP, LZBP, and LMP (with R= {1,2,3} and P=8) was 1.077 %, 2.131 %, and 1.828 %, respectively using the BPNN classifier, and 1.069 %, 1.505 %, and 1.138 %, respectively using the SVM classifier.

Table. 5. 12 Classification accuracy results using multiscale local LBP, LZBP, LMP descriptors, GF, and feature fusion of local descriptors with GF, using the BPNN classifier.

Datasets	LBP	LZBP	LMP	GF	LBP-GF	LZBP-GF	LMP-GF
UIUC	81.8	92.2	87.6	70.7	83.9	92.8	88.4
UMD	96.5	96.2	96.6	87.3	98.7	98.3	97.2
KTHTIP2b	90.72	78.54	89.36	81.09	91.36	89	92.81
Brodaze	88.725	88.325	90.55	90.275	90.7	91.85	91.4
Average	89.43625	88.81625	91.0275	82.34125	91.165	92.9875	92.4525

Table. 5. 13 Classification accuracy results using multiscale local LBP, LZBP, LMP descriptors, GF, and feature fusion of local descriptors with GF, using the SVM classifier.

Datasets	LBP	LZBP	LMP	GF	LBP-GF	LZBP-GF	LMP-GF
UIUC	83.3	93.8	89.5	77.4	86.5	94.9	90.6
UMD	96.3	96.9	97	91.9	96.5	96.6	97.1
KTHTIP2b	92.63	82.54	93	87.72	95.45	91.27	94
Brodaze	90.825	90.525	91.625	91.525	93.15	93.2	93.2
Average	90.76375	90.94125	92.78125	87.13625	92.9	93.9925	93.725

Although there was improvement when combining LBP, LZBP, and LMP with different scales and set of filters in a Gabor bank, the classification performance can be further improved by selecting only the relevant features from the joined features, instead of depending on the complete set of combined features. The feature size of multiscale LBP, LZBP, LMP is given by concatenating the features of different shared scales, which increased the feature size in these tests by three times for R= {1,2,3} and P=8, whereas the feature size of the Gabor bank consists of a group of filters. This may have an effect on the classification accuracy as well as computation time as result of the huge feature size stemming from using different filters and LBP, LZBP, and LMP scales.

Results of Feature Selection

Next, an attempt to improve the classification performance was made by depending on diversity of information and avoiding the high dimensionality problem arising from feature-level fusion in the previous subsection. The proposed feature selection approaches were applied for multiscale LBP, LZBP, and LMP with complementary features extracted by GF. In the feature selection stage, the performance of classification is expected to be improved by depending on only the relevant features from the previous feature-level fusion, as explained in Section 3.3.2.

In the first test, ABC was utilised for selecting the relevant features from combined set of features. ABC had the responsibility to select the optimum features, which involved selecting the optimum filters from Gabor bank and the relevant feature scales from the multiscale LBP, LZBP, and LMP descriptors.

Table 5.14 shows for results corresponding to the BPNN classifier, whereas Table 5.15 shows those corresponding to the SVM classifier, where the classifiers were used to evaluate the selected parts of features for the ABC wrapper method. From the results, combining the relevant features from different feature descriptors with the optimum GF resulted tiny very similar, which show improvement of GF on the classification accuracy of these descriptors. However, on average, combining LMP with GF achieved the best performance for most databases, recording a classification accuracy of 95.3675% using BPNN, and 95.8425% using SVM. It was also noticed that, on average, LZBP recorded a better classification rate of 95.2375% than LBP with 94.727%. Using the BPNN classifier, the largest accuracy difference was obtained using the UIUC database, where combining LZBP and GF outperformed combining LBP or LMP with GF by a difference of 7.8% and 3.6%, respectively. For the KTHTIP2b database, higher accuracies were obtained from combining LBP or LMP with GF compared with combining LZBP with GF, where the difference was 5.36% and 3.82%, respectively. For the UMD and Brodaze datasets, the differences in accuracy between different feature descriptors combination were very minor.

Table. 5. 14 Classification accuracy and feature dimensionality results using feature fusion and feature selection from local feature descriptors and GF using ABC and the BPNN classifier.

Methods	Datasets	Relevant features from fusion		Relevant features from local features		Relevant features from GF	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**	**Accuracy**	**F-Dim**
LBP-GF	UIUC	86.5	572	79.1	512	49.1	60
	UMD	98.3	816	96.9	768	78.1	48
	KTHTIP2b	94.81	820	88.63	768	77.45	52
	Brodaze	99.3	542	97.4	512	93.3	30
	Average	94.727	687.5	90.507	640	74.487	47.5
LZBP-GF	UIUC	94.3	554	92.6	546	27.5	8
	UMD	98.1	554	96.3	546	74.8	8
	KTHTIP2b	89.45	434	76.90	364	81.09	70
	Brodaze	99.1	214	98.4	182	94.4	32
	Average	95.2375	439	91.05	409.5	69.4475	29.5
LMP-GF	UIUC	90.7	544	85.2	512	52.2	32
	UMD	98.2	1568	97.9	1536	80.5	32
	KTHTIP2b	93.27	1552	91.18	1536	54.18	16
	Brodaze	99.3	1042	98.3	1024	89.7	18
	Average	95.3675	1176	93.145	1152	69.145	24.5

Table. 5. 15 Classification accuracy and feature dimensionality results using feature fusion and relevant feature from local feature descriptors and GF using ABC and the SVM classifier.

Methods	Datasets	Relevant features from fusion		Relevant features from local features		Relevant features from GF	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**	**Accuracy**	**F-Dim**
LBP-GF	UIUC	88.2	588	84	512	74.4	76
	UMD	98.29	826	97.5	768	81.7	58
	KTHTIP2b	95.45	816	92.45	768	83.18	48
	Brodaze	99.1	310	97.4	256	96.2	54
	Average	95.26	635	92.837	576	83.87	59
LZBP-GF	UIUC	95.2	548	94.1	546	19.3	2
	UMD	98	556	97.3	546	58.8	10
	KTHTIP2b	90.63	616	83	546	85.09	70
	Brodaze	99.2	260	97.9	182	97	78
	Average	95.7575	495	93.075	455	65.0475	40
LMP-GF	UIUC	91.9	1544	91	1536	41.4	8
	UMD	98	1584	97.6	1536	88	48
	KTHTIP2b	94.27	1588	92	1536	83	52
	Brodaze	99.2	1088	98.5	1024	96.7	64
	Average	95.8425	1451	94.775	1408	77.275	43

The second experiment served to test whether the feature length would be reduced by applying the hybrid ABC-NRS method, where ABC selects the optimum filters, and multiscale LBP, LZBP, and LZBP have their features reduced by NRS before combining with the optimum filters.

Table 5.16 and Table 5.17 report the results obtained from applying the hybrid ABC-NRS selection method, where the former shows the results of using the BPNN classifier for wrapper method, whereas the latter shows those corresponding to using the SVM classifier. Here, the classification accuracy results of different combinations of LBP, LZBP, and LMP with GF was very similar as when depending on ABC only for feature selection, whereas the feature length compared to depending on entire feature was significantly reduced by the new hybrid selection method. The number of features selected by ABC or the hybrid ABC-NRS method was less than the total number of features, where the hybrid ABC-NRS method offered a much lower total number of features. Furthermore, the average accuracy values using BPNN and selected features by ABC-NRS was 94.2575%, 95.1275%, 94.565 % for LBP, LZBP, and LMP, respectively, whereas using the hybrid ABC-NRS method and the SVM classifier resulted in average accuracy values of 95.32249%, 96.7575%, and 95.6725% for LBP, LZBP, and LMP, respectively. A detailed discussion of accuracy and feature length obtained from this experiment is provided in the next subsection.

Table. 5. 16 Classification accuracy and feature dimensionality results using feature fusion and relevant features selection from local feature descriptors and GF using ABC-NRS and the BPNN classifier.

Methods	Datasets	Relevant features from fusion		Relevant features from Local Features		Relevant features from GF	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**	**Accuracy**	**F-Dim**
LBP-GF	UIUC	85.8	293	79	253	61	40
	UMD	98.2	249	95.7	199	83.5	50
	KTHTIP2b	93.63	422	89.54	358	77.36	64
	Brodaze	99.4	225	97.9	167	95.8	58
	Average	94.2575	297.25	90.535	244.25	79.415	53
LZBP-GF	UIUC	92.8	72	89.4	70	37	2
	UMD	97.8	133	97.4	87	82.3	46
	KTHTIP2b	90.81	168	80.45	104	79.54	64
	Brodaze	99.1	62	97.4	32	93.7	30
	Average	95.1275	108.7	91.1625	73.25	73.135	35.5
LMP-GF	UIUC	89	374	83.7	298	62	76
	UMD	97.7	272	96.5	222	82.4	50
	KTHTIP2b	92.36	431	87.54	385	76.45	46
	Brodaze	99.2	196	97.9	168	92.2	28
	Average	94.565	318.2	91.41	268.2	78.2625	50

Table. 5. 17 Classification accuracy and feature dimensionality results using feature fusion and relevant features selection from local feature descriptors and GF using ABC-NRS and the SVM classifier.

Methods	Datasets	Relevant features from fusion		Relevant features from Local features		Relevant features from GF	
		Accuracy	**F-Dim**	**Accuracy**	**F-Dim**	**Accuracy**	**F-Dim**
LBP-GF	UIUC	88.5	407	85.9	341	74.6	66
	UMD	98.19994	235	95.2	199	81.6	36
	KTHTIP2b	95.09	252	91.18	222	78.81	30
	Brodaze	99.5	81	96.9	51	94.5	30
	Average	95.32249	243.7	92.295	203.2	82.3775	40.5
LZBP-GF	UIUC	95.5	123	94.7	99	61.3	24
	UMD	98.5	103	98	87	79.2	16
	KTHTIP2b	93.63	160	83.72	104	84.27	56
	Brodaze	99.4	143	98.5	77	96.3	66
	Average	96.7575	132.2	93.73	91.75	80.2675	40.5
LMP-GF	UIUC	91.2	316	90.8	298	61.4	18
	UMD	98.2	162	96.5	138	84	24
	KTHTIP2b	94.09	439	91.18	385	83.36	54
	Brodaze	99.2	216	98.2	168	95.9	48
	Average	95.6725	283.2	94.17	247.2	81.165	36

5.2.3.2 Discussion and Comparison of Results

In this research, features extracted by GF were used as complementary features for the proposed LZBP and LMP features descriptors. Features from GF and LBP have been as complementarily combined before, where the former extracts global features, and the latter extracts local features (Tan & Triggs, 2007). Furthermore, feature fusion has been tested to provide a diversity of features to improve the classification performance, and to overcome the dependence on a limited number of filters (M. Li & Staunton, 2008).

GF have been used in many studies for texture classification and have reported high performance. In this research, GF were tested and evaluated against other common feature extraction methods (see Subsection 5.2.1). In our work, the proposed improved LZBP and LMP descriptors based on

LBP were tested with complementary GF features. There is no agreement in the literature on the number of filters needed for texture classification, where different resources report a number of frequencies ranging from three to seven, and a number of orientations ranging from four to eight (Bianconi & Fernández, 2007). In our tests, 40 filters from five frequencies and eight orientations were used, and the features by each filter from the image were based on calculating mean and standard deviation (Haghighat, Zonouz, & Abdel-Mottaleb, 2015).

The tests were conducted on a number of different datasets. Fig 5.13 and 5.14 show a comparison of the average classification rate and feature size of different methods, respectively. The integration of features between GF and different local feature descriptors follows a number of cases. The direct combination of shared LBP, LZBP, and LMP features included single and multi-scales to improve feature extraction (Ojala, Pietikainen, et al., 2002). The other cases are for improved feature selection methods on the proposed combination of features. These cases can be explained in the context of testing as follows:

Combining the feature vector of un-optimized GF directly with the LBP, LZBP, and LMP descriptors, where these local feature descriptors were applied with a single features scale (1,8).

Combining the feature vector of un-optimized GF directly with the LBP, LZBP, and LMP descriptors, where these local feature descriptors were applied with three features scales, with R= {1,2,3} and P=8.

Extracting only the relevant features from the multiscale local descriptors and Gabor filter banks using the ABC method.

Extracting only the relevant features from the multiscale local descriptors and Gabor filter banks using the hybrid ABC-NRS method.

From the results using BPNN or SVM classifiers, an improvement in the performance of image classification is noted from fusing GF with other local feature descriptors (LBP, LZBP, and LMP). Furthermore, it can be observed that the performance of any of the feature level fusion methods is superior to that of using the feature descriptors alone. Combining GF with LZBP resulted in the highest accuracy, followed by combining GF with LMP. However, a greater performance improvement was achieved by depending only on the relevant features in the different states of combination.

The results by BPNN or SVM classifiers were similar in a number of different states of comparison, therefore the following discussion will be based on the results of the BPNN classifier.

Utilising only relevant features using the proposed feature selection methods, Fig 5.13 shows that higher classification accuracies were obtained from selecting the relevant feature from the fused vector of GF with LBP, LZBP, or LMP than applying direct feature fusion. Furthermore, Fig 5.14 demonstrates that compared to direct feature fusion, the hybrid feature selection method reduced the feature size significantly by extracting only the relevant features for both GF and LBP, LZBP, and LMP. The feature size was the smallest among other states of comparison, and smaller than direct feature fusion of single scale LBP, LZBP, and LMP descriptors.

In the feature selection stage, the relevant features of GF were selected automatically by the ABC method. In addition, ABC has the ability to select more relevant features from local feature descriptors, where it is usually applied with multiscale applications, which have high dimensionality. However, the hybrid ABC-NRS method can offer further feature length reduction for multi-scale LBP, LZBP, and LMP to match the original feature length.

The selected features by the hybrid ABC-NRS method from the original set of features improved the classification performance over depending on the complete set of features. Although combining GF with LBP or LMP resulted in slightly less classification accuracies using the hybrid ABC-NRS method than using ABC alone, the results of the hybrid ABC-NRS were still superior to direct feature fusion. Taking into consideration accuracy vs. feature length, it is therefore clear that using the hybrid ABC-NRS method with combined features provides excellent performance results for textures classification.

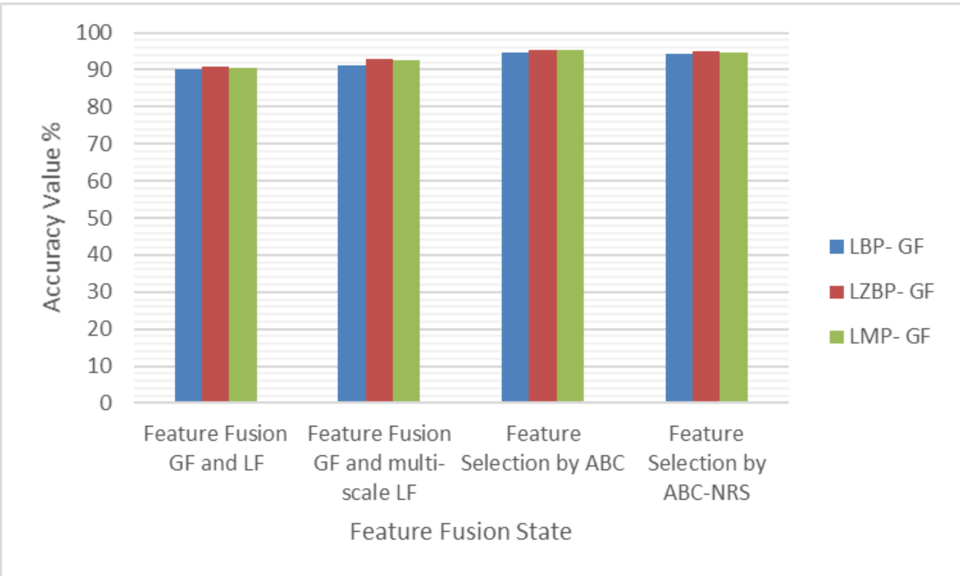


Fig. 5. 13 Comparison the classification accuracy value of feature fusion between local features descriptors (LBP, LZBP, and LMP) and GF using the BPNN classifier.

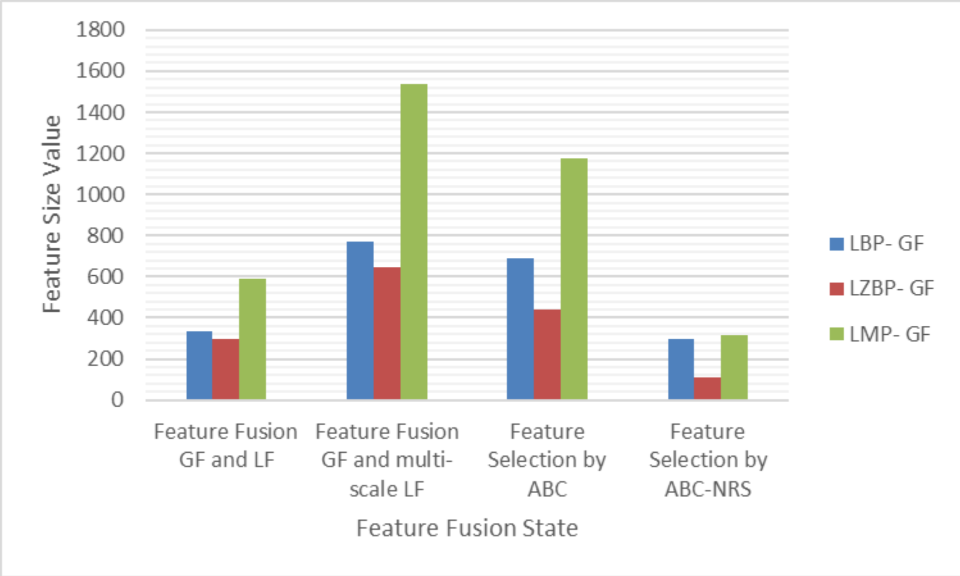


Fig. 5. 14 Comparison the classification feature size of feature fusion between local features descriptors (LBP, LZBP, and LMP) and GF using the BPNN classifier.

5.3 Summary

The difficulty of producing satisfactory discriminative feature extraction tools for complicated texture characteristics is the main shortcoming in classification systems. Research has shown that the improving feature extraction, which will reflect on the overall classification system, can be achieved by improving the existing texture descriptors and feature selection methods. In this work, the proposed texture descriptors and feature-level fusion were intended to achieve more effective features, whereas the improved hybrid feature selection method was utilised for the proposed feature-level fusion models, which is more effective approach than depending on direct fusion only.

An experimental testing has been carried out to investigate the proposed novel texture descriptors and improved hybrid selection methods. Although some limitations in the proposed methods were identified, the classification system demonstrated an outstanding performance.

The experimental testing of the proposed feature extraction methods started by assessing the new LZBP and LMP texture features descriptors. The testing included commonly used statistical and signal processing methods for textures classification using two different classifiers: BPNN and

SVM. The performance of texture features by LZBP and LMP compared to existing methods, where the new feature was found to be effective with improving in classification accuracy across different texture characteristics of the involved datasets tested.

The experimental testing also included combining the proposed LZBP and LMP descriptors with complementary features based on contrast measure and GF. The results showed that feature fusion seems to be robust against different texture characteristics. However, although there is an improvement was seen when combining multiscale LZBP and LMP with the contrast of texture image or GF, a substantial further improvement in the classification performance can be achieved by selecting only relevant features instead of depending on the complete set of combined features. In this context, to obtain optimum features from the combined feature set, the experiments investigated two feature selection approaches. The main aim was to achieve an improvement in classification accuracy while maintaining the smaller feature space by depending on a diverse set of features. In contrast to the argument that wrapper methods like ABC are more expensive than filter methods, we found that depending on the ABC algorithm reduced the feature length and improving the classification accuracy. However, combining the NRS method with the wrapper feature selection method based on the ABC algorithm resulted in a further significant reduction in feature space. While the hybrid ABC-NRS selection method was more effective in feature length reduction, a small reduction in classification performance was noticed in comparison to using the wrapper ABC method alone.

Chapter 6

Evaluation

The previous chapter discussed the results of testing the new proposed strategies for improving texture feature extraction in image classification applications. The focus of this chapter is to evaluate the performance of the methods proposed for improving texture feature. By implementing the pre-processing and post-processing stages of the proposed improved feature extraction methods (as explained in the methodology chapter), one can expect to obtain an improved classification performance. In the evaluation, an investigation is conducted on whether an improvement in the overall classification system can be attained owing to success of the proposed feature extraction approaches.

This chapter starts by evaluating the performance of the new LZBP and LMP feature descriptors in Section 6.1. In Section 6.2, the effect of combining these new feature descriptors with complementary image contrast features using the new hybrid ABC-NRS feature selection method is investigated. Section 6.3 adopts the same procedure of evaluation with GF serving as a complementary feature extraction method. Finally, Section 6.4 concludes the chapter by providing a summary of its main findings.

6.1 LZBP and LMP Descriptors

Various means were adopted in an attempt to evaluate the performance of proposed LZBP and LMP descriptors. The evaluation was based on comparing the results with other related methods used in texture classification applications. In addition, a number of public datasets were utilised to investigate the performance of the distinctive features extracted from the involved descriptors. For the texture classes of each dataset, the confusion matrix resulting from the classification was investigated.

6.1.1 Performance Comparison with Competing Methods

In general, the new methods can only be claimed successful if they produce results that can be validated to be superior to those obtained by existing methods. The performance of the proposed LZBP and LMP descriptors was compared to LBP, because these new LZBP and LMP texture descriptors were developed based on LBP, as explained in Section 3.3.1. We also involved in the comparison the most well-known traditional texture descriptors, which proved their ability in texture feature extraction for image classification (see Section 2.2). This comparison was also used to verify and prove that the choice of LBP as the most important texture feature extraction method, and its use as a basis to develop descriptors with improved discriminative feature extraction. The other involved descriptors used in the comparison were based on the co-occurrence matrix, such as GLCM, which is one of earliest methods used for extracting features from texture and is still used by different image classification applications (Chaki, Parekh, & Bhattacharya, 2015), where the features extracted describe the homogeneity, contrast, and the presence of organised structures within the image (Haralick & Shanmugam, 1973). GLDM was also used in the comparison, which is an improved form of GLCM, and used frequently in different texture extraction applications (J. K. Kim & Park, 1999), where the extracted features describe the contrast, angular second moment, entropy, mean, and inverse difference moment of the image (Weszka et al., 1976). In addition, the comparison included TS (He & Wang, 1990), which is an improved approach to co-occurrence methods that was developed to achieve better texture feature extraction from TUs. However, instead of using 6561 TUs, TS was enhanced by reducing the feature length to 15 units without significant loss in discriminating power (He & Wang, 2010). Finally, among the signal processing methods, GF and WT were included in the comparison, as they represent the most commonly used signal processing methods. GF with bi-dimensional Gaussian function of five scales and eight orientation were used, producing 40 filters, where from each filter, the mean and standard deviation were utilised for feature representation (Manjunath & Ma, 1996). For the WT method, the features were calculated by mean and standard deviation of approximation of sub-bands of three-level decomposed images by DWT (Zaid, Jadhav, & Deore, 2013).

The results of evaluating the aforementioned descriptors are shown in Fig 6.1 and Fig 6.2, which illustrate the average results of images classification on number of datasets using the ANN and SVM classifiers, with 5-fold cross-validation and 10-fold cross-validation, respectively. These results are based on Table 5.2 and Table 5.3, respectively. Considering the average accuracy of each method achieved across all the databases tested, it is clear that the co-occurrence matrix methods, namely the GLCM and GLDM methods, produced the lowest accuracies, which prove their weakness compared to methods based on TUs quantisation. However, TS also displayed poor quantisation of texture patterns, as its accuracy was much lower than that of LBP. For signal processing methods, although WT's results were acceptable compared to the aforementioned methods, GF outperformed WT. The main problem with GF is their high computation time. From the figures, excluding the proposed methods, the competition for the highest accuracy was between LBP as a statistical method and GF as a signal processing method. LZBP and LMP demonstrated their capability in competing with LBP and producing better classification accuracies compared to their counterparts, which was the main aim of this stage of the evaluation. In further evaluation, we concentrate on comparing between the improved LZBP and LMP descriptors and against LBP.

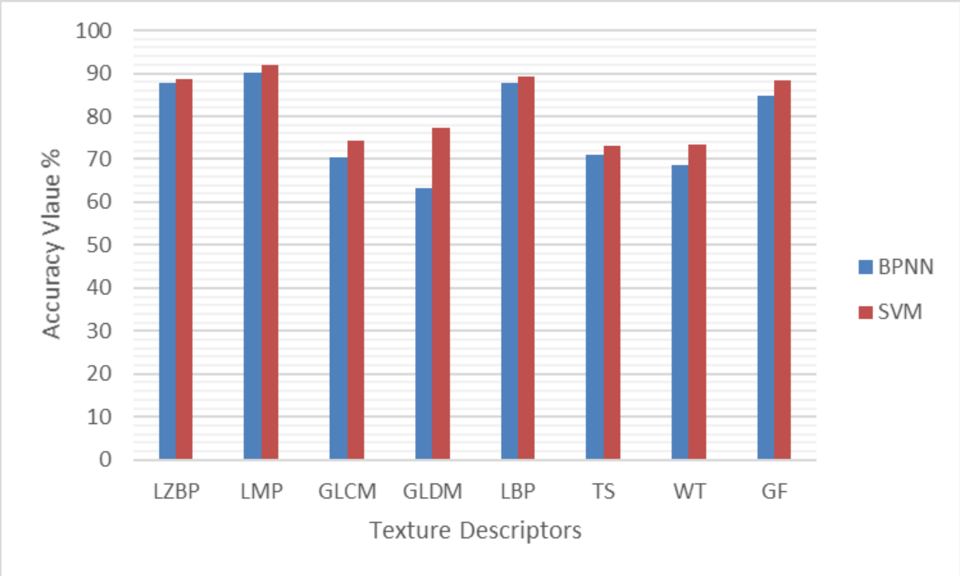


Fig. 6. 1 Comparison classification accuracy value between LZBP and LMP and other common texture descriptors using BPNN and SVM with 5-fold cross-validation.

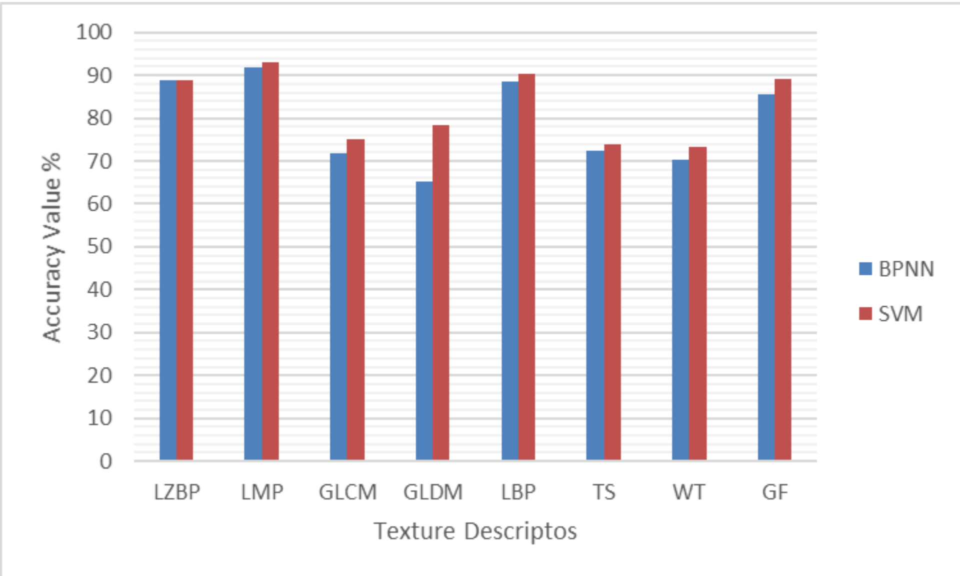


Fig. 6. 2 Comparison classification accuracy value between LZBP and LMP and other common texture descriptors using BPNN and SVM with 10-fold cross-validation.

6.1.2 Comparison of Performance on Different Datasets

This research utilised public datasets for texture classification. These datasets were used to investigate the performance of distinctive feature extraction by the proposed LZBP and LMP methods. The datasets involve different types of texture characteristics, which makes to make texture classification a challenging process. Furthermore, each dataset divides the images into a number of classes.

In the experiments, four public datasets were used, and their texture images are displayed in Fig 6.3, Fig 6.4, Fig 6.5, and Fig 6.6. The first set of images correspond to samples from the UIUC dataset (Lazebnik et al., 2005), which is one of the most modern datasets, designed with significant changes in strong viewpoints and different scales, with uncontrolled illumination conditions. The UMD texture images are also important (Xu et al., 2009), where the dataset shares the same number of classes as UIUC, with different rotations and scales. The Brodatz texture images are widely in texture classification. However, this dataset is one of oldest datasets, as it is non-rotational and scale invariant (Laine & Fan, 1993). The KTHTIPS2b dataset is also a relatively new dataset, consisting of samples with three viewing angles, four illumination sources, and nine different scales (Mallikarjuna et al., 2006). For other parameters of these datasets, refer to Table 5-1.

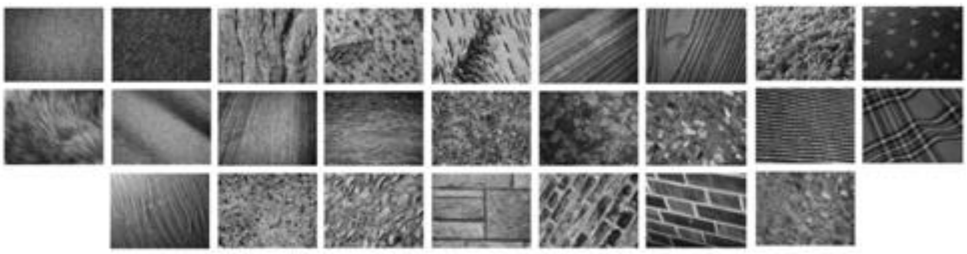


Fig. 6. 3 Samples of texture images from the UIUC dataset.

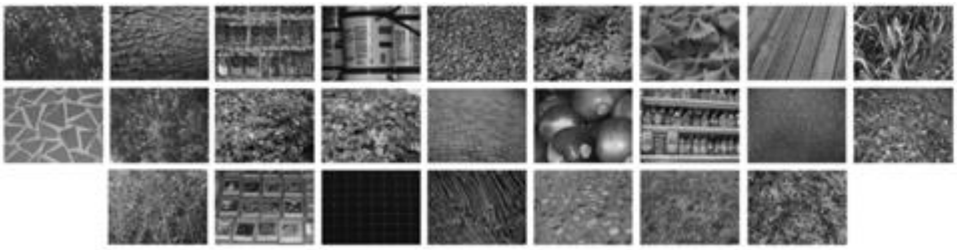


Fig. 6. 4 Samples of texture images from the UMD dataset.

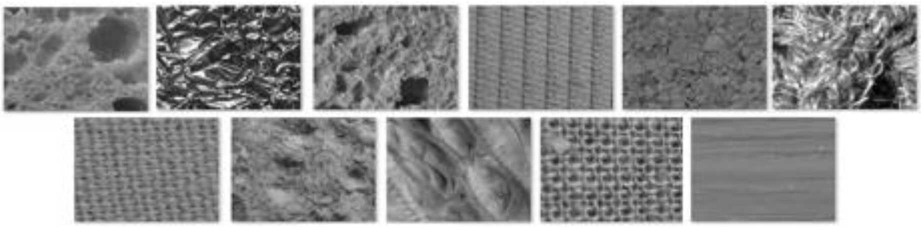


Fig. 6. 5 Samples of texture images from the KTHTIPS2b dataset.

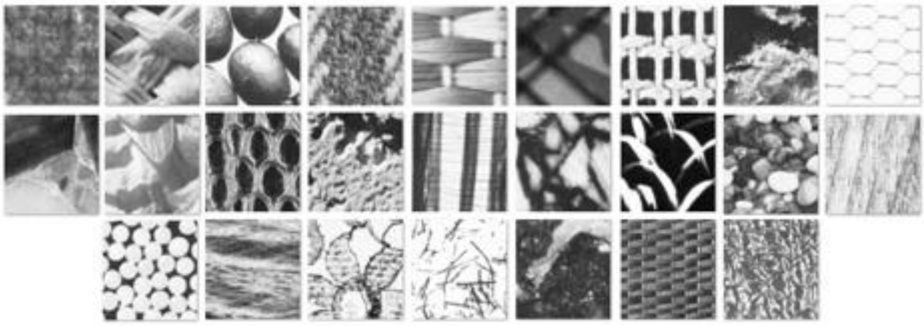


Fig. 6. 6 Samples of texture images from the Brodatz dataset.

Despite the good performance of LBP compared with other statistical methods, LZBP and LMP are proposed to process and cope with some limitations in LBP. The LZBP and LMP descriptors are designed to extract more distinctive features. Addressing the LBP limitations should result in a better performance of LZBP and LMP on different texture characteristics compared to LBP. This section thus represents a comparison of the performance of the proposed LZBP and LMP descriptors with that of the LBP descriptor, using the aforementioned datasets as benchmarks. These descriptors were applied in a multiscale analysis, with R= {1, 2, 3} and P =8. The performance of the descriptors using the involved datasets was compared in terms of the average classification accuracy on multi-scale state, as the mean or average classification accuracy is a widely accepted measure to evaluate a classifier’s performance.

According to the experimental results of Fig 6.7, the highest accuracy recorded for the UIUC database corresponded to the LZBP descriptor, which outperformed both the LBP and LMP descriptors. For the UMD dataset, the LZBP also somewhat outperformed both LMP and LBP, whereas the LZBP descriptor recorded the lowermost accuracy levels compared to other descriptors for the KTHTIP2b database. For all of the three descriptors compared, the accuracy levels were obtained using the Brodatz database, where LMP provided a slightly higher average classification accuracy than the other two descriptors. These differences in the results across different databases lead us to a more effective feature method that can deal with different texture characteristics, which can be achieved by integrating between different feature descriptors, while taking into consideration the issue of the resulting huge feature space.

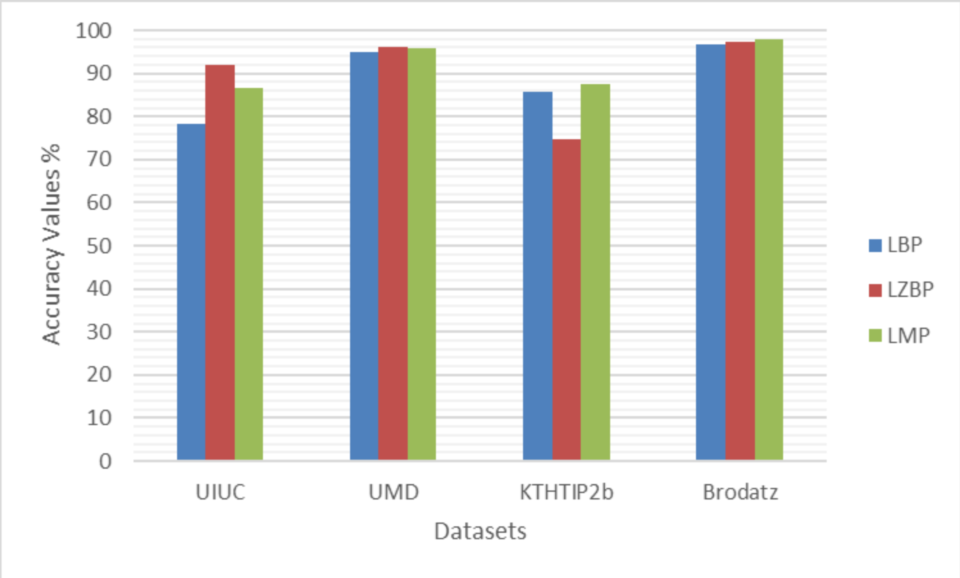


Fig. 6. 7 Comparison the average of classification accuracy value of multiscale LBP, LZBP, and LMP descriptors based on different datasets.

6.1.3 Comparison of Performance Using the Confusion Matrix

The results of texture classification are commonly used to evaluate texture feature descriptors. The datasets in the previous section divide the images into a number of classes. For further investigation of the capabilities of the new improved feature descriptors on different classes of texture datasets, the confusion matrix is an appropriate tool. Based on the outcome of the BPNN classifier using the confusion matrix, the LZBP, LMP, and LBP descriptors were evaluated and compared on texture datasets classes. This serves to highlight the most suitable texture descriptor for each texture surface, and which classes of texture images of datasets are the most challenging to classify by these descriptors.

Appendix 1 contains the results of the confusion matrix evaluation for the target descriptors in this comparison study. The results of confusion matrix evaluation were divided into four groups (A, B, C and D) for the involved datasets.

Tables A.1 to A.3 show the confusion matrix results of the LBP, LZBP and LMP descriptors, respectively, when applied on the UIUC dataset. From Table A.2, it is evident that LZBP is more appropriate for dealing with the UIUC database compared to other descriptors. LZBP provided 100% accuracy (40 out of 40) for four classes (7, 13, 17, 18), whereas its lowest number of correctly classified images was observed for Class 9, where a classification accuracy of 72.5 % (29 out of 40) was obtained. For LBP, the highest accuracy achieved was in classes 16 and 17, with 39 correct correctly classified images (97.5 %), whereas the lowest accuracy levels were recorded in four classes (3,4,6,25), with only 22 correctly classified images out of 40 (or 55%).

Tables B.1 to B.3 show the confusion matrix results corresponding to the UMD dataset. LBP showed a 100% accuracy in only Class 15 (40 out of 40), whereas LZBP achieved 100% accuracy in 6 classes (5, 7, 14, 15, 16, 24). LMP correctly classified 12 classes out of 40 with 100% accuracy, which makes it the more appropriate descriptor for most classes of this dataset.

Table C.1 to C.3 show the confusion matrix results corresponding to KTHTIPS2b dataset. In Table C.1, the highest accuracy achieved for LBP using this database was for the first class with 98%, followed by Classes 9 and 10 with 96%, whereas LZBP did not record good classification accuracies in these classes. Table C.2 shows a big drop in the performance of the LZBP for each class, where the highest accuracy obtained was only 91%, and was achieved for the first class. The second highest accuracy (90%) was achieved for Class 4, whereas the lowest accuracy was achieved for Class 5 with 57 %. Table C.3 shows that LMP also recoded good results for the first class with 96% classification accuracy, putting it 2% behind LBP and 5% ahead of LZBP in terms of Class 1 classification performance. Other classes recording good classification with LMP are Class 10 with 97%, and Classes 4 and 9 with 92%. The lowest classification accuracy achieved by LBP was recorded for Class 3 with 72%. In compassion, LZBP recorded its lowest classification accuracy for Class 3 with only 42%, where LMP recorded its lowest for Class 5 with 69%.

Table D.1 to D.3 show the confusion matrix results of the three descriptors when applied on the Brodatz dataset. In this dataset, the descriptors recorded the highest number of classes with perfect classification rates (i.e. 40 correctly classified images out of 40) compared with previous datasets. For instance, LBP obtained a 100% classification accuracy for 12 classes (2, 3, 5, 7, 8, 12, 13, 15, 18, 19, 20, 21), while LZBP achieved the same feat for 11 classes (3, 4, 5, 8, 12, 13, 15, 16, 18, 19, 22), and LMP for 15 classes (1, 2, 3, 5, 7, 8, 11, 13, 15, 16, 18, 19, 20, 21, 22). The lowest number of correctly classified images in one class among the different descriptors tested occurred for Class 25, where only 33 images out of 40 were correctly classified (82.5%). LBP recorded the lowest classification rate on class 23 with 77.5%, whereas LZBP recorded 82.5% for class 23. In comparison, for the same class (class 23), LMP had also its lowest accuracy with 72.5% (29 images out of 40).

By comparing the previous descriptors based on CM results, it can be noticed that LZBP outperformed LBP in most classes in the UIUC dataset. However, there are some classes where LBP produced the same results as LZBP, such as in Class 16. The same observation applied to the KTHTIPS2b dataset, where LBP outperformed LZBP in most classed, with the exception of Class 4, where both descriptors produced the same results.

6.2 LZBP and LMP Combined with Contrast Features

To improve the proposed LZBP and LMP descriptors, their features are suggested to be combined with the contrast of the texture image, where the same approach has been adopted before with LBP (Ojala et al., 1996). In the previous chapter, feature fusion between LZBP and LMP and contrast measure as a complementary feature was tested. In addition, new selection methods were tested on the fused features to reduce the feature space and depend only on the relevant features. In this part of the chapter, the performance of LZBP and LMP is evaluated after they are combined with contrast measure, and after the hybrid selection approach is utilised on the resulting fused features.

6.2.1 Comparison of Feature Fusion Results on Different Datasets

The results of the average classification accuracy of the different multiscale descriptors based on four texture databases are shown Table 5-4, where features contrast was used to enhance the accuracy of the descriptors. Fig 6.8 shows a comparison of the classification performance of the descriptors when applied on the UIUC dataset. The increase in classification accuracy of LBP, LZBP, and LMP as a result of being combined with the contrast features compared with using the descriptors separately was 7.343%, 0.4%, and 3.772%, respectively. Furthermore, in this dataset, the contrast features had a better effect on LBP and LMP than LZBP. However, despite the contrast features having little effect on LZBP, the accuracy of LZBP was still better than LBP\C and LMP\C. Fig 6.9 shows the performance of the descriptors when applied on the UMD dataset, where the contrast had a negative effect on LZBP, as its accuracy dropped by 0.43 %, whereas it slightly improved the performance of LBP and LMP by 1.03% and 0.557%, respectively. Fig 6.10 shows the performance of the descriptors when used with the KTHTIP2b database, where the contrast had a negative effect on the performance of LMP, as it reduced its accuracy by 1.701%, whereas it improved the accuracy of LZBP by 2.387%. It can therefore be noted that the contrast features have a better effect on descriptors with already low classification accuracies. Fig 6.11 show the results corresponding to the Brodatz database, where an increase in accuracy of 1.6%, 0. 59%, and 0.9% was obtained for LBP, LZBP, and LMP, respectively, as a result of fusion with contrast features. Although LMP\C achieved the highest accuracy, the other two descriptors came very close after being combined with the contrast of the texture images. One can therefore conclude that, although different patterns of results were recorded for the tested descriptors when combined with contrast features on different texture characteristics, in general, fusing with contrast features results in a more positive impact in terms of improving the classification accuracy, especially for feature descriptors with already low classification performance.

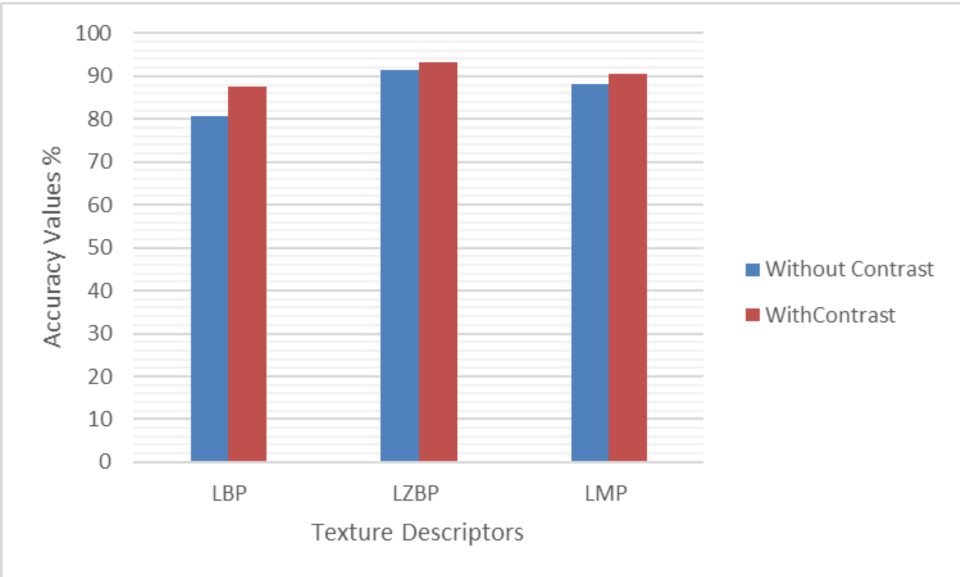


Fig. 6. 8 Comparison classification accuracy value of the LBP, LZBP, and LMP when combined with the contrast of the texture image based on the UIUC database.

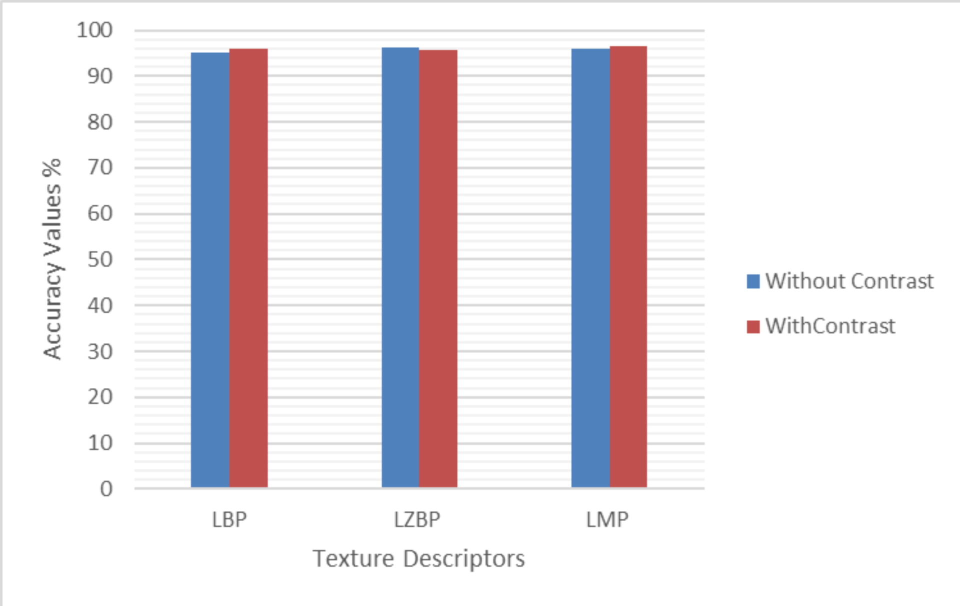


Fig. 6. 9 Comparison classification accuracy value of the LBP, LZBP, and LMP when combined with the contrast of the texture image based on the UMD database.

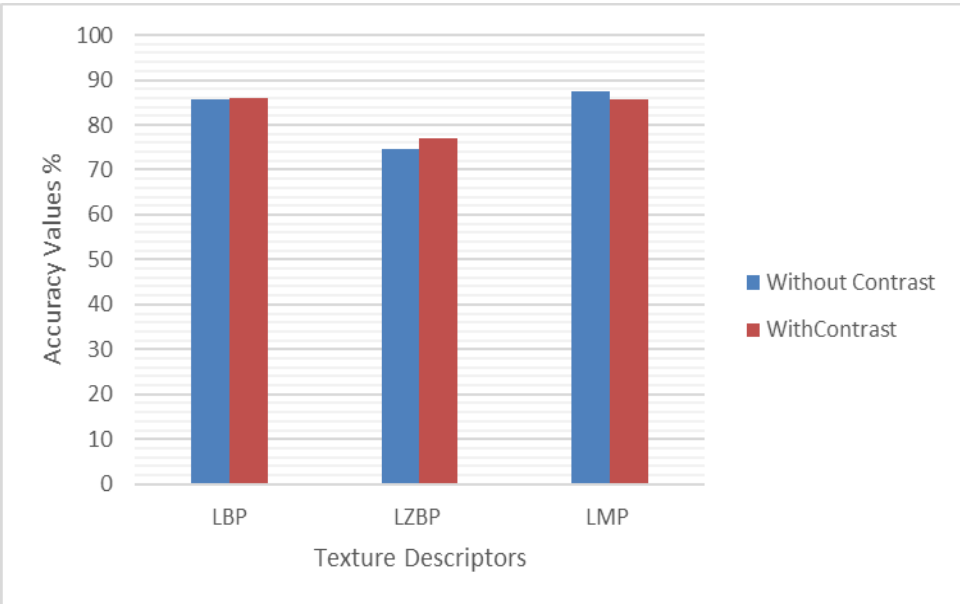


Fig. 6. 10 Comparison classification accuracy value of the LBP, LZBP, and LMP when combined with the contrast of the texture image based on the KTHTIP2b database.

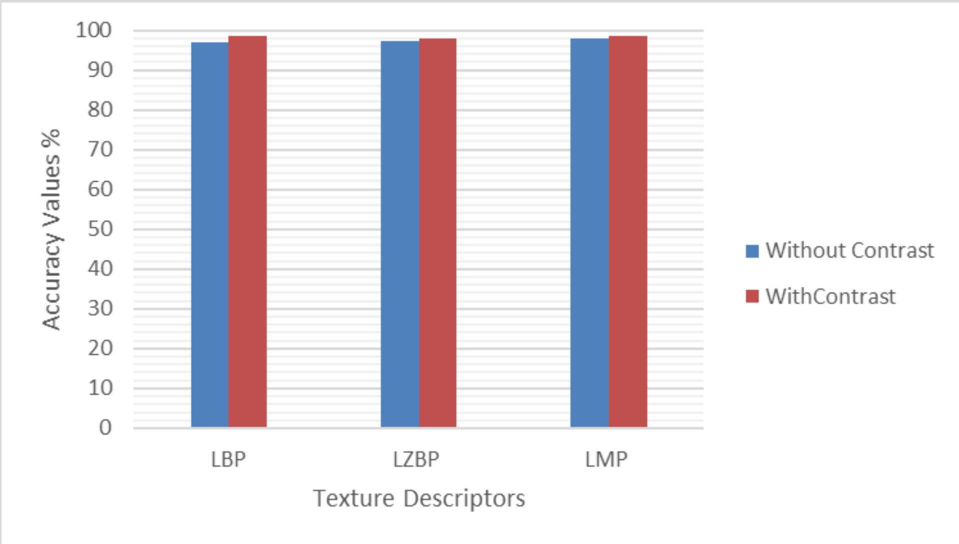


Fig. 6. 11 Comparison classification accuracy value of the LBP, LZBP, and LMP when combined with the contrast of the texture image based on the Brodatz database.

In general, it is clear from Fig 6-8 to Fig 6-11 that combining LBP, LZBP and LMP with contrast features as complementary features gave rise to better results than if these descriptors were used without combining to contrast features. However, from Fig 6.12, which compares the average of accuracy from the different databases, combining LBP, LZBP and LMP altogether with contrast features gave rise to better results than if these descriptors were combined separately with contrast features. This may due to the fact that the shared descriptors of LBP, LZBP and LMP depend on a different procedure for extracting local features from texture.

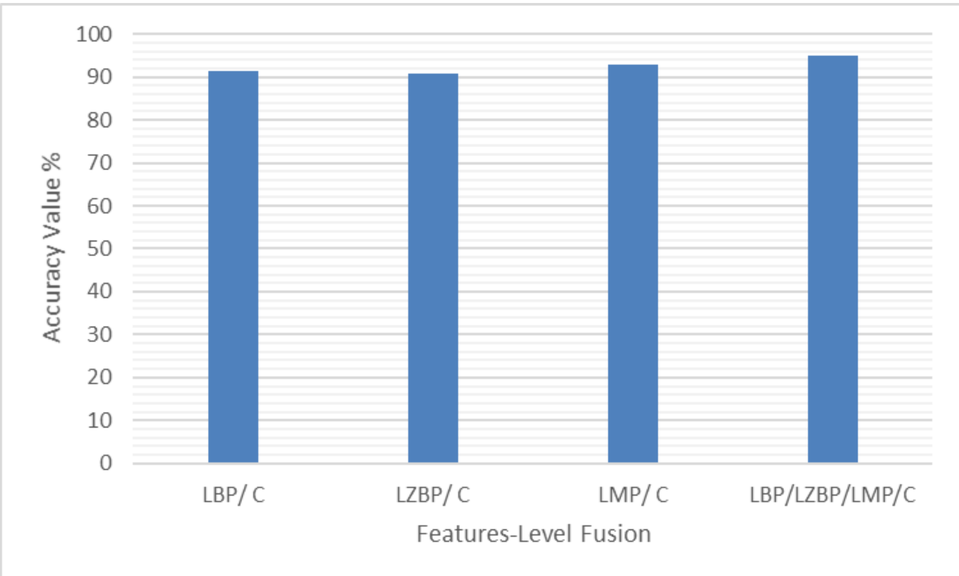


Fig. 6. 12 Comparison classification accuracy value of the LBP, LZBP and LMP when combined with contrast features separately, and when all are combined together.

6.2.2 Performance Comparison with Competing Methods

LZBP and LMP apply a different strategy to LBP for extracting features from texture. Feature fusion of complete feature set produced better results than using the feature set of each descriptor separately (see Fig 6.12). In addition, in the evaluation of the LZBP, LMP, and LBP descriptors in Subsection 6.1.3, it was noticed that these descriptors resulted in different results for different texture image classes. Furthermore, the proposed approach, feature fusion is processed by new selection methods to reduce feature length (see Subsection 5.2.2). The performance of local feature fusion with the proposed feature selection approach can be evaluated either by comparing the results before and after feature selection, or comparison with other related feature selection approaches (H. Liu & Motoda, 2007). The first type of comparison was conducted in the analysis of the results in Subsection 5.2.2.2, where it was found that the results obtained when using the relevant features of the combined feature descriptors offer significant improvement over those obtained using only single features or complete set of features by direct fusion with no feature selection.

Here, the evaluation of the proposed methods is based on comparing them with relevant methods that were reported in Subsection 2.3.2, such as CLBP (Z. Guo, Zhang, et al., 2010a), CINIRD (L. Liu et al., 2012), DLBP (Liao et al., 2009), and LBPV (Z. Guo, Zhang, et al., 2010b). Table 6.1 lists the results of the compared methods on the UIUC, UMD, and KTHTIP2b databases utilising the data in (L. Liu et al., 2017).

Table 6.1 compare the proposed approach (feature fusion, relevant feature by ABC and relevant feature ABC-NRS) with aforementioned methods. From the table, one can observe that CLBP produced the best results on the UIUC database, with an accuracy of 95.75%, whereas the proposed approach using ABC was nearly the same with feature size 1814. Based on feature selection using the ABC algorithm produced the best overall results on the UMD database, with an accuracy of 99%. The proposed approach obtained excellent result using the KTHTIP2b database when compared with the CLBP and CINIRD methods. In this context, CLBP used a huge feature size of 3552 features, whereas CINIRD suffers from feature dimension problems when using the features for multiscale (L. Liu et al., 2012). The direct feature fusion of the proposed LZBP and LMP descriptors with complementary features of LBP and contrast also resulted in a long feature length of 3189. Furthermore, from this comparison, only LBPV produced an acceptable feature length of only 158.

Despite the high performance achieved by most of these methods, the long feature length increases the classification computation time. However, depending on only the relevant features through the use of the proposed ANC-NRS feature selection approach achieved good classification accuracies while maintaining the ability to reduce the feature dimensionality (see Table 5.9).

Table. 6. 1 Classification accuracy results of various methods when applied to a number of databases.

Methods	UIUC	UMD	KTHTIP2b	Feature Dim
CLBP	95.75	98.62	64.18	3552
CINIRD	94.61	98.93	64.84	2200
DLBP	93.58	83.71	61.72	14150
LBPV	93.79	81.98	59.03	158
Feature Fusion	92	98.2	92.6	3189
Relevant feature by ABC	95.2	99	94.4	-
Relevant feature ABC-NRS	94.2	99	92.6	-

For evaluation on noise and blur problems, many methods designed to deal with noise and blur problems are based on MRELBP (L. Liu, Fieguth, Pietikäinen, & Lao, 2015), NTLBP (Fathi & Naghsh-Nilchi, 2012), or BRINT (L. Liu et al., 2014). The proposed approached was compared with the aforementioned methods by testing them all on the same Outex-11 database. Table 6.2 compares the performance of the methods for salt and pepper noise, whereas Table 6.3 compares their performance on blurred images. The results of the two tables clearly demonstrate the give effectivity of the proposed method with these image problems. For blurred images, classification by the proposed approach was particularly superior to other methods for $\sigma = 3.0$ and above.

Table. 6. 2 Classification accuracy results of various classification methods when applied to noisy images.

Method	$\rho = 0.05$	$\rho = 0.15$	$\rho = 0.30$	$\rho = 0.40$	$\rho = 0.50$
MRELBP	100.0	100.0	100.0	85.8	50.2
NTLBP	74.4	22.1	4.8	5.0	6.3
BRINT	30.8	7.1	6.0	4.4	4.2
Hybrid ABC-NRS	100	100	99.6875	99.27083	98.75

Table. 6. 3 Classification accuracy results of various classification methods when applied to blurred images.

Method	σ = 0.5	σ = 0.75	σ = 1.0	σ = 2.0	σ = 3.0	σ = 4.0	σ = 5.0
MRELBP	100.0	100.0	93.8	75.4	-	-	-
NTLBP	96.3	49.0	33.1	19.4	-	-	-
BRINT	100.0	97.1	80.4	44.6	-	-	-
Hybrid ABC-NRS	100	100	100	100	100	99.583	98.646

6.3 LZBP and LMP Combined with GF Features

The LZBP and LMP descriptors were also tested when combined with complementary GF features. GF is the most successful and appropriate complementary features used with LBP in many applications (M. Li & Staunton, 2008). Feature-level fusion is processed by the new hybrid selection approach to avoid the high dimensionality problem of the resulting features. Thus, the evaluation of LZBP and LMP was conducted when they were combined with GF, before utilising the new hybrid selection approach on the proposed feature fusion to select the optimum features.

6.3.1 Comparison of Feature Fusion Methods Based on the Confusion Matrix

In the experimental testing (see Subsection 5.2.3), LBP was found to result in enhanced accuracies when combined with GF. Similarly, the LZBP and LMP descriptors also produced improved accuracies when combined with GF. The improvement in the accuracy of LBP, LZBP and LMP when combined with GF compared if applied without combining with GF. It is therefore clear that LZBP and LMP features can be combined with complementary GF features to yield improved results.

For further investigation, the proposed methods (LZBP and LMP) where combined with GF and compared to combining LBP with GF on different databases. This was conducted based on the confusion matrix (refer to Appendix 2).

Group A is for UIUC database, where Tables A.1 to A.3 show the confusion matrix results of combining GF with LBP, LZBP and LMP respectively. From A.2, it is evident that the combining GF with LZBP is the most appropriate approach to dealing with UIUC database images compared to the other methods. The highest performance accuracy was 100% (40 out of 40) for 5 classes (7, 16, 17, 18, 22). This makes combining GF with LZBP more effective on UIUC classes than the conventional combination of GF with LBP, since the latter achieved its highest performance accuracy of only 97.5% (39 out of 40) for only 2 classes (16, 17). In contrast, the classes which the classification accuracy in this database when using GF with LZBP were Classes 4, 19, 23, and 25, which all had a classification accuracy of 85% (34 out of 40). For the combined LBP and GF approach, the lowest classification accuracy achieved was for Class 23 with 57.5 % (27 and 40), Class 25 with 60% (24 and 40), and Class 6 with 70% (28 and 40).

Group B (Tables B.1 to B.3) shows the results corresponding to the UMD database, where although the accuracy of GF with LBP, LZBP and LMP were 96%, 96.4% and 96.5% respectively (see Table 5.7), LBP with GF provided 100% accuracy for 18 classes, whereas combining LZBP with GF resulted in 14 classes with 100% accuracy, and which is five classes more than those classified using the conventional multi-scale LBP with 100% accuracy. LMP with GF provided 100% classification accuracy for only 12 classes.

In Group C, Tables C.1 to C.3 is for the KTHTIPS2b dataset, combining LBP with GF produced the highest possible accuracy of 100% for Class 1, followed by 9 classes with 98% classification accuracy. In comparison, combining LZBP with GF did not record good classification accuracies in these classes. The lowest classification accuracy for LBP with GF was obtained for Class 5, with 82%. Using LMP with GF, the first class had a good classification accuracy of 98%, which is 2% the classification accuracy achieved by LBP with GF. Other classes recording good classification accuracies with LMP were Class 10 with 97%, Class 4 with 96%, whereas the lowest was Class 5 with 80%.

Analysing the results, one can conclude that, in general, LZBP with GF produced superior classification performance for the UIUC database to others, while for the KTHTIPS2b database, the accuracies of the LZBP with GF was somewhat worse compared to conventional LBP with GF. While LBP with GF was effective on most classes of the UMD databases, LMP with GF achieved high classification rate for a larger number of classes of the database together.

6.3.2 Comparison of the Hybrid Selection Approach with Other Related Methods

Extracting only part of the features (or relevant features) from the complete set of features by the proposed wrapper ABC selection approach based on NRS was evaluated by a comparison with the Principal Component Analysis (PCA). PCA is the most successful reduction method used with LBP and GF features (Tan & Triggs, 2007).

Using PCA with LBP starts by applying PCA on the LBP histograms, then selecting a limited number of features, which results in producing the highest accuracy. Here, PCA was applied directly on the LBP histograms, before combining LBP with complementary GF features. This approach was compared with the proposed hybrid ABC-NRS selection method, where, as explained before, NRS is applied to reduce the feature length of multiscale LBP, LZBP and LMP.

To evaluate the performance according to the classification accuracy using the BPNN classifier, the three different LBP, LZBP and LMP feature descriptors were combined with GF after feature reduction. The implementation of PCA was coded by MATLAB, and the most important feature corresponding to 32 feature items were selected to be combined with GF features. For the proposed hybrid selection method, NRS was used to reduce the multiscale LBP, LZBP and LMP features, before combining with the optimum filters selected by ABC (see section 4.3.3.1). Comparing the performance of the two selection approaches, Fig 6.13 shows that the proposed hybrid ABC-NRS outperformed the reduction achieved by PCA for LBP, LZBP or LMP combined with GF features, where the difference in accuracy was 1.32%, 2.102%, and 1.348%, respectively.

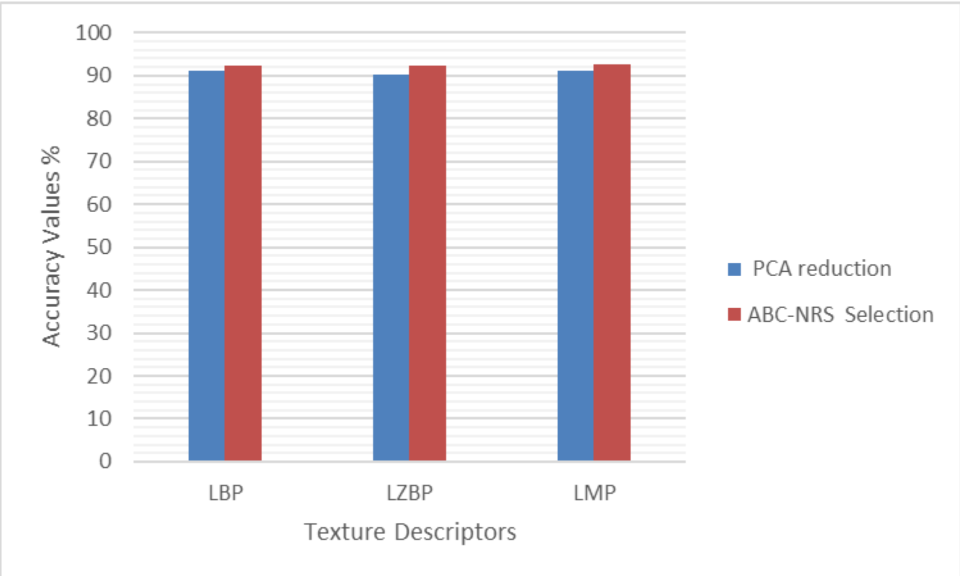


Fig. 6. 13 Comparison classification accuracy value of PCA and ABC-NRS performance for feature selection when applied on combined features of LBP, LZBP, and LMP with GF.

The proposed hybrid feature selection method combines the wrapper method (ABC) and the filter method (NRS). PCA also utilises the filter method, which is frequently applied for reducing the feature length of LBP and GF. For comparison, PCA replaced NRS in the hybrid ABC-PCA selection approach. Figure 6.14 shows the classification accuracy, where the proposed hybrid ABC-NRS outperformed the hybrid ABC-PCA for LZBP with GF features, whereas for LBP or LMP with GF features, the accuracy of ABC-PCA was only slightly higher than ABC-NRS. However, depending on PCA in the hybrid feature selection approach changed the semantics of the resulting feature, which do not exist in relevant features yielded by NRS (Guyon & Elisseeff, 2003).

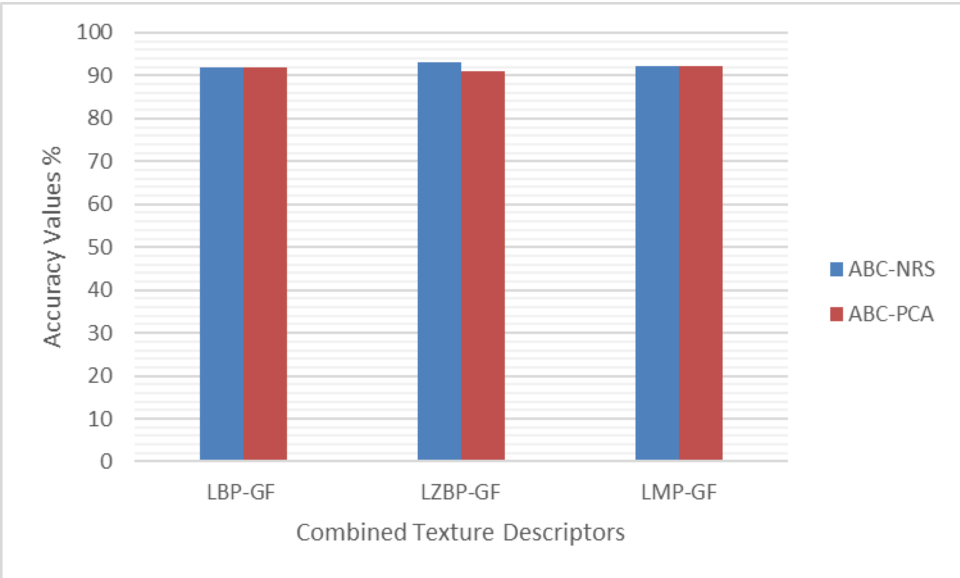


Fig. 6. 14 Comparison classification accuracy value of hybrid ABC-NRS and ABC-PCA performance for feature selection when applied on combined features of LBP, LZBP, and LMP with GF.

6.4 Summary

Evaluation the performance of improved feature extraction and selection methods was based on using the most appropriate methods and metrics that are commonly used in classification systems. The evaluation of new LZBP and LMP feature descriptors was based on using different datasets and the confusion matrix to investigate the effects of utilising these descriptors on the different texture characteristics and compare them with other related methods. Based on the BPNN and SVM classifiers, LZBP and LMP were found to improve the outcome of the classifiers compared to others texture descriptors. Furthermore, the results of the evaluation proved that the new descriptors improved the classification of features of most classes of texture characteristics that previous descriptors like LBP failed to classify.

The next steps proposed towards improving feature extraction of different texture characteristics were feature fusion followed by feature selection approaches. The feature selection approaches usually supplement feature fusion. The performed evaluation demonstrated improved results of the proposed approaches through two types of comparison. Firstly, the LZBP and LMP descriptors were shown to have the ability to complement other features, such as those corresponding to contrast and GF. The new combined features improved the features by increase the classification performance when compared with outcome of the methods separately. Secondly, the goal of improving feature reduction was successfully achieved by developing a new hybrid selection method based on the wrapper ABC and NRS filter algorithms. This hybrid selection method produced the least relevant features space when compared with other commonly used feature reduction methods, where it outperformed traditional feature reduction algorithms and achieved better image classification performance.

The advantage of the overall proposed approach is that it can extract powerful features by fusing the proposed feature descriptors with other complementary features, then processing the resulting features using the new feature selection method. The evaluation of the complete approach demonstrated its strength and the possible significant improvements in feature extraction that can be attained through adopting it in image classification applications.

Chapter 7

Conclusion

This research aimed to improve texture feature in image classification applications. At the beginning, this research stated several challenges associated with texture and its different characteristics. Furthermore, in this research, the research gap is defined as addressing the challenges faced by texture feature in order to improve texture-based image classification. While pursuing this line of research, it was observed that one of most effective ways to improve texture-based image classification is through the extraction of powerful and distinctive features. Subsequently, further key issues associated with improving features from texture for overall classification performance were specified. All objectives stated in Chapter 1 have been met.

Many studies have been conducted on texture features representation, and on addressing texture feature challenges. The best solutions that have been achieved so far included improving feature by either developing effective texture descriptors, feature-level fusion, and/or depending on only the relevant features using appropriate feature selection methods. All these procedures may participate together towards providing improved and powerful features from texture. An extensive review of the important related works was presented in Chapter 2. This included a discussion of the unique characteristics and properties of the most commonly used texture descriptors. Furthermore, our investigation studied the disadvantages and drawbacks of the current texture feature descriptors, and whether a single feature descriptor is satisfactory for extracting texture characteristics. Subsequently, an investigation was conducted of the problem resulting from combining a set of different features together, where such an approach results in increasing the feature space and giving rise to what is referred to by ‘‘curse of dimensionality’’.

7.1 Achievements and Novel Contributions

In the following subsections, a short review of main achievements of this research is presented, followed by research novel contributions.

7.1.1 Achievements

The aim of this research was to find a solution for previous problems related to improving texture feature extraction. The proposed solution is fundamentally based on developing effective texture descriptors. Furthermore, for more discriminative texture features, the developed texture descriptors are combined with other complementary features. In addition, to avoid the curse of dimensionality problem that may result from the feature fusion process, an improved feature selection method is used. The achievements of this research therefore match its main objective, which is to improve texture features detection for image classification applications.

Achievement 1: LZBP and LMP descriptors have been developed to extract texture features, where their development was motivated by the shortcoming of the LBP descriptor. The experimental results of Subsection 5.2.1 confirmed that LZBP and LMP are suited for effective textures classification, where these new feature descriptors usually show higher classification accuracies and improvements over the results of the LBP descriptor. The proposed new features descriptors were evaluated in Section 6.1 using a variety of benchmarks. The results of the evaluation revealed that LZBP and LMP are capable of yielding more distinctive texture feature characteristics than the majority of the most commonly used classical methods.

Achievement 2: The effectiveness of feature extraction from different texture characteristics has been improved by combining the extracted features by LZBP and LMP with the extracted features from contrast image and GF. From the literature, contrast images and GF have been used as complementary features to LBP features. In the experimental results of Subsection 5.2.2 and 5.2.3 and the evaluation in Subsection 6.2 and 6.3, it was indicated that integrating LZBP and LMP through the complementary features of contrast of the texture image or GF produced better results than applying these feature descriptors separately. The feature vector from the combined descriptors improved the accuracy of classification. However, the resulting feature length of the combined texture descriptors was long and computationally expensive.

Achievement 3: A hybrid ABC-NRS method has been developed and applied to select only the relevant features from the proposed combined features. The hybrid selection method dealt efficiently with resulting feature space of the combined texture descriptors. From the experimental results of Subsection 5.2.2 and 5.2.3, combining the proposed feature descriptors with complementary features, and subsequently processing the resulting features by the proposed selection approach produced a significantly improved classification performance. In the results of Subsection 5.2.2.3, the fusion approach involving combining LBP, LZBP and LMP with local image contrast proved to be tolerant to image problems such as blur and noise, providing better texture feature classification performance than other existing methods. Furthermore, in Subsection 5.2.3.2 combining LZBP and LMP with complementary GF features, improved the extracted features with the least feature length. Overall, as show in sections 6.1 and 6.2, there are several methods were utilised throughout the development and evaluation process for proposed models to ensure an appropriate and fair assessment of the developed methods.

7.1.2 Novel Contributions

This research proposes novel texture descriptors and improved feature selection methods for images classification applications. The proposed texture descriptors represent a new way of extracting features from texture. In addition, according to the author's knowledge, the following attributes of the improved feature extraction and selection methods have not been reported in the literature:

Novel Local Texture Descriptors

In this research, local texture descriptors, namely LZBP and LMP, have been developed as novel methods of feature extraction. These texture descriptors take more consideration of pixels' intensity values in TUs than LBP. Similar to LBP, the extraction of features from the texture by LZBP and LMP is done via three steps, the first of which involves converting the original image into local texture patterns. This is done by constructing quantisation zones for thresholding the intensity values of pixels. Then, suitable weighting coefficients that correspond to the local patterns of descriptors are prepared, which are utilised in computing the matrices of LZBP and LMP. Finally, the statistics of local textures are computed by applying the histogram to the resulting matrices of LZBP and LMP.

The mathematical details of the LZBP and LMP feature extraction processes can be found Subsection 3.3.1.2, while the implementation algorithms can be referred to in Subsection 4.2.

Improved Feature Selection Method

The other contribution of this research is the improved feature selection approach. By the proposed hybrid ABC-NRS feature selection method, which takes advantages of wrapper and filter methods, the limitations of the previous feature selection methods can be overcome removing part of the feature space in order to reduce the computation cost and avoid the curse of dimensionality. Subsection 4.3.3 explained the implementation of the improved selection method for the proposed feature fusion.

In Subsection 3.3.2, feature-level fusion models were presented. The proposed feature-level fusion models produced a distinctive feature vector, however, at same time, applying the concatenation style of feature-level fusion generated a large feature space, which required intensive computational power. A major part of the selection approach's success is due to its capability to effectively remove the redundant features while retaining the distinctive and characteristic features, which thus reduce the feature length while maintaining excellent classification accuracy. The use of the Rough Set method was shown to reduce the feature length of the histogram resulting from the multi-scale LBP, LZBP and LMP descriptors, whereas the ABC algorithm was demonstrated to result in selecting the optimum features resulting from Gabor filter.

7.2 Research limitations and Future Work

While the proposed techniques have succeeded in producing valuable results, further proposals can be made to provide further improvement in the classification results.

7.2.1 Research Limitations

1. Feature extraction by LZBP and LMP mostly produces better information about different types of texture characteristics. Compared to LBP, the proposed texture descriptors extract this information by constructing more quantization zones for the intensity values of pixels in TUs. However, this results in longer feature length, especially for LMP, where the feature length is derived from the weights' coefficients for the corresponding quantized intensity values of pixels. This may have negative impacts on the computation time for online applications, which was the main advantage of using LBP.

2. From the experimental results, the classification rate by the hybrid feature selection method is higher than that resulting from combining the features directly without feature selection. Although the hybrid ABC-NRS also produces the least feature length, the classification rate by the hybrid ABC-NRS is slightly lower than applying ABC alone for feature selection.

7.2.2 Future Work

1. Our approach relied on developing effective methods for improving feature extraction and selection. The proposed methods can be widely used in classification systems, or be applied in other applications such as feature selection to avoid the curse of dimensionality. Furthermore, the LZBP and LMP descriptors can be relied on wider variety of different image classification applications of other texture descriptors, such as the LBP descriptor (see Subsection 2.1.3.2).
2. In context of the first limitation point; while LZBP and LMP offer outstanding feature extraction performance, the feature length of LMP is somewhat long, as it is increased by the zones in the neighbour places. In our implementation, for a shorter LMP histogram, the 16-bit patterns resulting from TUs were divided into 8-bit diagonal patterns and 8-bit non-diagonal patterns. However, there are other methods and approaches that can be utilised for a shorter histogram, such as the centre-symmetric LBP method (Heikkilä et al., 2009), or depending on ‘‘uniform’’ texture patterns (Ojala, Pietikainen, et al., 2002). These methods can be tested with LMP before combined its features with other complementary features and applying feature selection. This may also improve the performance of the feature selection method, as it would have to deal with a shorter feature space.
3. In context of the second limitation point; the work based on the hybrid feature selection method was based on combining the ABC algorithm and the NRS method. The ABC algorithm was fully automated with initial parameters, whereas the NRS method required manual determination of the distance that replaces the equivalence neighbourhood relation in RS. Depending on a range of distances in different datasets that may improve the performance of the selected features by NRS. Thus, improving feature selection by NRS can be achieved by determining the appropriate distance automatically. Achieving this feat for the hybrid feature selection method could be result in improvements in the resulting classified features and classification performance.

References

Afshang, M., Helfroush, M. S., & Zahernia, A. (2009). Gabor filter parameters optimization for texture classification based on genetic algorithm. *Second International Conference on Machine Vision (pp. 199-203).*

Ahmadian, A., & Mostafa, A. (2003). An efficient texture classification algorithm using Gabor wavelet. Paper presented at the Engineering in Medicine and Biology Society, Proceedings of the 25th Annual International Conference of the IEEE, (Vol. 1, pp. 930-933).

Ahmed, F., Hossain, E., Bari, A. H., & Shihavuddin, A. (2011). Compound local binary pattern (CLBP) for robust facial expression recognition. *Paper presented at the Computational Intelligence and Informatics (CINTI), 2011 IEEE 12th International Symposium on,* (pp. 391-395).

Al-Janobi, A. (2001). Performance evaluation of cross-diagonal texture matrix method of texture analysis. *Pattern recognition*, 34(1), 171-180.

Amato, A., & Di Lecce, V. (2008). A knowledge based approach for a fast image retrieval system. *Image and Vision Computing, 26*(11), 1466-1480.

Arivazhagan, S., Ganesan, L., & Priyal, S. P. (2006). Texture classification using Gabor wavelets based rotation invariant features. *Pattern recognition letters*, *27*(16), 1976-1982.

Atasever, Ü. H., Özkan, C., & Sunar, F. (2011). The use of artificial intelligence optimization algorithms in unsupervised classification. *Remote Sensing and Geoinformation not only for Scientific Coorperation, EARSeL.*

Bae, C., Yeh, W.-C., Chung, Y. Y., & Liu, S.-L. (2010). Feature selection with intelligent dynamic swarm and rough set. *Expert Systems with Applications, 37*(10), 7026-7032.

Banerjee, S., Bharadwaj, A., Gupta, D., & Panchal, V. (2012). Remote sensing image classification using artificial bee colony algorithm. *International Journal of Computer Science and Informatics, 2*(3), 67-72.

Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms, 5*(1-2), 123-159.

Barley, A., & Town, C. (2014). Combinations of feature descriptors for texture image classification. *Journal of Data Analysis and Information Processing, 2*(03), 67.

Bashar, M. K., & Ohnishi, N. (2002). Fusing cortex transform and intensity based features for image texture classification. *Paper presented at the Information Fusion, Proceedings of the Fifth International Conference on* (Vol. 2, pp. 1463-1469).

Belongie, S., Carson, C., Greenspan, H., & Malik, J. (1998). Color-and texture-based image segmentation using EM and its application to content-based image retrieval. *In Sixth International Conference on Computer Vision* (pp. 675-682).

Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research, 5*(Sep), 1089-1105.

Bianconi, F., & Fernández, A. (2007). Evaluation of the effects of Gabor filter parameters on texture classification. *Pattern recognition, 40*(12), 3325-3335.

Biswas, A., Mishra, K., Tiwari, S., & Misra, A. (2013). Physics-inspired optimization algorithms: a survey. *Journal of Optimization,* ID 438152, 16 pages

Blum, A. L., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial intelligence,* 97(1-2), 245-271.

Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3), 268-308.

Blum, R. S., & Liu, Z. (2005). *Multi-sensor image fusion and its applications*: CRC press.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *From natural to artificial swarm intelligence*: Oxford university press.

Bonabeau, E., Marco, D. d. R. D. F., Dorigo, M., Théraulaz, G., & Theraulaz, G. (1999). *Swarm intelligence: from natural to artificial systems*: Oxford university press.

Bonabeau, E., Theraulaz, G., & Deneubourg, J.-L. (1996). Quantitative study of the fixed threshold model for the regulation of division of labour in insect societies. *Proceedings of the Royal Society of London. Series B: Biological Sciences,* 263(1376), 1565-1569.

Brahnam, S., Jain, L. C., Nanni, L., & Lumini, A. (2014). *Local binary patterns: new variants and applications*: Springer Berlin Heidelberg.

Brodatz, P. (1966). Textures: a photographic album for artists and designers: *Dover Pubns.*

Brownlee, J. (2011). Clever algorithms: nature-inspired programming recipes: *Jason Brownlee.*

Candade, N., & Dixon, B. (2004). Multispectral classification of Landsat images: a comparison of support vector machine and neural network classifiers. *Paper presented at the ASPRS Annual Conference Proceedings, Denver, Colorado.*

Castano, R., Manduchi, R., & Fox, J. (2001). Classification experiments on real-world texture. *Empirical Evaluation Methods in Computer Vision* (pp. 1-20).

Chaki, J., Parekh, R., & Bhattacharya, S. (2015). Plant leaf recognition using texture and shape features with neural classifiers. *Pattern recognition letters, 58*, 61-68.

Chan, C.-H., Kittler, J., & Messer, K. (2007). Multi-scale local binary pattern histograms for face recognition. *Paper presented at the International conference on biometrics.* (pp. 809-818).

Chaudhuri, B., Sarkar, N., & Kundu, P. (1993). Improved fractal geometry based texture segmentation technique. *IEEE Proceedings (Computers and Digital Techniques),* 140(5), 233-242.

Chellappa, R., Kashyap, R., & Manjunath, B. (1993). Model-based texture segmentation and classification. In *Handbook of pattern recognition and computer vision,* (pp. 277-310).

Chen, L., Lu, G., & Zhang, D. (2004). Effects of different Gabor filters parameters on image retrieval by texture. *10th International Multimedia Modelling Conference* (pp. 273-278).

Chen, Y., Miao, D., & Wang, R. (2010). A rough set approach to feature selection based on ant colony optimization. *Pattern recognition letters, 31*(3), 226-233.

Clausi, D. A., & Deng, H. (2005). Design-based texture feature fusion using Gabor filters and co-occurrence probabilities. *IEEE transactions on image processing, 14*(7), 925-936.

Clausi, D. A., & Jernigan, M. E. (2000). Designing Gabor filters for optimal texture separability. *Pattern recognition, 33*(11), 1835-1849.

Conners, R. W., & Harlow, C. A. (1980). A theoretical comparison of texture algorithms. *IEEE Transactions on pattern analysis and machine intelligence* (3), 204-222.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

Crosier, M., & Griffin, L. D. (2008). Texture classification with a dictionary of basic image features. *Paper presented at Conference on Computer Vision and Pattern Recognition.* (pp. 1-7).

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis, 1*(3), 131-156.

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, 2*(7), 1160-1169.

Dell'Acqua, F., & Gamba, P. (2003). Texture-based characterization of urban environments on satellite SAR images. *IEEE Transactions on geoscience and remote sensing, 41*(1), 153-159.

Deng, Y., & Manjunath, B. (2001). Unsupervised segmentation of color-texture regions in images and video. *IEEE Transactions on pattern analysis and machine intelligence, 23*(8), 800-810.

Deng, Z., Gu, H., Feng, H., & Shu, B. (2011). Artificial bee colony based mapping for application specific network-on-chip design. *Paper presented at the International Conference in Swarm Intelligence.* (pp. 285-292).

Deogun, J. S., Choubey, S. K., Raghavan, V. V., & Sever, H. (1998). Feature selection and effective classifiers. *Journal of the American Society for Information Science, 49*(5), 423-434.

Di Cataldo, S., & Ficarra, E. (2017). Mining textural knowledge in biological images: Applications, methods and trends. *Computational and structural biotechnology journal, 15*, 56-67.

Digalakis, J. G., & Margaritis, K. G. (2002). An experimental study of benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics, 79*(4), 403-416.

Dorigo, M., & Birattari, M. (2010). Ant colony optimization. *IEEE Computational Intelligence Magazine,* (pp. 36-39).

Duan, H., Deng, Y., Wang, X., & Xu, C. (2013). Small and dim target detection via lateral inhibition filtering and artificial bee colony based selective visual attention. *Journal PloS one,* 8(8), 1-12

Duda, R. O., Hart, P. E., & Stork, D. G. (2012). *Pattern classification*: John Wiley & Sons. (2nd ed.).

Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Paper presented at the Micro Machine and Human Science,* (pp. 39-43).

Evgeniou, T., Pontil, M., & Poggio, T. (2000). Statistical learning theory: A primer. *International Journal of Computer Vision, 38*(1), 9-13.

Fan, G., & Xia, X.-G. (2003). Wavelet-based texture analysis and synthesis using hidden Markov models. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 50*(1), 106-120.

Fathi, A., & Naghsh-Nilchi, A. R. (2012). Noise tolerant local binary pattern operator for efficient texture analysis. *Pattern recognition letters, 33*(9), 1093-1100.

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering, 93(26), 429-441.

Galloway, M. M. (1974). Texture analysis using grey level run lengths. *NASA STI/Recon Technical Report N, 75*.

Gheyas, I. A., & Smith, L. S. (2010). Feature subset selection in large dimensionality domains. *Pattern recognition, 43*(1), 5-13.

Gozde, H., Taplamacioglu, M. C., & Kocaarslan, I. (2012). Comparative performance analysis of Artificial Bee Colony algorithm in automatic generation control for interconnected reheat thermal power system. *International Journal of Electrical Power & Energy Systems, 42*(1), 167-178.

Guo, Y., Zhao, G., & PietikäInen, M. (2012). Discriminative features for texture description. *Pattern recognition, 45*(10), 3834-3843.

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

Guo, Z., Zhang, L., & Zhang, D. (2010b). Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern recognition, 43*(3), 706-719.

Guo, Z., Zhang, L., Zhang, D., & Zhang, S. (2010). Rotation invariant texture classification using adaptive LBP with directional statistical features. *Paper presented at the Image Processing (ICIP), 17th IEEE International Conference on.* (pp. 285-288).

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research, 3*(Mar), 1157-1182.

Haghighat, M., Zonouz, S., & Abdel-Mottaleb, M. (2015). CloudID: Trustworthy cloud-based and cross-enterprise biometric identification. *Expert Systems with Applications, 42*(21), 7905-7916.

Haralick, R. M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE, 67*(5), 786-804.

Haralick, R. M., & Shanmugam, K. (1973). Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*(6), 610-621.

He, D.-C., & Wang, L. (1990). Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on geoscience and remote sensing, 28*(4), 509-512.

He, D.-C., & Wang, L. (2010). Simplified texture spectrum for texture analysis. *Journal of Communication and Computer, 7*(8), 44-53.

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

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press.

Hoque, N., Bhattacharyya, D., & Kalita, J. K. (2014). MIFS-ND: a mutual information-based feature selection method. *Expert Systems with Applications, 41*(14), 6371-6385.

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.

Hu, K., Lu, Y., & Shi, C. (2003). Feature ranking in rough sets. *AI communications, 16*(1), 41-50.

Hu, Q., Yu, D., Liu, J., & Wu, C. (2008). Neighborhood rough set based heterogeneous feature subset selection. *Information sciences, 178*(18), 3577-3594.

Hu, Q., Yu, D., & Xie, Z. (2008). Neighborhood classifiers. *Expert Systems with Applications, 34*(2), 866-876.

Huet, F., & Mattioli, J. (1996). A textural analysis by mathematical morphology transformations: Structural opening and top-hat. *Paper presented at the Image Processing,* (Vol. 3, pp. 49-52).

Hussain, S. U., & Triggs, W. (2010). Feature sets and dimensionality reduction for visual object detection. *Paper presented at the BMVC Machine Vision Conference*. (pp. 112-1).

Idrissa, M., & Acheroy, M. (2002). Texture classification using Gabor filters. *Pattern recognition letters, 23*(9), 1095-1102.

Ivanciuc, O. (2007). Applications of support vector machines in chemistry. *Reviews in computational chemistry, 23*, 291.

Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on pattern analysis and machine intelligence, 19*(2), 153-158.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence, 22*(1), 4-37.

Jain, A. K., & Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern recognition, 24*(12), 1167-1186.

Jayanth, J., Koliwad, S., & Kumar, T. A. (2015). Classification of remote sensed data using Artificial Bee Colony algorithm. *The Egyptian Journal of Remote Sensing and Space Science, 18*(1), 119-126.

Jensen, R., & Shen, Q. (2007). Fuzzy-rough sets assisted attribute selection. *IEEE Transactions on fuzzy systems, 15*(1), 73-89.

Kamarainen, J., Kyrki, V., & Kälviäinen, H. (2002). Fundamental frequency Gabor filters for object recognition. *In Object recognition supported by user interaction for service robots* (Vol. 1, pp. 628-631).

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Retrieved from (Vol. 200). Erciyes university, engineering faculty, computer engineering department.

Karaboga, D., & Akay, B. (2009). Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization. *Paper presented at the Innovative production machines and systems virtual conference.*

Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation, 214*(1), 108-132.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization, 39*(3), 459-471.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing, 8*(1), 687-697.

Karu, K., Jain, A. K., & Bolle, R. M. (1996). Is there any texture in the image?. *Pattern recognition, 29*(9), 1437-1446.

Ke, L., Feng, Z., & Ren, Z. (2008). An efficient ant colony optimization approach to attribute reduction in rough set theory. *Pattern recognition letters, 29*(9), 1351-1357.

Khalid, M., Yusof, R., & Meriaudeau, F. (2010). A comparative study of feature extraction methods for wood texture classification. *Paper presented at the Signal-Image Technology and Internet-Based Systems,* (pp. 23-29).

Khellah, F. M. (2011). Texture classification using dominant neighborhood structure. *IEEE transactions on image processing, 20*(11), 3270-3279.

Kim, J. K., & Park, H. W. (1999). Statistical textural features for detection of microcalcifications in digitized mammograms. *IEEE transactions on medical imaging, 18*(3), 231-238.

Kim, K. I., Jung, K., Park, S. H., & Kim, H. J. (2002). Support vector machines for texture classification. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(11), 1542-1550.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence, 97*(1-2), 273-324.

Kohonen, T. (1990). The self-organizing map. Proceedings of the IEEE, 78(9), 1464-1480.

Krishnanand, K., Nayak, S. K., Panigrahi, B. K., & Rout, P. K. (2009). Comparative study of five bio-inspired evolutionary optimization techniques. *Paper presented at the Nature & Biologically Inspired Computing,* (pp. 1231-1236).

Kyrki, V., Kamarainen, J.-K., & Kälviäinen, H. (2004). Simple Gabor feature space for invariant object recognition. *Pattern recognition letters*, *25*(3), 311-318.

Laine, A., & Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on pattern analysis and machine intelligence, 15*(11), 1186-1191.

Langley, P. (1994). Selection of relevant features in machine learning. *Paper presented at the Proceedings of the AAAI Fall symposium on relevance,* (pp. 1-5).

Laws, K. I. (1980). *Textured image segmentation*. University of Southern California Los Angeles Image Processing INST.

Lazebnik, S., Schmid, C., & Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on pattern analysis and machine intelligence*, *27*(8), 1265-1278.

Li, M., & Staunton, R. C. (2008). Optimum Gabor filter design and local binary patterns for texture segmentation. *Pattern recognition letters, 29*(5), 664-672.

Li, R., & Wang, Z.-o. (2004). Mining classification rules using rough sets and neural networks. *European Journal of Operational Research*, *157*(2), 439-448.

Li, S., & Shawe-Taylor, J. (2005). Comparison and fusion of multiresolution features for texture classification. *Pattern recognition letters, 26*(5), 633-638.

Li, W., Mao, K., Zhang, H., & Chai, T. (2010). Designing compact Gabor filter banks for efficient texture feature extraction. *Paper presented at the Control Automation Robotics & Vision (ICARCV),* (pp. 1193-1197).

Liao, S., Law, M. W., & Chung, A. C. (2009). Dominant local binary patterns for texture classification. *IEEE transactions on image processing, 18*(5), 1107-1118.

Lin, H., Wang, J.-y., & Liu, S. (2010). The Classification Study of Texture Image Based on the Rough Set Theory. *Paper presented at the Granular Computing (GrC),* (pp. 720-723).

Lin, W. C., Hays, J., Wu, C., Kwatra, V., & Liu, Y. (2004). A comparison study of four texture synthesis algorithms on regular and near-regular textures. *Tech. Rep. School of Computer Science Carnegie Mellon University College of Computing.*

Liu, H., & Motoda, H. (2007). *Computational methods of feature selection*: CRC Press.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering, 17*(4), 491-502.

Liu, L., Fieguth, P., Guo, Y., Wang, X., & Pietikäinen, M. (2017). Local binary features for texture classification: taxonomy and experimental study. *Pattern recognition, 62*, 135-160.

Liu, L., Lao, S., Fieguth, P. W., Guo, Y., Wang, X., & Pietikäinen, M. (2016). Median robust extended local binary pattern for texture classification. *IEEE Transactions on Image Processing*, 25(3), 1368-1381.

Liu, L., Fieguth, P., Wang, X., Pietikäinen, M., & Hu, D. (2016). Evaluation of LBP and deep texture descriptors with a new robustness benchmark. *Paper presented at the European Conference on Computer Vision,* (pp. 69-86).

Liu, L., Long, Y., Fieguth, P. W., Lao, S., & Zhao, G. (2014). BRINT: binary rotation invariant and noise tolerant texture classification. *IEEE transactions on image processing, 23*(7), 3071-3084.

Liu, L., Zhao, L., Long, Y., Kuang, G., & Fieguth, P. (2012). Extended local binary patterns for texture classification. *Image and Vision Computing, 30*(2), 86-99.

Lu, D., & Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International journal of remote sensing, 28*(5), 823-870.

Ma, W.-Y., & Manjunath, B. (1995). A comparison of wavelet transform features for texture image annotation. *Paper presented at the Image Processing,* (Vol. 2, pp. 256-259).

Mäenpää, T., & Pietikäinen, M. (2005). Texture analysis with local binary patterns *Handbook of pattern recognition and computer vision* (pp. 197-216): World Scientific.

Majid, M., & Xianghua, X. (2008). *Handbook of Texture Analysis*: World Scientific.

Mallikarjuna, P., Targhi, A. T., Fritz, M., Hayman, E., Caputo, B., & Eklundh, J.-O. (2006). The kth-tips2 database. *Computational Vision and Active Perception Laboratory (CVAP), Stockholm, Sweden*.

Manjunath, B. S., & Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on pattern analysis and machine intelligence, 18*(8), 837-842.

Manthalkar, R., Biswas, P. K., & Chatterji, B. N. (2003). Rotation invariant texture classification using even symmetric Gabor filters. *Pattern recognition letters, 24*(12), 2061-2068.

Mohammadi, F. G., & Abadeh, M. S. (2014). Image steganalysis using a bee colony based feature selection algorithm. *Engineering Applications of Artificial Intelligence, 31*, 35-43.

Murray, H., Lucieer, A., & Williams, R. (2010). Texture-based classification of sub-Antarctic vegetation communities on Heard Island. *International Journal of Applied Earth Observation and Geoinformation, 12*(3), 138-149.

Nailon, W. H. (2010). Texture analysis methods for medical image characterisation. *Biomedical imaging*. *In Biomedical imaging. IntechOpen,* (pp. 75-100).

Nailon, W. H., McLaughlin, S., Spencer, T., & Ramo, M. P. (1996). Comparative study of textural analysis techniques to characterise tissue from intravascular ultrasound. *In Proceedings of 3rd IEEE International Conference on Image Processing,* (Vol. 3, pp. 303-306).

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

Ojala, T., Maenpaa, T., Pietikainen, M., Viertola, J., Kyllonen, J., & Huovinen, S. (2002). Outex-new framework for empirical evaluation of texture analysis algorithms. *Paper presented at the Pattern Recognition. In Object recognition supported by user interaction for service robots* (Vol. 1, pp. 701-706).

Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. *Paper presented at the Pattern Recognition,* (Vol. 1, pp. 582-585).

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

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

Olaode, A., Naghdy, G., & Todd, C. (2014). Unsupervised classification of images: A review. *International Journal of Image Processing, 8*(5), 325-342.

Ozkan, C., Ozturk, C., Sunar, F., & Karaboga, D. (2011). The artificial bee colony algorithm in training artificial neural network for oil spill detection. *Neural Network World, 21*(6), 473.

Pakdel, M., & Tajeripour, F. (2011). Texture classification using optimal Gabor filters. *Paper presented at the Computer and Knowledge Engineering (ICCKE),* (pp. 208-213). IEEE.

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

Pawlak, Z. (1982). Rough sets. International journal of computer & information sciences, 11(5), 341-356.

Pawlak, Z. (1997). Rough set approach to knowledge-based decision support. *European Journal of Operational Research, 99*(1), 48-57.

Pawlak, Z. (2012). *Rough sets: Theoretical aspects of reasoning about data*: Springer Science & Business Media.

Pereira, F. H., & Sassi, R. J. (2012). Rough sets and principal components analysis: A comparative study on customer database attributes selection. *African Journal of Business Management, 6*(10), 3822 -3828.

Petrou, M., & Sevilla, P. G. (2006). Image processing: dealing with texture. *Chichester: John Wiley & Sons,* (pp. 329-379).

Pietikäinen, M., Hadid, A., Zhao, G., & Ahonen, T. (2011). *Computer vision using local binary patterns*: Springer Science & Business Media.

Pohl, C., & Van Genderen, J. L. (1998). Review article multisensor image fusion in remote sensing: concepts, methods and applications. *International journal of remote sensing, 19*(5), 823-854.

Pratt, W. K. (1991). Image segmentation. *Digital Image Processing: PIKS Inside, Third Edition*, 551-587.

Rahmat-Samii, Y. (2007). Modern antenna designs using nature inspired optimization techniques: Let darwin and the bees help designing your multi band MIMO antennas. *Paper presented at the Radio and Wireless Symposium,* (pp. 463-466).

Rajpoot, K. M., & Rajpoot, N. M. (2004). Wavelets and support vector machines for texture classification. *Paper presented at the Multitopic Conference,* (pp. 328-333). IEEE.

Randen, T., & Husoy, J. H. (1999). Filtering for texture classification: A comparative study. *IEEE Transactions on pattern analysis and machine intelligence, 21*(4), 291-310.

Rao, A. R., & Lohse, G. L. (1993). Identifying high level features of texture perception. *CVGIP: Graphical Models and Image Processing, 55*(3), 218-233.

Robinson, G. E. (1992). Regulation of division of labor in insect societies. *Annual review of entomology, 37*(1), 637-665.

Sathya, D. J., & Geetha, K. (2013). Mass classification in breast DCE-MR images using an artificial neural network trained via a bee colony optimization algorithm. *ScienceAsia, 39*(3), 294-305.

Satpathy, A., Jiang, X., & Eng, H.-L. (2014). LBP-based edge-texture features for object recognition. *IEEE transactions on image processing, 23*(5), 1953-1964.

Schiezaro, M., & Pedrini, H. (2013). Data feature selection based on Artificial Bee Colony algorithm. *EURASIP Journal on Image and Video Processing, 2013*(1), 47.

Shan, C., & Gritti, T. (2008). Learning Discriminative LBP-Histogram Bins for Facial Expression Recognition. *Paper presented at the BMVC,* (pp. 1-10).

Shang, C., & Shen, Q. (2008). Aiding neural network based image classification with fuzzy-rough feature selection. *Paper presented at the Fuzzy Systems,* (pp. 976-982).

Shanthi, S., & Bhaskaran, V. M. (2014). Modified artificial bee colony based feature selection: a new method in the application of mammogram image classification. *International Journal of Science, Engineering and Technology Research (IJSETR)*, 3(6), 1664-1667.

Shrivastava, N., & Tyagi, V. (2014). An effective scheme for image texture classification based on binary local structure pattern. *The Visual Computer, 30*(11), 1223-1232.

Siew, L. H., Hodgson, R. M., & Wood, E. J. (1988). Texture measures for carpet wear assessment. *IEEE Transactions on pattern analysis and machine intelligence*, *10*(1), 92-105.

Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied soft computing*, *9*(2), 625-631.

Singh, M., & Singh, S. (2002). Spatial texture analysis: a comparative study. *Paper presented at the Pattern Recognition,* (Vol. 1, pp. 676-679).

Sklansky, J. (1978). Image segmentation and feature extraction. *IEEE Transactions on systems, man, and cybernetics*, *8*(4), 237-247.

Soh, L.-K., & Tsatsoulis, C. (1999). Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices. *IEEE Transactions on geoscience and remote sensing, 37*(2), 780-795.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427-437.

Solberg, A. S. (1996). Texture fusion and classification based on flexible discriminant analysis. *Paper presented at the Pattern Recognition,* (Vol. 2, pp. 596-600). IEEE.

Solberg, A. S., & Jain, A. K. (1997). Texture fusion and feature selection applied to SAR imagery. *IEEE Transactions on geoscience and remote sensing, 35*(2), 475-479.

Song, A., & Ciesielski, V. (2003). Fast texture segmentation using genetic programming. *Paper presented at the Evolutionary Computation,* (Vol. 3, pp. 2126-2133).

Song, K., Yan, Y., Zhao, Y., & Liu, C. (2015). Adjacent evaluation of local binary pattern for texture classification. *Journal of Visual Communication and Image Representation, 33*, 323-339.

Sonka, M., Hlavac, V., & Boyle, R. (2014). *Image processing, analysis, and machine vision*: Cengage Learning (4th ed.).

Sun, Z., Bebis, G., & Miller, R. (2003). Evolutionary gabor filter optimization with application to vehicle detection. *Paper presented at the Data Mining*, (pp. 307-314). IEEE.

Svozil, D., Kvasnicka, V., & Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. *Chemometrics and intelligent laboratory systems, 39*(1), 43-62.

Tahooneh, A., & Ziarati, K. (2011). Using artificial bee colony to solve stochastic resource constrained project scheduling problem. *Paper presented at the International Conference in Swarm Intelligence*, (pp. 293-302).

Talatahari, S., Mohaggeg, H., Najafi, K., & Manafzadeh, A. (2014). Solving parameter identification of nonlinear problems by artificial bee colony algorithm. *Mathematical Problems in Engineering*.

Tamura, H., Mori, S., & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on systems, man, and cybernetics, 8*(6), 460-473.

Tan, X., & Triggs, B. (2007). Fusing Gabor and LBP feature sets for kernel-based face recognition. *Paper presented at the International Workshop on Analysis and Modeling of Faces and Gestures*, (pp. 235-249).

Tan, X., & Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing, 19*(6), 1635-1650.

Topi, M., Timo, O., Matti, P., & Maricor, S. (2000). Robust texture classification by subsets of local binary patterns. *Paper presented at the Pattern Recognition*, (Vol. 3, pp. 935-938).

Tou, J. Y., Tay, Y. H., & Lau, P. Y. (2009). Recent trends in texture classification: a review. *Paper presented at the Symposium on Progress in Information & Communication Technology*, (Vol. 3, No. 2, pp. 56-59).

Tsai, C.-C., Taur, J.-S., & Tao, C.-W. (2009). Iris recognition using Gabor filters optimized by the particle swarm algorithm. *Journal of Electronic Imaging, 18*(2), 023009.

Tu, J. V. (1996). Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of clinical epidemiology, 49*(11), 1225-1231.

Tuceryan, M., & Jain, A. (1998). Texture Analysis Handbook of Pattern Recognition and Computer Vision: 207-248. *World Scientific Publishing* Co.

Tuceryan, M., & Jain, A. K. (1990). Texture segmentation using Voronoi polygons. *IEEE Transactions on pattern analysis and machine intelligence, 12*(2), 211-216.

Tuceryan, M., & Jain, A. K. (1993). Texture analysis Handbook of pattern recognition and computer vision (pp. 235-276): World Scientific.

Turtinen, M. (2007). Learning and recognizing texture characteristics using local binary patterns. *University of Oulu, Finland.*

Turtinen, M., & Pietikäinen, M. (2006). Contextual analysis of textured scene images. *Paper presented at the BMVC,* (Vol. 2, No. 849)

Uzer, M. S., Yilmaz, N., & Inan, O. (2013). Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification. *The Scientific World Journal.*

Vailaya, A., Jain, A., & Zhang, H. J. (1998). On image classification: City images vs. landscapes. *Pattern recognition, 31*(12), 1921-1935.

Varma, M., & Garg, R. (2007). Locally invariant fractal features for statistical texture classification. *Paper presented at the 2007 IEEE 11th international conference on computer vision,* (pp. 1-8).

Wang, Q., Li, H., & Liu, J. (2008). Subset selection using rough set in wavelet packet based texture classification. *Paper presented at the Wavelet Analysis and Pattern Recognition,* (Vol. 2, pp. 662-666).

Wang, X., Han, T. X., & Yan, S. (2009). An HOG-LBP human detector with partial occlusion handling. *Paper presented at the Computer Vision, 2009 IEEE 12th International Conference on,* (pp. 32-39).

Wang, X., Yang, J., Teng, X., Xia, W., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern recognition letters, 28*(4), 459-471.

Weszka, J. S., Dyer, C. R., & Rosenfeld, A. (1976). A comparative study of texture measures for terrain classification. *IEEE Transactions on systems, man, and cybernetics*(4), 269-285.

Xie, X. (2008). A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA Electronic Letters on Computer Vision and Image Analysis,* 7(3), 1-22.

Xu, Y., Ji, H., & Fermüller, C. (2009). Viewpoint invariant texture description using fractal analysis. *International Journal of Computer Vision, 83*(1), 85-100.

Xu, Y., Yang, X., Ling, H., & Ji, H. (2010). A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid. *Computer Society Conference on Computer Vision and Pattern Recognition,* (pp. 161-168). IEEE.

Yao, Y., & Zhao, Y. (2008). Attribute reduction in decision-theoretic rough set models. *Information sciences, 178*(17), 3356-3373.

Zaid, M. S., Jadhav, R. J., & Deore, P. (2013). An efficient wavelet based approach for texture classification with feature analysis. *Paper presented at the Advance Computing Conference (IACC),* (pp. 1149-1153).

Zavaschi, T. H., Britto Jr, A. S., Oliveira, L. E., & Koerich, A. L. (2013). Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications, 40*(2), 646-655.

Zhang, J., Marszałek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision, 73*(2), 213-238.

Zhang, M., & Yao, J. (2004). A rough sets based approach to feature selection. *Paper presented at the Fuzzy Information,* (Vol. 1, pp. 434-439).

Zhang, Y., Wu, L., & Wang, S. (2011). Magnetic resonance brain image classification by an improved artificial bee colony algorithm. *Progress In Electromagnetics Research, 116*, 65-79.

Zhao, H., Sinha, A. P., & Ge, W. (2009). Effects of feature construction on classification performance: An empirical study in bank failure prediction. *Expert Systems with Applications, 36*(2), 2633-2644.

Zhou, F., Feng, J. F., & Shi, Q. Y. (2001). Texture feature based on local Fourier transform. *Paper presented at the Image Processing,* (Vol. 2, pp. 610-613).

Zhu, S.-C., Guo, C.-e., Wu, Y., & Wang, Y. (2002). What are textons?. *Paper presented at the European conference on computer vision,* (pp. 793-807).

Zupan, J. (1994). Introduction to artificial neural network (ANN) methods: what they are and how to use them. *Acta Chimica Slovenica, 41,* 327-327.

Appendices

Appendix 1

Group A of Confusion Matrices for UIUC database

Table A.1 Confusion matrix of LBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	31	2	0	0	0	0	0	0	0	0	0	3	0	2	0	0	0	1	0	0	0	1	0	0	0
2	0	22	0	0	0	0	0	2	0	0	0	8	0	6	0	0	0	0	0	0	0	0	2	0	0
3	1	0	33	1	1	0	0	0	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	22	1	9	0	0	1	0	0	0	4	1	2	0	0	0	0	0	0	0	0	0	0
5	0	0	2	1	32	1	3	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
6	0	0	1	3	0	22	0	0	5	0	0	0	4	1	2	0	1	0	0	0	0	0	0	0	1
7	0	0	0	2	2	0	33	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0
8	0	1	0	0	0	0	0	36	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	3	0	0	31	2	2	0	0	0	1	0	1	0	0	0	0	0	0	0	0
10	0	2	0	0	0	1	0	1	5	23	5	0	0	0	0	0	1	0	1	0	0	0	0	0	1
11	0	0	0	0	0	1	0	0	1	2	32	0	0	0	0	0	0	0	1	0	0	1	2	0	0
12	3	7	0	0	0	0	0	2	0	0	0	23	0	3	0	0	0	0	0	0	0	0	0	2	0
13	0	0	0	1	1	2	0	0	0	0	0	0	35	1	0	0	0	0	0	0	0	0	0	0	0
14	3	5	0	0	1	0	0	0	0	0	0	3	1	26	0	1	0	0	0	0	0	0	0	0	0
15	0	0	1	4	0	0	0	0	2	0	2	0	0	0	31	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
18	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	1	0	0	0	2	0
19	0	1	0	0	0	0	0	2	0	0	0	0	0	2	0	0	2	0	32	0	0	0	1	0	0
20	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	35	1	0	0	0	0
21	0	0	0	1	1	0	1	0	1	2	0	0	0	0	2	0	0	0	0	0	27	2	1	0	2
22	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	37	0	0	0
23	2	0	1	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	3	0	0	0	27	1	0
24	1	0	0	0	1	0	0	2	0	0	0	1	0	0	0	2	3	0	0	1	0	0	0	29	0
25	1	0	0	1	0	0	0	0	0	0	1	2	0	3	0	1	0	2	0	3	1	0	1	2	22

Table A.2 Confusion matrix of LZBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	38	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	32	0	0	0	0	0	4	0	0	0	0	0	1	0	0	0	0	2	0	0	0	0	0	0
3	1	0	35	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	32	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0
5	0	0	0	1	36	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	8	0	30	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	0	0	0	0	0	34	0	0	0	0	0	1	0	0	0	0	3	0	0	0	0	1	0
9	0	0	0	1	0	1	0	0	29	0	4	0	0	0	0	0	0	0	0	0	4	0	1	0	0
10	0	0	0	0	0	0	0	0	3	35	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	33	0	0	0	1	0	0	0	0	0	0	0	0	0	5
12	0	0	0	0	0	0	0	0	0	0	0	35	0	3	0	0	0	0	0	0	0	1	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	1	0	0	0	0	0	0	0	0	2	0	31	0	5	0	0	0	0	0	0	0	0	0
15	0	0	0	0	1	0	0	0	1	1	2	0	1	0	33	0	0	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	1	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
19	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	1	0	0	1	2	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	38	0	0	0	1	0
21	0	0	1	0	0	0	0	0	7	0	0	0	0	0	1	0	0	0	0	0	30	1	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	37	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	2	0	1	0	34	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	0	0	0	0	37	0
25	1	0	0	0	0	0	0	0	0	5	2	0	0	0	0	0	0	0	0	0	0	0	1	0	31

Table A.3 Confusion matrix of LMP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	31	0	0	0	0	0	0	0	0	0	0	1	0	3	0	1	0	0	0	1	0	2	0	0	1
2	1	26	0	0	0	0	0	2	0	0	0	4	0	3	0	0	0	1	3	0	0	0	0	0	0
3	1	0	31	0	2	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	1	1	1	0	0
4	0	0	0	26	0	12	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	1	31	1	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0
6	0	0	1	5	0	31	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
8	0	3	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	2	0	0	36	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
10	0	0	0	0	0	0	0	0	4	30	1	3	0	0	1	0	0	0	1	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	2	34	0	0	0	3	0	0	0	0	0	0	0	0	0	1
12	0	3	0	0	0	0	0	0	0	0	1	32	0	4	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	1	1	0	0	0	0	0	0	0	37	1	0	0	0	0	0	0	0	0	0	0	0
14	0	3	1	0	0	0	0	0	0	0	1	3	0	30	0	1	0	0	0	0	0	0	1	0	0
15	0	0	1	0	0	0	0	0	1	0	1	0	0	0	37	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	1	0	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
18	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	1	0
19	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0
20	0	0	0	0	0	0	1	0	0	2	0	0	0	0	1	0	0	0	0	34	0	1	0	0	1
21	0	0	0	1	0	1	0	0	3	1	1	0	0	0	2	0	0	0	0	0	30	0	0	0	1
22	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	0	0	36	0	0	0
23	1	1	2	0	0	0	0	0	0	1	0	1	0	0	0	0	2	2	3	0	0	0	27	0	0
24	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	2	0	0	1	1	0	0	0	30	3
25	1	0	0	0	0	0	2	0	0	8	1	0	0	1	0	0	1	2	2	0	0	1	0	1	20

Group B of Confusion Matrices for UMD database

Table B.1 Confusion matrix of LBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	38	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	36	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	35	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	1
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	39	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
10	1	1	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	1	0	0	0	37	0	0	0	0	1	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	2	0	0	0	1	1	0	35	0	0	0	1	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	2	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	3	0	0	0	0	1	0	0	0	0	36	0	0	0	0	0	0	0	0
18	0	0	0	3	0	1	0	1	0	0	0	0	0	0	0	0	0	28	0	0	0	0	0	7	0
19	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	3	0	0	0	0	0
20	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	1	0	0	0	0
21	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	38	0	0	0	0
22	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	37	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	38	0	0
24	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	0	0	2	0	0	0	0	0	35	0
25	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	37

Table B.2 Confusion matrix of LZBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	37	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	38	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3
4	0	0	0	34	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	2	1	0	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	1	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	2	0	0	0	36	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
9	0	0	0	0	1	0	0	0	36	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0
10	5	2	0	1	0	0	0	0	0	32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	1	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	1	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	1	0	36	3	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	33	0	0	0	0	0	4	1
19	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0
20	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	5	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	36	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	39	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	38	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
25	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38

Table B.3 Confusion matrix of LMP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	37	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	37	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
4	0	0	0	35	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	2	0	0	0	0	0
5	0	0	0	0	39	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	0	35	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0
9	0	0	0	0	1	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0	1	0	3	0
10	1	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	1	38	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	1	0	0	1	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	2	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	1	0	2	0	1	0	0	0	0	0	0	1	0	0	28	0	0	0	0	0	7	0
19	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	38	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	1	0	0	0	0
21	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	37	0	0	0	1
22	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	39	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	39	0	0
24	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	37	0
25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39

Group C of Confusion Matrices for KTHTIPS2b database

Table C.1 Confusion matrix of LBP

	1	2	3	4	5	6	7	8	9	10	11
1	98	0	0	0	1	0	1	0	0	0	0
2	1	81	0	2	0	7	1	0	4	0	4
3	0	2	72	3	2	5	6	1	3	0	6
4	3	0	1	90	3	3	0	0	0	0	0
5	0	1	7	1	78	1	0	6	0	5	1
6	0	8	0	3	2	86	1	0	0	0	0
7	1	4	3	0	0	0	84	0	3	0	5
8	0	0	1	0	4	3	0	88	0	1	3
9	0	3	1	0	0	0	0	0	96	0	0
10	0	0	0	0	3	0	1	0	0	96	0
11	1	0	4	1	2	2	1	2	1	0	86

Table C.2 Confusion matrix of LZBP

	1	2	3	4	5	6	7	8	9	10	11
1	91	0	1	0	0	0	2	0	0	3	3
2	0	65	0	4	0	16	8	0	6	0	1
3	2	2	42	4	3	5	22	2	10	3	5
4	0	4	1	90	0	1	1	0	0	0	3
5	1	2	1	2	57	4	0	14	1	7	11
6	0	4	2	6	5	68	4	1	0	0	10
7	1	6	10	1	0	4	73	0	5	0	0
8	0	0	0	1	17	1	0	76	0	0	5
9	0	7	4	0	0	0	4	0	84	1	0
10	0	0	1	0	4	0	4	0	3	88	0
11	3	2	1	1	6	2	0	14	0	0	71

Table C.3 Confusion matrix of LMP

	1	2	3	4	5	6	7	8	9	10	11
1	96	0	0	2	0	0	0	0	0	1	1
2	0	83	0	0	0	6	3	0	2	0	6
3	1	0	85	2	4	2	1	0	2	0	3
4	0	0	3	92	0	4	0	1	0	0	0
5	0	0	6	1	69	1	0	9	1	9	4
6	0	6	2	3	1	87	1	0	0	0	0
7	1	1	4	0	0	1	90	0	3	0	0
8	0	0	0	0	10	3	2	80	0	1	4
9	0	7	0	0	0	0	1	0	92	0	0
10	0	0	0	0	0	0	1	0	2	97	0
11	2	0	1	0	3	2	2	1	1	0	88

Group D of Confusion Matrices for Brodaze database

Table D.1 Confusion matrix of LBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
2	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	33	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	2	0	0	1	0	1	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
10	2	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
11	0	0	0	1	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	37	0	1	0	0	0	0	0	0	1	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	1	0	0
17	0	2	0	0	0	4	0	0	0	0	0	0	0	1	0	0	33	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0
22	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0
23	0	0	0	4	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	31	2	0
24	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	37	0
25	0	0	0	0	0	0	0	0	0	2	0	0	0	4	0	0	1	0	0	0	0	0	0	0	33

Table D.2 Confusion matrix of LZBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
2	0	39	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	1	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0
10	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	0	0	0	0	1	5	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	3	0	0	0	0	0	1	0	0	0	36	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	36	0	1	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0
23	0	0	0	3	0	0	0	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	33	1	0
24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	38	0
25	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	2	0	0	1	0	1	1	0	33

Table D.3 Confusion matrix of LMP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	38	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
10	1	0	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
11	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	2	0	0	0	0	0	0	0	0	0	35	0	0	2	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0
23	0	0	0	6	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	29	2	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	39	0
25	0	0	0	0	0	0	0	0	0	1	1	0	0	2	0	0	1	0	0	0	0	0	1	1	33

Appendix 2

Group A of Confusion Matrices for UIUC database

Table A.1 Confusion matrix of GF combined with LBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	32	1	0	0	0	0	0	0	0	0	0	1	1	4	0	0	0	1	0	0	0	0	0	0	0
2	0	33	0	0	0	0	0	1	0	0	1	2	0	2	0	0	0	0	1	0	0	0	0	0	0
3	1	0	35	0	0	0	0	0	0	0	0	1	2	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	30	0	4	0	0	0	0	0	0	3	0	1	0	0	0	0	0	2	0	0	0	0
5	0	0	3	0	33	0	2	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0
6	0	0	0	4	1	28	0	0	3	1	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	5	0	33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
8	0	1	0	0	0	0	0	37	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	33	4	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	2	34	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	2	0	35	0	0	0	0	0	1	0	1	0	0	0	1	0	0
12	0	0	0	0	0	0	0	0	0	0	1	38	0	1	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	2	2	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0
14	2	2	1	0	0	0	0	0	0	0	2	1	1	31	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	1	0	0	3	0	0	0	1	0	35	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	1	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	2	0	0	0	0	0	0
19	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3	0	35	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	2	33	2	0	0	0	0
21	0	0	0	1	0	0	0	0	1	0	0	0	0	0	3	0	0	0	0	0	33	0	0	1	1
22	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	1	0	0
23	3	0	1	0	0	1	0	0	1	0	2	0	0	0	0	0	0	1	1	2	0	0	27	1	0
24	1	0	1	0	1	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	1	32	0
25	0	1	0	0	0	2	0	0	0	1	0	0	1	2	0	0	0	1	0	0	2	4	2	0	24

Table A.2 Confusion matrix of GF combined with LZBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	35	2	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1
2	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
3	0	0	36	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	34	0	5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
5	0	0	1	1	35	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
6	0	0	0	3	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	37	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
9	0	0	0	1	0	1	0	0	37	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
10	0	0	0	0	0	0	0	0	1	38	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	38	0	0	0	2	0	0	0	0	0	0	0	0	0	0
12	1	1	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	1	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	1	2	0	36	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	0	1	1	0	1	0	36	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
19	0	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	34	0	0	0	0	2	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	38	0	0	0	0	0
21	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	36	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0
23	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	2	0	1	0	34	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	38	0
25	0	0	0	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	0	1	0	0	0	0	34

Table A.3 Confusion matrix of GF combined with LMP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	32	0	0	0	0	0	0	0	0	0	1	0	0	3	0	2	0	0	0	0	0	1	0	0	1
2	0	37	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0
3	1	0	32	0	3	0	0	0	1	0	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	33	0	5	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	2	34	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
6	0	0	0	3	2	28	0	0	4	0	0	0	2	0	0	0	0	0	0	0	1	0	0	0	0
7	0	0	0	0	1	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	39	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	1	0	1	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	39	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	1	0	0
13	0	0	2	0	3	0	0	0	0	0	0	0	35	0	0	0	0	0	0	0	0	0	0	0	0
14	2	1	0	0	0	0	0	0	0	0	3	0	0	34	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	0	0	2	0	0	0	37	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	38	0	1	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	39	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0	36	0	0	0	0	0	0
20	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	2	0	35	0	0	0	0	0
21	0	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	0	2	33	0	0	1	0
22	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	37	0	0	0
23	1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	2	0	1	0	33	0	0
24	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	37	0
25	0	2	1	0	0	0	0	0	0	4	2	0	0	1	0	0	2	0	0	0	0	3	0	1	24

Group B of Confusion Matrices for UMD database

Table B.1 Confusion matrix of GF combined with LBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	36	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	1	0
9	0	0	0	0	0	1	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	1	0	0	0	38	0	0	0	0	1	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	1	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0
23	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	38	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Table B.2 Confusion matrix of GF combined with LZBP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
4	0	0	0	37	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	37	1	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	37	0	0	0	0
22	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	39	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Table B.3 Confusion matrix of GF combined with LMP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
1	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	37	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
3	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	39	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	39	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	1	0	0	0	36	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0
9	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2	0	0	0	0	0	0	0	0	38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	38	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	39	1	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36	0	0	0	0	0	1	3
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0
20	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0	0	0	0	0
21	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	36	0	0	0	1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	39	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	39	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	38	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	40

Group C of Confusion Matrices for KTHTIPS2b dataset

Table C.1 Confusion matrix of GF combined with LBP

	1	2	3	4	5	6	7	8	9	10	11
1	100	0	0	0	0	0	0	0	0	0	0
2	1	89	0	0	0	5	0	0	3	0	2
3	0	2	88	0	6	0	2	0	2	0	0
4	0	0	0	97	0	1	0	0	2	0	0
5	0	0	2	2	82	4	1	3	0	5	1
6	0	2	0	1	0	95	1	0	0	0	1
7	1	3	0	4	1	3	88	0	0	0	0
8	0	1	0	0	6	3	0	85	0	0	5
9	0	2	0	0	0	0	0	0	98	0	0
10	0	0	0	1	5	0	0	0	0	94	0
11	2	3	3	0	0	1	2	0	0	0	89

Table C.2 Confusion matrix of GF combined with LZBP

	1	2	3	4	5	6	7	8	9	10	11
1	97	0	0	1	0	0	1	0	0	1	0
2	0	89	0	2	0	2	3	0	4	0	0
3	1	0	92	0	1	1	1	1	1	2	0
4	0	0	0	96	0	2	0	0	2	0	0
5	0	1	3	0	76	1	1	7	0	5	6
6	0	0	0	6	2	82	3	1	0	0	6
7	0	1	3	0	0	2	90	0	4	0	0
8	0	0	1	0	8	0	0	87	0	0	4
9	0	5	2	0	0	0	5	0	88	0	0
10	0	0	0	1	2	0	2	0	1	94	0
11	2	0	1	0	1	3	0	5	0	0	88

Table C.3 Confusion matrix of GF combined with LMP

	1	2	3	4	5	6	7	8	9	10	11
1	98	0	0	0	1	0	0	0	0	0	1
2	0	93	0	1	0	2	2	0	2	0	0
3	0	1	94	2	1	0	2	0	0	0	0
4	0	1	0	96	0	3	0	0	0	0	0
5	0	0	1	2	80	2	1	2	1	6	5
6	0	2	0	3	0	93	1	0	0	0	1
7	1	0	1	1	0	2	94	0	1	0	0
8	0	1	0	0	4	3	0	88	0	1	3
9	0	5	0	0	1	0	0	0	94	0	0
10	0	0	0	0	2	0	0	0	1	97	0
11	2	0	0	0	1	1	1	1	0	0	94