TEXTURE CLASSIFICATION USING WAVELET TRANSFORM AND SUPPORT VECTOR MACHINES

By

Alvin Yeo Zong Yun

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION SYSTEMS (HONS)

INFORMATION SYSTEMS ENGINEERING

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2011

DECLARATION OF ORIGINALITY

I declare that this report entitled "**TEXTURE CLASSIFICATION USING WAVELET TRANSFORM AND SUPPORT VECTOR MACHINES**" is my own work except as cited as in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :

Name : Alvin Yeo Zong Yun

Date : 8th April 2011

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Mr. Tou Jing Yi, for his patient guidance throughout the whole progress of this thesis. This is the first time I was involved in computer vision and machine learning field, and I have no much idea on these fields at the beginning. Luckily, I am able to learn the knowledge and finish this thesis under his patient and thorough guidance.

I would also like to say a million thanks to a very special person in my life, Shu Jun, for her patience, unconditional love and support, and for standing by my side during the hard time.

A special thanks to my best friend in my university life too, Mr. Ong Hen Ching, for his precious suggestions, guidance, and various helps throughout the course. He is a very nice, helpful and knowledgeable person. It is definitely my pleasure having such a good friend in my life. A million thanks for him too.

Finally, I must say thanks to my parents, my family, and my friends for their love, support and continuous encouragement throughout my life.

ABSTRACT

Texture has been widely used in human life since it provides useful information that appeared on the surface of every object. The most common use of texture is to help everyone to identify different objects in daily life. Texture is also often involved in many important real life applications such as biomedical image processing, remote sensing, wood species recognition, etc. Such situation has encouraged extensive researches to be conducted on texture, such as texture analysis and texture classification under the computer vision field. This thesis has conducted a research study on texture classification, by using wavelet transform and support vector machines (SVM) as the main feature extraction and classification method respectively. The objective of this work is to discover the main factors that will affect the performance of wavelet transform and SVM during a texture classification process. This thesis was started with designing and developing a texture classification system by using wavelet transform and SVM. Next, a series of experiments were carried out in order to test the different main factors, such as wavelet families, numbers of decomposition level, and kernel functions, on the developed texture classification system. The experimental dataset used in this thesis is 32 types of textures from Brodatz album, with a total number of 2048 texture samples. The final experimental results show that the developed texture classification system is able to achieve the highest classification rate at 87%, whereas the fastest time for processing a whole texture dataset (2048 texture samples in 64x64 pixels) is 30 seconds. Such results have proved that the developed texture classification system by using wavelet transform and SVM is potential and worth to be implemented in real life applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GLCM	Grey Level Co-occurrence Matrix
k-NN	k-Nearest Neighbor
SVM	Support Vector Machines
LBP	Local Binary Patterns
STFT	Short-time Fourier Transform
DWFT	Discrete Wavelet Frame Transform
CWT	Continuous Wavelet Transform
DWT	Discrete Wavelet Transform
RBF	Radial-basis Function

CHAPTER 1.0

INTRODUCTION

1.1 Background

Texture can be defined as the properties appeared on the surface or structure of an object (Sonka *et al.*, 2008), such as a person's fingerprint, repeated patterns on a clothes, structure of a textile, etc. In other words, texture describes the characteristics such as shapes, patterns, and coarseness that appear on the surface on an object. These characteristics found on textures are very important and useful in human daily life. In fact, everyone is using textures to recognize and distinguish different objects around the world (Wager, 1999). Such important feature has caused that textures are inevitable to be extensively involved in many real life applications, such as remote sensing, document processing, biomedical image processing, etc. (Tuceryan and Jain, 1998).

However, texture is complicated as it has no precise definition due to its large and wide variability (Sonka *et al.*, 2008). The total number of variety in textures is almost equivalent to the total number of variety in objects that can be found around the world. Besides, structure of a texture can be simple and also complicated. All of these negative issues have significantly increased the difficulties in applications dealing with textures, especially involved in huge amount of data.

Texture classification techniques have been developed in order to minimize the issues mentioned above. The main objective of texture classification is to classify or label various kinds of textures into the correct texture groups or classes according to the features found on each texture. This includes two major processes, i.e. feature extraction and texture classification. Over the last few decades, many texture classification approaches have been developed, ranging from conventional statistical perspective to modern computational perspective. The common feature extraction examples are grey level co-occurrence matrix (GLCM) and wavelet transform, with classifiers such as k-nearest neighbor (k-NN) and support vector machines (SVM).

There is no researcher yet has successfully discovered a perfect texture classification approach due to the limitations on texture images, especially scale issue. This statement can be proven by studies from some researchers that they concluded there is no texture feature is consistently superior for all texture images (Li *et al.*, 2003). Obviously, it is insufficient from research aspect, although most of the current texture classification approaches are able to give a promising performance in classifying some texture classes under certain constraints or conditions.

This thesis conducted a research study on texture classification approaches with an objective to discover a promising texture classification solution. In this work, the texture classification approach was tested with 32 types of texture from Brodatz Album (Brodatz, 1996). The main feature extraction method and classifier used in experiments were wavelet transform and SVM. The experimental results of the developed texture classification approach were evaluated and a conclusion was made lastly.

1.2 Problem Statement

There are two main processes in texture classification approaches, which are feature extraction and classification. This implies that these two processes are the main criteria in determining the success of a texture classification approach. In other words, discovering a good feature extraction method and classifier is the main problem in constructing a promising texture classification approach.

1.3 Motivation

The following are the main motivations to study texture classification approaches:

1. Discovering promising texture classification approach

Texture classification is generally a difficult problem due to wide diversity and complexity of textures (Zhou, 2006). Many texture classification approaches have been proposed over the last three decades, ranging from conventional statistical perspective to modern computational perspective. Some studies have even tried to combine both statistical and computational methods in order to obtain better results, such as combination of wavelet transform and GLCM. However, there is no approach yet that achieves high level performance in all situations (Busch and Boles, 2002). The problem of texture classification still remains a challenging area of research in computer vision.

2. Fulfilling the market demand

The concept of texture classification has been widely implemented in other computer or machine vision applications such as automated visual inspection and content based image retrieval (Hiremath and Shivashankar, 2006). Besides, texture classification has been extensively implemented in many real life applications or industries, such as rock texture recognition, wood species identification (Tou, 2009), biomedical image processing, and remote sensing. Texture classification applications are shifting rapidly from traditional areas such as remote sensing to new areas like content based image retrieval (Picard *et al.*, 1993). This implies that texture classification is essential in contributing and assisting many real life texture-based applications. This arouses in many researchers a sense of meaningful study to ease those complicated work related to textures.

1.4 Objective

The following are the main objectives to conduct this thesis:

1. To develop a complete texture classification approach composed of wavelet transform and SVM.

2. To implement the developed texture classification approach on a set of Brodatz textures.

3. To discover and test the main factors that will affect the performance on wavelet transform and support vector machines.

4. To make a conclusion for the performance of the developed texture classification system.

1.5 Scope

The following are the main focuses of this thesis on studying texture classification approaches:

1. The main feature extraction method and classifier developed in the texture classification approach are wavelet transform and SVM.

2. The developed texture classification approach will be tested with a set of grey scale and stationary texture images from Brodatz album.

CHAPTER 2.0

LITERATURE REVIEW

2.1 Texture

Texture refers to properties that represent the surface or structure of an object (Sonka *et al.*, 2008). Some examples of the surface properties that can be easily observed are shapes, patterns, coarseness, etc. In fact, the surface of any visible object is textured at certain scale (Zhou, 2006). Therefore, textures can be easily found on every object, but it is hard to be defined due to its wide variability and complexity (Tuceryan *et al.*, 1998) (Zhou, 2006) (Sonka *et al.*, 2008).

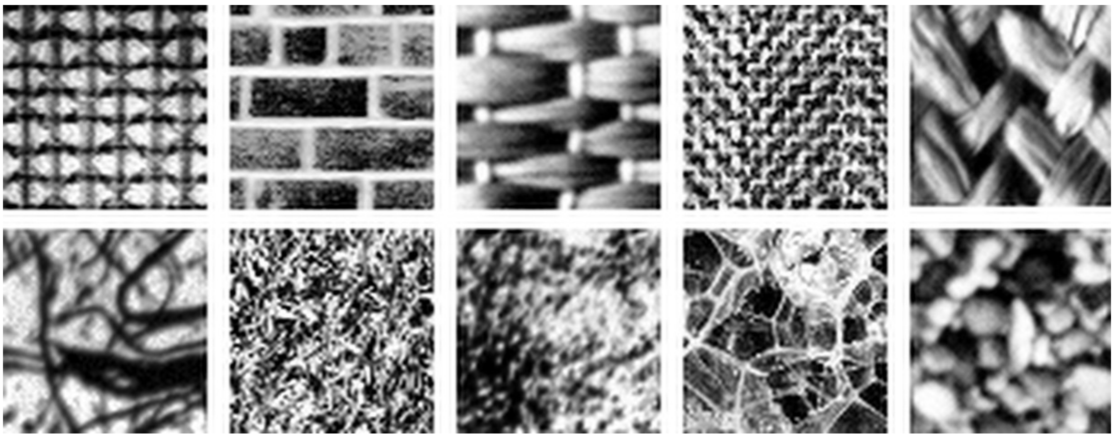


Figure 2.1: Examples of textures from Brodatz Album (Brodatz, 1996).

The structure of textures can be simple but also complicated. Figure 2.1 shows some of the texture examples which are extracted from Brodatz Album. From the figure, it is quite obvious that the first row of textures tend to have a regular and repeating pattern throughout the image. On the other hand, the second row of textures demonstrate rather irregular or random pattern throughout the image.

Texture has been playing a significant role in assisting us in many real life applications. The most basic and general use of the texture is to help every one of us to recognize and distinguish different objects around the world (Wager, 1999). The concept is, different surface properties carried by textures give us an idea how an object's surface physically is and hence help us to identify the object (Tou, 2009). This actually has implied that texture is somehow related to human visual perception system and it can be an important clue in computer vision.

Apart from that, texture has been widely involved in many real life applications such as remote sensing (Ruiz *et al.*, 2004), biomedical image processing (Tuceryan *et al.*, 1998), content based image retrieval (Manjunath and Ma, 1996), and rock and wood species classification (Tou, 2009). Most of the time, these applications are dealing with large amount of complicated texture images, which are very difficult to be processed manually. As a result, texture analysis has become one of the interesting topics under computer vision that aims to minimize the texture problems encountered in these applications.

The structure of a texture can be further explained by its texture primitives (Sonka *et al.,* 2008). Texture primitives can be considered as the fundamental elements that build up a texture. For example, a stack of bricks are the texture primitives that build up a wall texture. Texture primitives contain important information that describes a texture, which also known as features. Haralick (1979) has proposed that such features can be found in texture primitives through their pixel intensity (tone) and spatial relationship (structure).

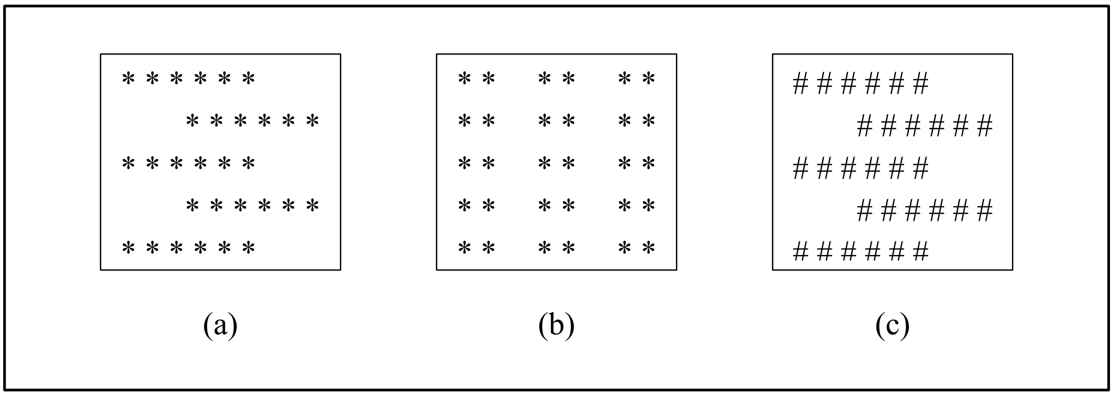


Figure 2.2: Examples of tone and structure in three textures (Sonka *et al.,* 2008).

Figure 2.2 demonstrates a simple idea of how a texture can be described by the tonal and structure properties of its texture primitives. In simple words, tone can be considered as texture primitive's type and structure is the spatial relationship that defines how texture primitives are organized in a texture. From the figure 2.2, it is obvious that textures in (a) and (b) share the same tonal property while textures in (a) and (c) have the same type of structure. However, three of them represent different textures. This implies that both tonal and structure properties of texture primitives are fundamental and significant factors in describing a texture.

From another point of view, Petrou and Sevilla (2006) has defined that texture is the variation of data at scale smaller than the scales of interest. This definition implies that texture is somehow related to the problem of scale. This is absolutely true as one of the main properties of texture is scale-dependent.

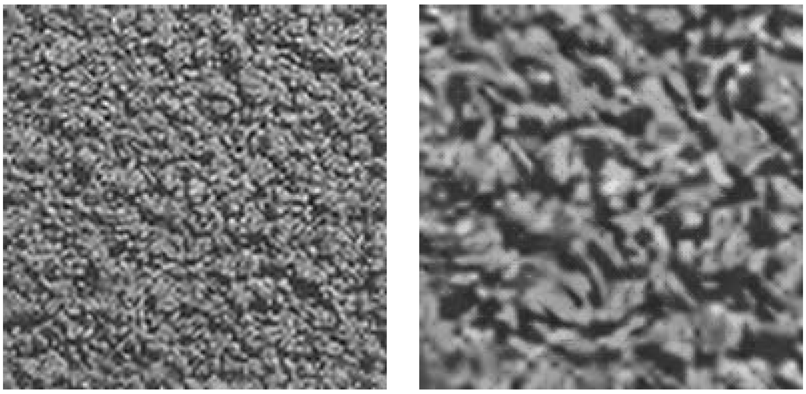


Figure 2.3: Observe a texture from different scales may yield different texture images.

Figure 2.3 illustrates the problem of scales clearly. This can be explained by the concept of scale that used in map. For example, a non-detailed global view at high scale and a detailed area view at low scale are totally different although both views are focusing on the same area.

Similar to the scale problem, imaging geometry or viewing angle is also a critical factor that can affect the texture descriptions (Petrou *et al.*, 2006). Since texture is an image which is composed of two-dimensional pixel matrix, values changed on the pixel matrix that caused by different imaging geometry or viewing angles, may greatly change the texture descriptions too. Figure 2.4 and 2.5 illustrates some of the examples of image geometry and viewing angle problems in textures.

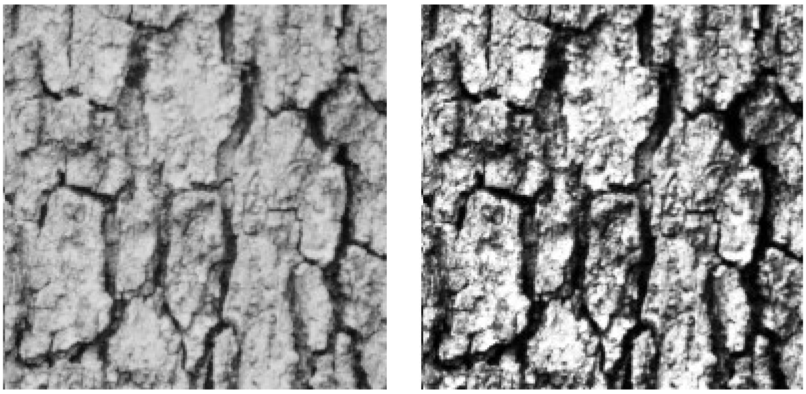


Figure 2.4: Observe a texture from different illumination directions may yield different texture descriptions.

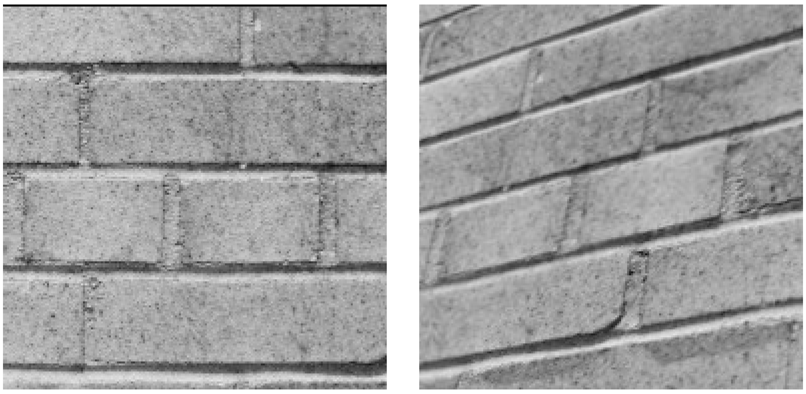


Figure 2.5: Observe a texture from different viewing angel may yield different textures descriptions.

2.2 Texture Classification Approaches

One of the main goals of texture analysis in computer vision is texture classification. Texture classification is a process to classify or label different kinds of textures into correct texture groups based on the feature vectors found on them. Therefore, texture classification includes two fundamental processes, which are feature extraction and classification.

However, texture classification is a difficult problem due to the wide variability and complexity of textures. Over the last few decades, many researchers have been trying to describe textures from many perspectives of their nature. As a result, a large number of feature extraction methods have been proposed and developed, such as GLCM, local binary patterns (LBP), Gabor filters, and wavelet transform.

From the review study conducted by Tou *et al.* (2009) on the recent trends in texture classification, texture classification approaches can be categorized up to five main groups in general, namely (1) structural; (2) statistical; (3) signal processing; (4) model-based stochastic, and; (5) morphology-based methods. The review study also shows that statistical and signal processing approaches are the popular choices in recent years. The main reason most probably is because of their promising performance in terms of time and accuracy.

2.2.1 Structural Approaches

Haralick (1979) has defined a texture as an "organized area phenomenon", which can be decomposed into a set of texture primitives that are organized in a particular spatial structure indicating certain underlying placement rules. This definition is also known as structural approach (Zhou, 2006). Figure 2.5 shows an example of structured textures with a number of bricks that serve as a set of texture primitives in a wall texture.

Structural approaches are based on the theory of formal language, which use syntactic grammars to discriminate objects belonging to different groups, based on the arrangement of their structural features (Olszewski, 2001). This requires both texture primitive descriptions and rules of the texture primitive placement must be determined first in order to obtain a set of structural texture features (Sonka *et al.,* 2008). This has implied that structural approaches are difficult to be applied to unstructured or stochastic textures, which usually fail to produce structural features with their irregular or random texture primitives.

Two examples of the feature extraction methods used in structural approaches are shape chain grammars and graph grammars (Sonka *et al.,* 2008). However, structural approaches are not widely applied in texture classification since they are more suitable for structured textures, and unfortunately, most of the real world natural textures are not in structured form (Tou, 2009).

2.2.2 Statistical Approaches

Texture analysis has been once defined as a class of mathematical procedures and models which characterize the spatial variations within imagery as a means of extracting information (Srinivasan and Shobha, 2008). This definition is somehow close to the characteristic of statistical approaches, which are a set of mathematical procedures to analyze the spatial distribution of the grey values in a texture image (Tuceryan *et al.*, 1998) (Srinivasan *et al.*, 2008).

The general procedures carried out in most of the statistical approaches are (1) computing local features at each point in the texture image, and next; (2) deriving a set of statistics from the distributions of the local features (Ojala and Pietikainen, 2004). In this way, texture features are able to be computed from the obtained statistics, such as contrast, correlation, entropy, homogeneity, etc. (Gonzalez and Woods, 2010). Table 2.1 shows some of the examples of texture features computed from statistical approaches and their descriptions.

Feature	Description
Contrast	A measure of intensity contrast between a pixel and its neighbor over the entire image.
Correlation	A measure of how correlated a pixel is to its neighbor over the entire image.
Uniformity	A measure of uniformity in the range [0,1].

Table 2.1: Examples of texture features computed from statistical approaches and their descriptions (Gonzalez *et al.*, 2010).

Statistical approaches can be computed in several dimensions or orders. The level of dimension or order is depending on the number of pixels that define the local features (Ojala *et al.*, 2004). First-order statistic calculates features from individual pixel by ignoring the spatial relationship between the pixel and its neighbor, whereas second-order statistic calculates features from two or more pixels and determines the spatial relationship between them at the same time. Second-order statistic is definitely more suitable for texture analysis since spatial relationship is an important factor in describing a texture (Gonzalez and Woods, 2010).

Some of the examples of statistical approaches are GLCM, LBP, covariance matrices, grey level aura matrices, co-occurrence histograms, etc. (Tou, 2009).

2.2.3 Signal Processing Approaches

Signal processing approaches analyze a texture image in frequency domain, rather than the spatial domain which is used in statistical approaches (Tou *et al.*, 2009). The psychophysical studies have proved that our human visual system analyzes an image by decomposing it into various frequency and orientation components (Tuceryan *et al.*, 1998). Therefore, transforming the spatial domain of a texture image into frequency domain always give better and visible information in texture classification (Sonka *et al.*, 2008).

Fourier transform is one of the earlier signal processing approaches used in texture classification. Initially, Fourier transform is just a mathematical approach used in transforming a time-domain signal into frequency-domain signal. The objective of Fourier transform is quite obvious, which is to retrieve the information existed only in the frequency domain (Polikar, 1996). Later, researchers found that the concept can be also applied in images processing, to transform an image from the spatial domain into frequency domain. However, Fourier transform only describes the global frequency content of an image, without any reference to localization in the spatial domain, has caused a poor performance in texture classification (Ojala *et al.*, 2004).

In order to overcome the drawback of missing information of localization in the spatial domain that occurs in Fourier transform, short-time Fourier transform (STFT) has been introduced. The main concept of STFT is to apply a window function during the analysis process of Fourier transform. This window function will cover a small portion of the signal or image until the frequency content inside the window function is almost stationary. In this way, both frequency and localization information are able to be retrieved by analyzing the window function throughout the whole signal or image (Polikar, 1996).

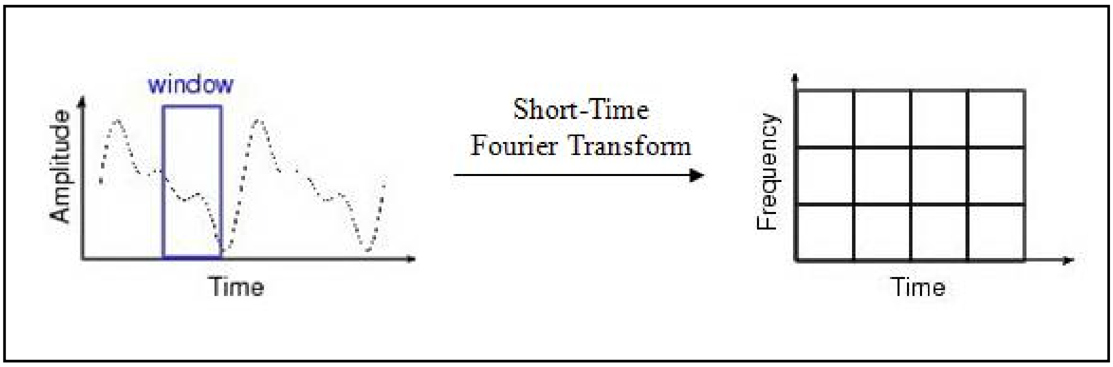


Figure 2.6: Short-time Fourier transform (Mathworks, 2010).

Figure 2.6 illustrates a rough idea on how STFT works. However, the major drawback of STFT is once a window function is selected, the time-frequency resolution is fixed over the entire time-frequency plane (Tuceryan *et al.*, 1998). This implies that STFT is not a perfect approach on texture classification yet since texture is highly scale-dependent. In order to overcome this drawback, Mallat (1989) has proposed a new multiresolution analysis approach, named wavelet transform. Wavelet transform will be further discussed in the following section and chapter 3.

2.3 Wavelet Transform

Wavelet transform is one of the current popular feature extraction methods used in texture classification. It was first proposed by Mallat back in 1989 (Ruiz *et al.*, 2004). The main motivation of developing the wavelet transform is to overcome the constant resolution problem in STFT (Barford *et al.*, 1992). Based on this motivation, Mallat has successfully created wavelet transform as a powerful multiresolution analysis technique, which is able to be applied in many fields, such as signal processing, computer vision, chemistry, neurophysiology, economy, etc. (Polikar, 1999).

In fact, the concept of the wavelet transform is quite similar to STFT. Both of them apply a window function during the analysis process. However, the window function used in the wavelet transform is a wavelet function, which features the capability of multiresolution analysis. In contrast with the conventional window function, the main wavelet function, also known as mother wavelet, is able to be scaled and translated over the entire data, to capture both frequency and spatial localization information in the frequency domain (Graps, 1995) (Polikar, 1996).

From the statement above, it is obvious that there are two important properties in wavelet transform, which are scaling and translation. Scaling is a property to create different lengths of wavelet functions by compressing or dilating the mother wavelet, in order to capture different frequency resolutions in the entire data. However, translation is a property to translate or move every generated wavelet function over the entire data, in order to capture the spatial localization information. These two important properties have actually explained the capability of multiresolution analysis in wavelet transform.

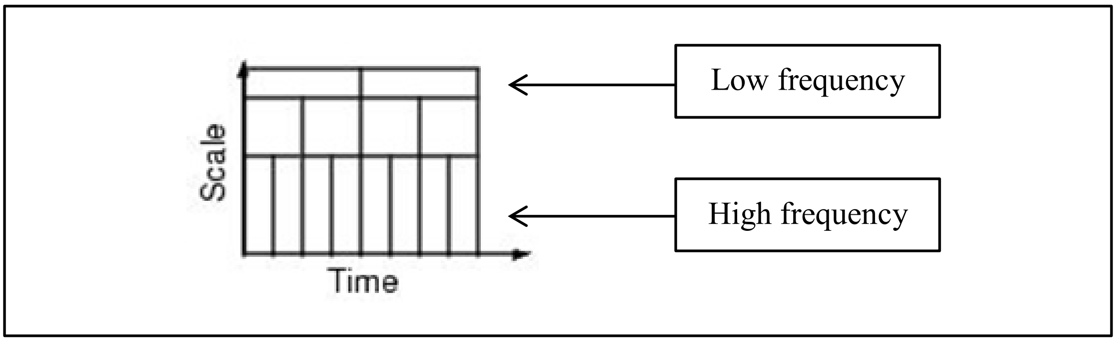


Figure 2.7: Scale-time plane of wavelet transform (Mathworks, 2010).

Figure 2.7 illustrates the common output result of wavelet transform in a two dimensional scale-time plane. It is very obvious that the result of wavelet transform is mapped into different resolutions, compared to the consistent resolution output result of STFT which is shown in figure 2.6. From figure 2.7, it is also quite easy to notice that the low frequency components always have larger scales and longer translation period, whereas the high frequency components always have smaller scales and shorter translation period. In fact, such rules are almost tally with the real life phenomenon (Polikar, 1996).

In texture classification, Hiremath and Shivashankar (2006) have pointed out some important characteristics of the wavelet transform based approaches: (1) The wavelet transform is able to decorrelate the data and achieve the same goal as the linear transformation used in Brodatz album. (2) The wavelet transform provides orientation sensitive information which is essential in texture analysis. (3) The computational complexity is significantly reduced by considering the wavelet decomposition.

As a result, a lot of different wavelet transform based approaches have been proposed and implemented in texture classification problems in the recent years. Some of the examples include discrete wavelet transform (Sebe and Lew, 2000), discrete wavelet frame transform (Li *et al.*, 2003), wavelet frame packets (Liu and Lee, 1996), multiple wavelet analysis (Busch *et al.*, 2002), etc. Other details of wavelet transform will be further discussed in the chapter 3.

2.4 Support Vector Machines (SVM)

SVM is one of the popular classifiers used in solving binary classification problems in recent years (Tou *et al.*, 2009). Similar to other binary classification approaches, the main goal of SVM is to construct a model representation which is able to classify two classes of data by using a separating hyperplane or decision surface. Figure 2.8 shows a simple SVM model which is used in a binary classification problem.

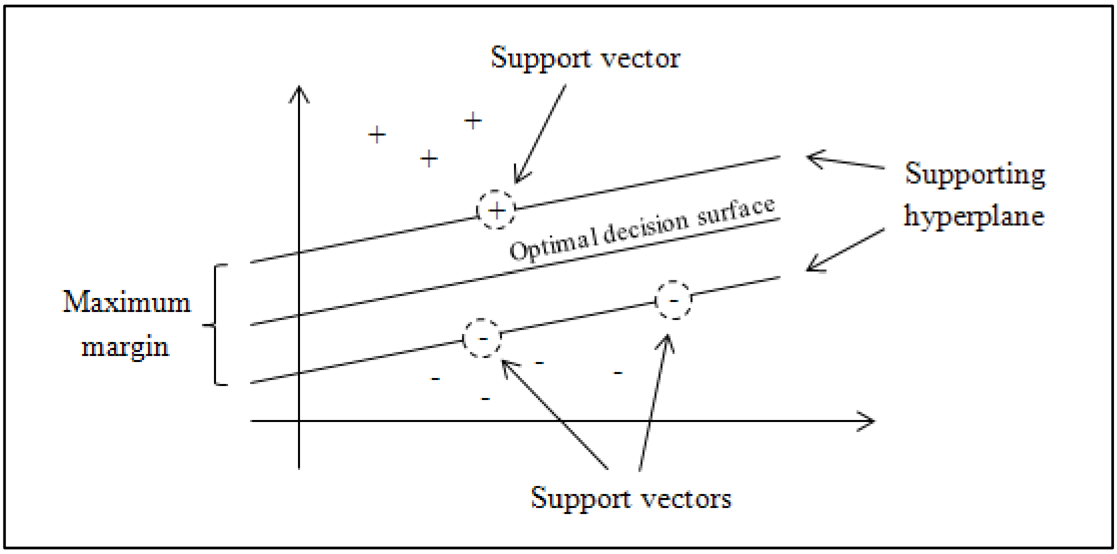


Figure 2.8: Example of a simple SVM model (Hamel, 2009).

Computing the optimal decision surface can be considered as the biggest problem in binary classification approaches. In SVM, the optimal decision surface is computed by determining the maximum margin between the two supporting hyperplanes. This implies that the support vectors or supporting hyperplanes are the main factors that influence the optimal decision surface in SVM. This also explains how SVM is named.

In order to construct the optimal decision surface, support vectors have to be determined at first. Defining the support vectors in a SVM model is also equivalent to defining the supporting hyperplanes, since the supporting hyperplanes are located on the top of support vectors. This can be done by using one of the characteristics of SVM, i.e. duality. In the conventional primal approach, a classifier constructs a decision surface in terms of a normal vector $\bar{w}$ and an offset term b. On the other hand, SVM adopts the dual approach to construct a decision surface by using the training data coefficients $\bar{\alpha}$ and the offset term b, as shown in figure 2.9 (Hamel, 2009).

The main idea behind the dual approach is to search the data points which have the biggest influence to the optimal decision surface. Usually, such data points are the one that located on the class boundaries, which are support vectors.

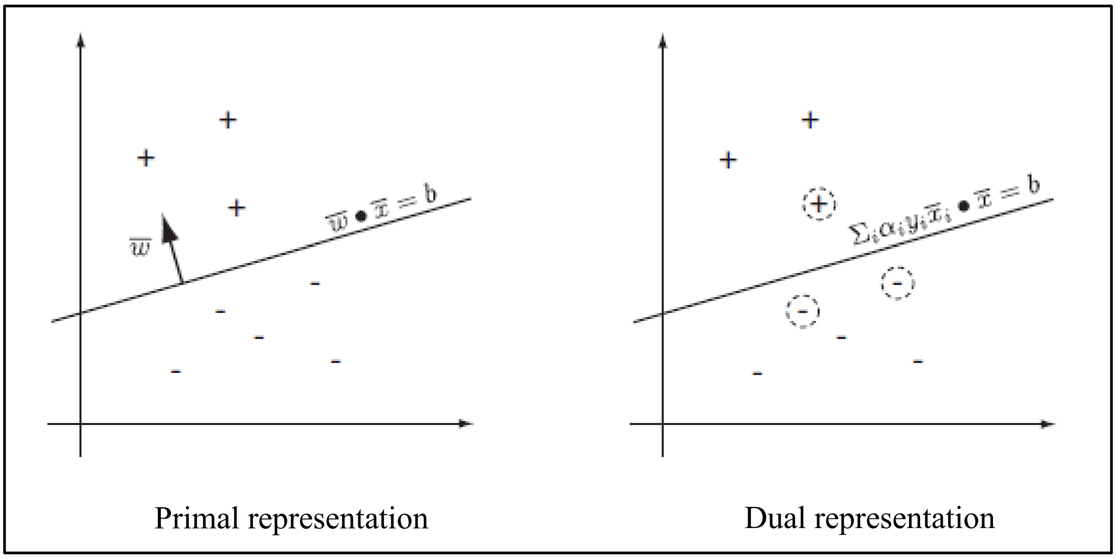


Figure 2.9: Decision surface representation (Hamel, 2009).

From figure 2.9, it is quite obvious that the data points close to the class boundaries have exerted the most constraints on the decision surface. Therefore, it will be more efficient to construct a decision surface by searching the support vectors in the dual approach, rather than blindly adjusting the normal vector $\overline{w}$ of the decision surface in the primal approach.

Similarly, the maximum margin between the two supporting hyperplanes can be computed by using dual approach. In SVM, the dual approach for the maximum-margin classifier optimization problem can be solved by applying Lagrangian optimization theory (Hamel, 2009). Computing the maximum margin is also one of the essential steps in constructing the optimal decision surface in SVM, since the optimal decision surface is located in the right middle of the two supporting hyperplanes, which is half of the maximum margin.

The dual view of maximum-margin classifiers has brought a significant advantage in extending the linear classifiers to non-linear classifiers (Hamel, 2009). The fundamental idea behind such extension is to transform the input space, where the data are not linearly separable, into a higher dimensional space, called feature space, where data are linearly separable. Such transformation process has been known as the kernel trick, while the functions associated within the transformation are called kernel functions. A kernel trick does not even require any extra computational cost, if the correct kernel function is picked.

Furthermore, SVM has introduced a soft-margin property which allows the underlying maximum-margin classifier to make mistakes on the training data set (Hamel, 2009). This is because most of the data used in the real life applications contain errors or noises. In order to ensure the performance of separating hyperplane is not affected by these error data, it is necessary to allow some error data to fall on the "wrong side" of the separating hyperplane (Noble, 2006). This can be done by introduced a slack variable. A maximum-margin classifier that incorporates slack variables is also known as a soft-margin classifier.

2.5 Summary

This chapter introduces textures and some of the factors that may affect them at the beginning. Later, some of the common texture classification approaches are briefly introduced, including structural, statistical, and signal processing approaches. Lastly, this chapter ends with the reviews about the feature extraction and classification method used in this project, which are wavelet transform and SVM.

CHAPTER 3.0

SOFTWARE DESIGN FOR TEXTURE CLASSIFICATION

3.1 Introduction

This chapter describes the overall software architecture and design for the developed texture classification system in this project. The developed system includes two main processes, i.e. feature extraction and classification. The feature extraction method used in this project is discrete wavelet frame transform (DWFT), whereas the classification method used is SVM. Lastly, the developed system will be tested with 32 types of texture classes which are retrieved from Brodatz album.

3.2 Software Architecture

The developed texture classification system is composed of two major processes, i.e. feature extraction and classification. For the classification process, it is further decomposed into training and testing phases, which are the fundamental steps in a supervised learning system.

Figure 3.1 illustrates the overall process flow for the developed texture classification system. First and foremost, the system starts with reading a texture image. Next, the feature extraction method will analyze the input image and extract the texture features. Once the system obtains enough amounts of texture features, the classifier will start either the training or testing phase. In training phase, the classifier will be trained by labeling the input image to a specific texture class. On the other hand, the classifier will test and classify the input image into the correct texture class in testing phase, based on the available trained data. In this project, a set of collected dataset will be divided into two portions for each training and testing phase. Lastly, the output results produced by the classifier will be evaluated.

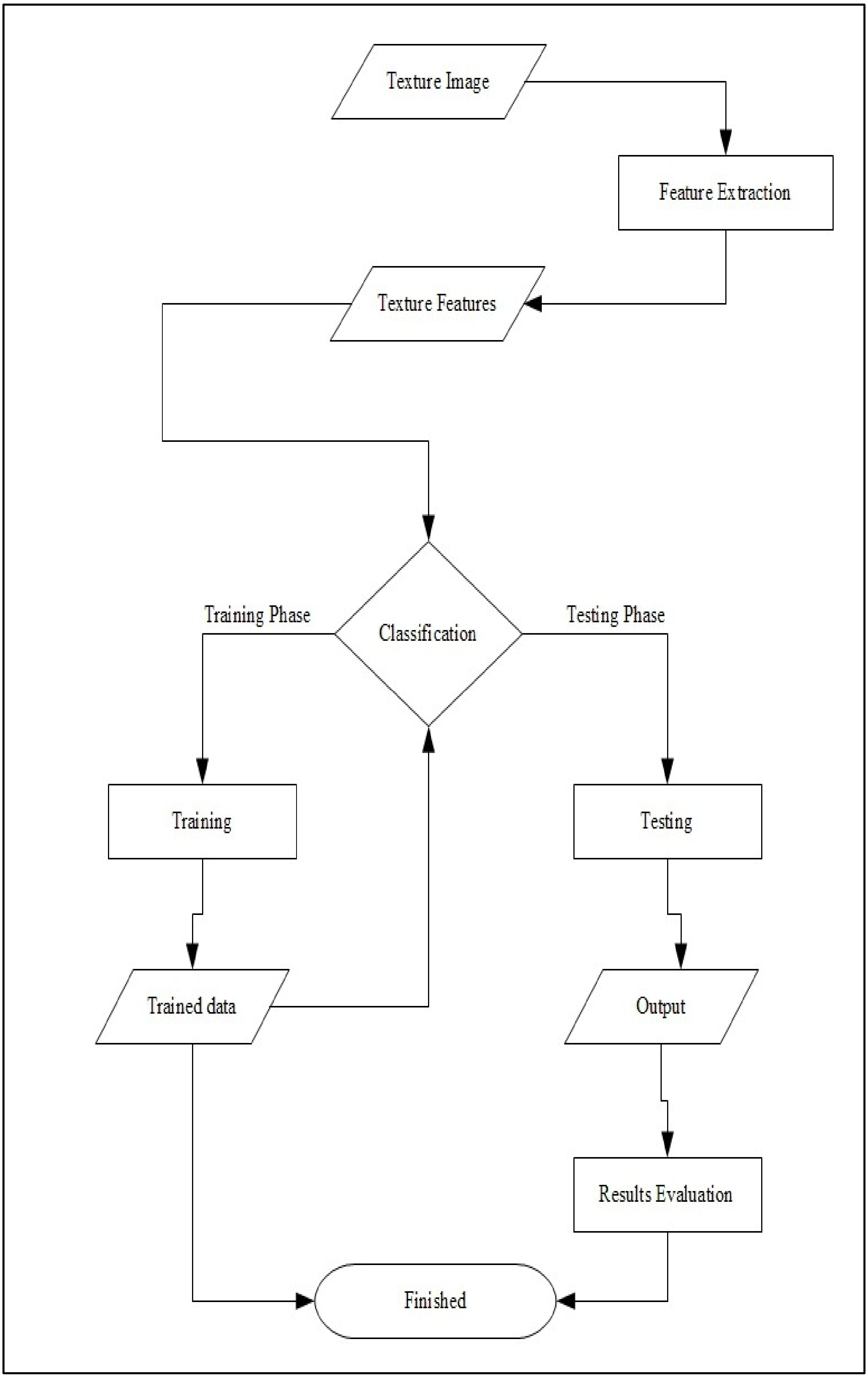


Figure 3.1: Overall process flow for developed texture classification system.

3.3 Feature Extraction

Feature extraction is one of the major processes in a texture classification system. The main purpose of the feature extraction process is to extract the useful information that is enough to describe or represent a texture from the input image. Feature extraction is also a prerequisite step for the classification process, as the classifier always requires texture features to perform its tasks. Theoretically, texture images which belong to a same class should provide similar texture features. However, this is not necessary true because some natural textures have a very weak spatial structure over its texture image (Tou, 2009).

3.3.1 Wavelet Transform

Texture features can be extracted from the input image by using wavelet transform. Wavelet transform is able to analyze an image by decomposing it into a series of subband images. These subband images can be considered as one form of wavelet coefficients, where texture features are computed.

There are a lot of wavelet transform methods have been proposed over the last two decades. Any function can be considered as a wavelet function, $\Psi(t)$ as long as it satisfies the following conditions (Polikar, 1999) (Amgaa, 2003):

1. A wavelet function's square integral is finite. This defines a wavelet must have a finite (small) energy.

$$\int_{-\infty}^{+\infty} |\Psi(t)|^2 \, dt < \infty \tag{3.1}$$

2. The integral of wavelet function, $\Psi(t)$ equals to zero. This defines that the function must be oscillatory (a wave).

$$\int_{-\infty}^{+\infty} \Psi(t) \, dt = 0 \tag{3.2}$$

3.3.1.1 Continuous Wavelet Transform (CWT)

Wavelet transform is represented by Equation (3.3) in general. It is also the equation for the CWT of a signal function in the time domain,$x(t)$, where a and b represent the scale and translation parameters respectively, Ψ represents the mother wavelet function, while $*$ represents the convolution operator (Polikar, 1999).

$$W(a,b) = \frac{1}{\sqrt{a}} \int x(t)\, \Psi * \left(\frac{t-b}{a}\right) dt,\ \ a > 0, b\ \in \Re \tag{3.3}$$

Equation (3.3) can be further interpreted as an inner product of (1) the signal function, $x(t)$, and; (2) the scaled and translated versions of the mother wavelet function, Ψ (Polikar, 1999). In fact, the scale and translation property of the mother wavelet has explained the multiresolution analysis in the wavelet transform.

$$W(a,b) = \int x(t) * \Psi_{(a,b)}(t) dt \tag{3.4}$$

$$\Psi_{(a,b)}(t) = \frac{1}{\sqrt{a}}\, \Psi \left(\frac{t-b}{a}\right),\ \ a > 0, b\ \in \Re \tag{3.5}$$

CWT, which calculates wavelet coefficients at every possible scale and position, has generated a lot of unnecessary information and cost expensive computing power and time. In order to make wavelet transform to become more practical and efficient in real life applications, Mallat has proposed the discrete wavelet transform (DWT), which only selects a subset of scales and positions during the analysis process.

3.3.1.2 Discrete Wavelet Transform (DWT)

DWT can be performed by iteratively filtering a signal or image through the low-pass and high-pass filters, and subsequently downsampling the filtered data by two (Polikar, 1999). This process will decompose the input image into a series of subband images. Figure 3.2 illustrates an example of DWT, where h and g represent the low-pass and high-pass filter respectively, while the symbol with a down arrow inside a circle represents the downsampling operation.

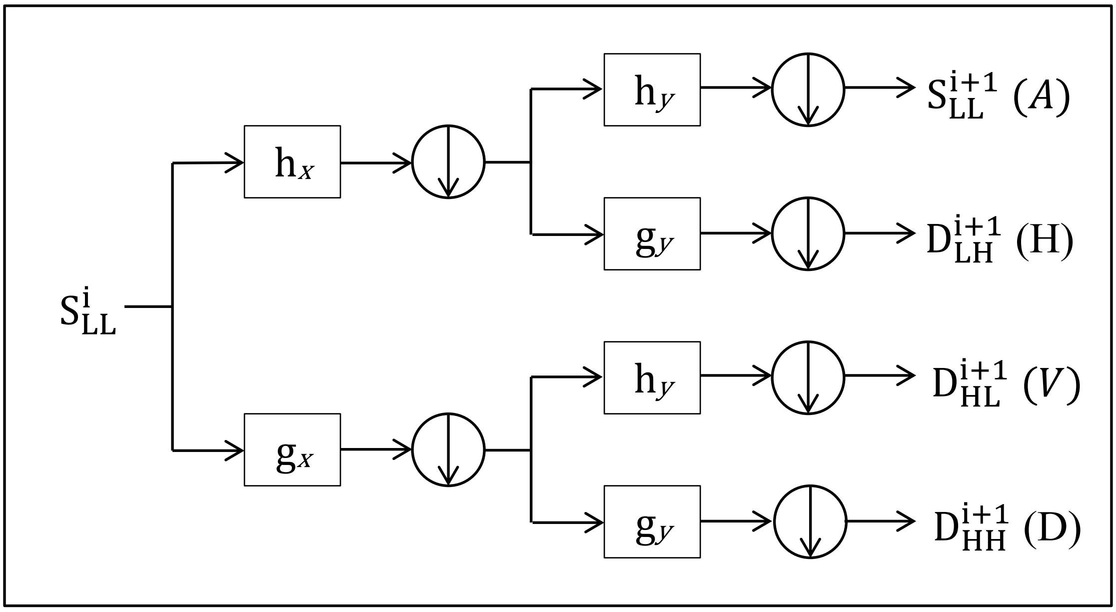


Figure 3.2: Discrete wavelet transform (Li *et al.*, 2003).

From figure 3.2, an image *S* at resolution level *i* was decomposed into four subband images after going through one stage of decomposition process. The four subband images consist of one approximation image and three detail images. The approximation image is actually the low-frequency components of the original image *S*, whereas the detail images are the high-frequency components of the original image *S* in different orientations, i.e. vertical, horizontal, and diagonal. The output of a decomposed image using wavelet transform is shown in figure 3.3.

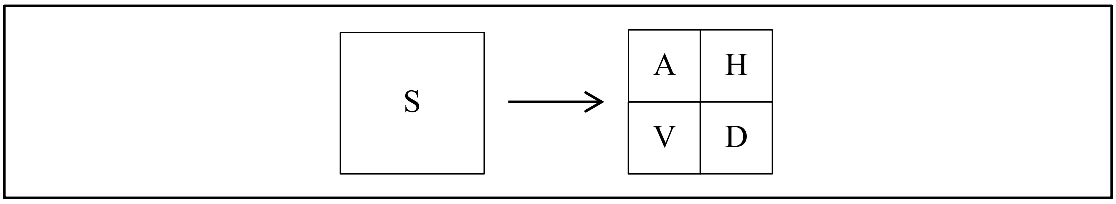


Figure 3.3: Subband images for one level of image decomposition using DWT.

Every detail image contains the information of specific scale and orientation. This means that the spatial information is also retained within the subband images

(Hiremath and Shivashankar, 2006). Therefore, the detail images are suitable to be used for deriving a set of texture features in the input image. On the other hand, the approximation image can be used for higher levels of decomposition for the input image.

The subband images obtained through the low-pass and high-pass filters can be computed by using Equation (3.6):

$$A = [\, h_x * \left(h_y * S\right)_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$H = [\, h_x * \left(g_y * S\right)_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$V = [\, g_x * \left(h_y * S\right)_{\downarrow 2,1}]_{\downarrow 1,2}$$
$$D = [\, g_x * \left(g_y * S\right)_{\downarrow 2,1}]_{\downarrow 1,2} \tag{3.6}$$

where h and g represent the low-pass and high-pass filter respectively, S represents the original input image, while ↓ represents the downsampling operator. In the decomposition process, the low-pass and high-pass filters are only responsible for filtering the high or low frequency components in the input image, but not removing the filtered components or samples from the image. In other words, it doubles the original size of the input image. Therefore, downsampling is an operation to remove those useless and filtered samples from the subband images, as shown in figure 3.4.

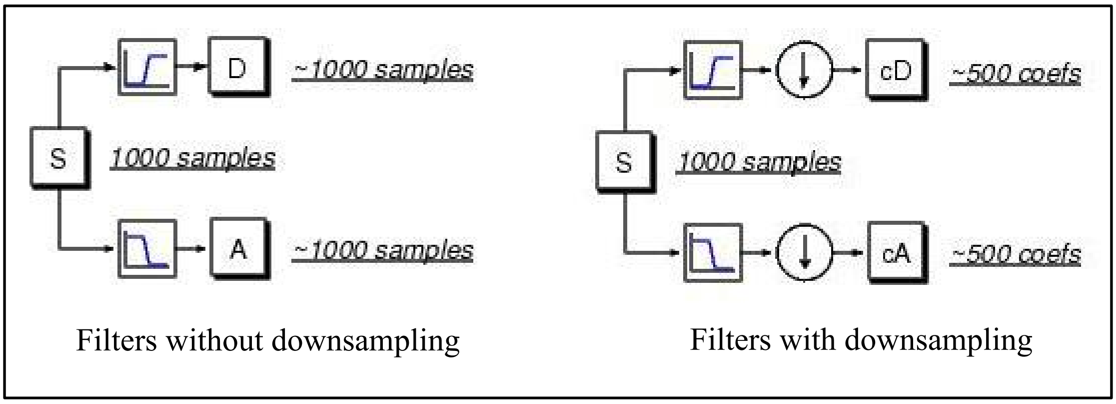


Figure 3.4: The concept of downsampling operation (Mathworks, 2010).

Downsampling operation has helped to reduce the useless and redundant samples in the decomposition process. However, removing these samples will cause a translation-variant property for the decomposition results. This approach is not suitable for texture classification due to texture features are translation-invariant (Liu *et al.*, 1996).

3.3.1.3 Discrete Wavelet Frame Transform (DWFT)

In order to overcome this problem, some researchers have suggested applying the DWFT method in texture classification (Liu *et al.*, 1996) (Li *et al.*, 2003). DWFT is almost exactly same to the traditional DWT method. The only difference between them is DWFT does not implement the downsampling operations, and hence create a translation-invariant property for the decomposition results, as shown in figure 3.5.

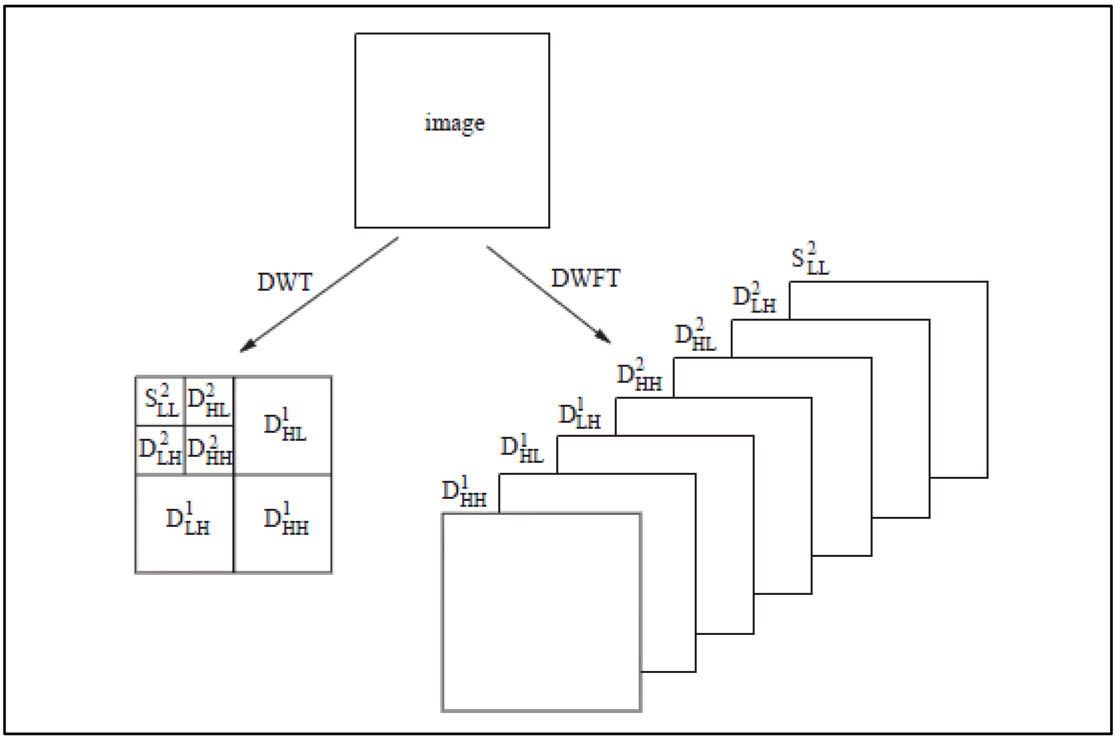


Figure 3.5: Difference between DWT and DWFT (Li *et al.*, 2003).

3.3.2 Wavelet Energy

After decomposing the input image into a series of subband images by using DWFT, the last part of the feature extraction process will be calculating the texture features from all details images. Wavelet energy has been proven as a popular and powerful feature representation in texture classification (Van de Wouwer *et al.*, 1999). The wavelet energy, E for each detail image can be computed by using the following equation, where $(M \times N)$ denotes the size of the detail image, I:

$$E_i = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} I_i^2 (x, y) \tag{3.7}$$

The wavelet energy reflects the distribution of energy along the frequency axis over the scale and orientation of a detail image. Since most of the relevant texture information has been removed by iteratively low-pass filtering, the energy of an approximation image is usually not considered as a texture feature (Van de Wouwer *et al.*, 1999).

3.4 Classification

Classification is another main process in texture classification. In this project, supervised learning system is implemented. Therefore, the classification process is further divided into training and testing phases. First of all, the developed classifier will be trained by mapping different types of textures into different classes in the training phase. Next, the classifier will be able to classify and label the testing samples into a correct texture class respectively in the testing phase, based on the information collected in the training phase. The developed classifier for the texture classification system in this project is SVM.

3.4.1 Support Vector Machines (SVM)

In this project, there are a total number of 32 classes of textures in the experiment data. This is a multiclass classification problem. However, SVM is a binary classification approach in general, as mentioned in chapter 2. Due to the various complexities, a direct solution for a multiclass classification problem by using a single SVM formulation is not recommended (Duan and Keerthi, 2005). On the other hand, a combination of several binary classifiers is suggested.

In this project, one-versus-one method has been applied in SVM in order to solve the multiclass texture classification problem. In this case, a series of binary classifiers will be created and trained under a concept of every texture class will be compared to all other 31 textures classes. As a result, a total number of 496 binary classifiers will be created and trained.

Class	1	2	3	4	5	6	7	8	…	32
1										
2										
3										
4										
5										
6										
7										
8										
:										
32										

Table 3.1: Simple idea of constructing binary classifiers using one-versus-one method.

Table 3.1 illustrates the simple idea on how 496 binary classifiers are constructed in this project. First, the dark boxes denote the unnecessary binary classifiers, whereas the white boxes denote the binary classifiers that will be created in this project. Obviously, it is impossible to create a binary classifier to compare a same class, such as "1st-class vs. 1st-class". Besides, it is also unnecessary to create the redundant binary classifiers, such as "1st-class vs. 2nd-class" and "2nd-class vs. 1st-class". By removing all these unnecessary binary classifiers, a total number of 496 binary classifiers will be obtained in a 32-class texture classification system at last.

There are two options available for classifying the texture samples in the testing phase while using the one-versus-one method. The first option is to use only 31 binary classifiers to classify a texture sample. This can be done by using the "1st-class vs. 2nd-class" binary classifier to classify a texture sample at first, and then the winner class will be compared with the subsequent classes' binary classifiers, until the last "32nd-class" binary classifier. On the other hand, the second option will use all the 496 binary classifiers to classify a texture sample. In this case, the most class number returned by the 496 binary classifiers will become the winner class for that texture sample.

In this project, the first option will be adopted in the developed texture classification system. This is because comparing a texture sample to 31 binary classifiers will significantly reduce the duration in testing phase, compared to 496 binary classifiers.

3.5 Summary

This chapter describes the software design and process flow of the developed texture classification system. The chapter starts with introducing the software architecture design for the developed system. Next section discusses the detail information which is necessary for constructing a feature extractor, i.e. wavelet transform. Similarly, the information for constructing the classifiers, SVM, is also included.

CHAPTER 4.0

EXPERIMENTS AND ANALYSIS

4.1 Introduction

This chapter describes the experimental materials and settings used in the experiments, such as dataset, tools, and parameter settings for the wavelet transform and SVM. The evaluation and analysis on the experimental results are also included.

4.2 Experimental Materials and Settings

This section describes the details of datasets which will be used in the experiments. Next, the development tool used for this project's system and various kinds of experiments will be briefly introduced. Lastly, the default settings used in the experiments will be stated out clearly.

4.2.1 Experimental Datasets

In this project, the dataset used in the experiments on the developed texture classification system is 32 types of texture images, which are retrieved from Brodatz album (Brodatz, 1996). Each type of texture image consists of 16 equal size samples (64x64), and each sample is further divided into 4 variants, which are the original image, the scaled image, the rotated image and lastly the image that is both scaled and rotated. Therefore, the whole dataset used in the experiments has a total number of 2048 texture images or samples. Later, the whole dataset is further divided into two portions equally during the experiments. The first portion is used as the training dataset, whereas another portion will be used as the testing dataset.

4.2.2 Experimental Tool

The main development tool used for this project's texture classification system and experiments is MATLAB (R2010a), with 2 toolboxes used, which are Wavelet Toolbox for wavelet transform, and Bioinformatics Toolbox for SVM.

4.2.3 Experimental Default Settings

Since different types of experiments have been carried out throughout the whole experiment process, there is no constant setting for most of the parameters. However, a default setting for all parameters is stated out clearly in table 4.1. This is because some parameters' value will be only modified in certain experiments, and usually they will be reset back to their default value in the other experiments.

Parameters	**Value**
Wavelet transform approach	Discrete wavelet frame transform (DWFT)
Wavelet family	Haar (haar)
Level of decomposition	3
SVM approach	One-versus-one for multiclass classification
SVM's kernel function	Radial-basis function (RBF) with width = 1
Number of training sample	1024
Number of testing sample	1024

Table 4.1: The default settings for all parameters used in the experiments. Parameter's value will not be changed unless it is specifically stated in experiments.

4.3 Experiments

Several experiments have been carried out in this project in order to test the different factors that will influence the performance of wavelet transform and SVM. Therefore, the overall experiment process can be divided into two main parts, i.e. wavelet transform and SVM. In wavelet transform experiment part, the tested factors include discrete wavelet transform approaches, wavelet families, and the number of decomposition levels in a wavelet transform. However, in SVM experiment part, the tested factors will be the kernel function types and their corresponding parameters. Lastly, a final experiment will be carried out by using the best parameter settings which are gathered from all the previous experiments, in order to observe the highest classification rate and the fastest time that can be achieved in the developed system.

4.3.1 Experiments on Wavelet Transform

The factors that will be tested on the wavelet transform include wavelet transform approaches, wavelet families, and the number of decomposition levels in a wavelet transform. All the experiments carried out will base on the default settings stated in 4.2.3, unless certain requirements are specifically stated in the experiments.

4.3.1.1 Wavelet Transform Approaches

As mentioned in chapter 3, DWFT has been suggested by some researchers as it outperforms the conventional DWT. In this experiment, the performance between DWFT and DWT will be tested in terms of accuracy and time. All the parameters' setting in this experiment are based on the default setting stated in 4.2.3, except the wavelet transform approach, it will be compared between DWFT and DWT. However, the experiments have been carried out for 3 times with 3 randomly selected training and testing datasets. The purpose is to increase the accuracy of the experimental results.

Experiments / Approach	Accuracy (%)			Time (second)		
	1	2	3	1	2	3
DWFT	84.3	82.9	82.1	32.4	33.9	35.1
DWT	65.7	67.3	65.7	42.2	41.9	41.7

Table 4.2: Experimental results for DWFT and DWT approaches.

The experimental results from table 4.2 have proved that DWFT outperforms DWT in terms of both accuracy and time. This is because the downsampling operations in DWT have produced the translation-variant property which is not suitable for texture classification. Besides, DWT has cost longer time to perform the downsampling operations. Take note that the calculated time includes both feature extraction and classification processes.

4.3.1.2 Wavelet Families

The objective of this experiment is to compare the performance of wavelet transform while using different types of wavelet basis. This experiment is tested with 4 randomly selected wavelet basis from different kinds of wavelet families, which are Haar family (haar), Daubechies family (db3), Coiflets family (coif1), and BiorSplines family (bior1.5). The wavelet basis' name is shown in the brackets. The way of executing this experiment is similar to the previous experiment, except the wavelet family substitutes the discrete wavelet transform approach as the main factor in the experiment.

Experiments / Wavelet basis	Accuracy (%)			Time (second)		
	1	2	3	1	2	3
haar	84.3	82.1	80.7	32.4	33.4	35.0
db3	79.2	78.6	80.8	35.5	34.6	37.1
coif1	82.3	80.0	79.2	35.1	35.7	33.4
bior1.5	81.7	82.5	80.8	39.8	36.8	36.9

Table 4.3: Experimental results for 4 wavelet basis from different wavelet families.

The experimental results show that there is no significant difference among the performance of 4 types of wavelet families. Each of them tends to have their own advantages while classifying different datasets. Take note that the calculated time includes both feature extraction and classification processes.

4.3.1.3 Number of Decomposition Levels

In fact, the number of decomposition levels performed in a wavelet transform represents the number of feature vectors collected for each texture sample. This is because one level of decomposition of a texture sample will produce 3 detail images, which are the texture features. Therefore, performing higher level of decomposition in a wavelet transform will produce larger amount of texture features for a texture sample, which may help to increase the classification's accuracy. All the parameters' setting in this experiment are based on the default setting stated in 4.2.3, except the level of composition, it will be tested from 1 to 10.

Experiments / Decomposition level	Accuracy (%)			Time (second)		
	1	**2**	**3**	**1**	**2**	**3**
1	70.0	66.3	69.4	30.7	30.0	31.8
2	82.3	78.2	82.1	30.1	30.5	30.0
3	84.3	80.8	83.4	32.4	33.1	32.6
4	83.3	81.2	84.5	37.3	38.4	36.6
5	83.5	80.2	84.4	41.9	42.1	42.1
6	84.0	79.5	84.4	44.6	47.6	46.2
7	82.0	77.7	82.1	49.7	52.9	52.6
8	82.0	78.3	80.4	54.2	57.6	56.5
9	80.5	76.6	78.6	61.4	63.2	61.0
10	78.3	77.0	77.2	64.6	67.2	68.1

Table 4.4: Experimental results for different levels of decomposition.

The experimental results from table 4.4 have proved that larger amount of texture features collected in feature extraction process does not guarantee to get a higher classification rate in the classification process. On the other hand, it will cost longer time to complete the feature extraction and classification processes. The results have showed that 3 to 6 levels of decomposition will be ideal in getting promising accuracy. In terms of both accuracy and time, 3 levels of decomposition seem to be the best candidate. Take note that the calculated time includes both feature extraction and classification processes.

4.3.2 Experiments on SVM

The factors that will be tested on the SVM include the types of kernel function and their corresponding parameters. All the experiments carried out will base on the default settings stated in 4.2.3, unless certain requirements are specifically stated in the experiments.

4.3.2.1 Kernel Function

This experiment will test the different types of kernel functions and their corresponding parameters, in order to observe the impacts caused on the classification rate and time. The tested kernel functions include linear kernel, radial-basis function (RBF) kernel, and polynomial kernel. There is no parameter associated with linear kernel. However, RBF kernel will be tested with a scaling factor, σ from 1 to 5, whereas polynomial kernel will be tested with a degree of from 1 to 5. Similarly, all the parameters' setting in this experiment are based on the default setting stated in 4.2.3, except the kernel function's type and its parameter.

Experiments / Kernel function	Accuracy (%)			Time (second)		
	1	**2**	**3**	**1**	**2**	**3**
Linear kernel	85.5	86.3	83.9	60.2	63.0	60.5
Polynomial kernel of degree 1	85.5	86.3	83.9	60.8	61.6	60.9
Polynomial kernel of degree 2	85.4	85.3	86.2	62.2	62.9	63.1
Polynomial kernel of degree 3	84.2	82.9	83.1	64.7	66.9	68.4
Polynomial kernel of degree 4	75.3	73.4	75.2	65.5	67.0	68.5
Polynomial kernel of degree 5	76.2	74.6	75.5	64.7	66.3	66.7
RBF kernel, σ = 1	83.4	81.6	84.3	33.6	35.6	32.4
RBF kernel, σ = 2	82.8	82.1	84.5	41.6	44.1	42.5
RBF kernel, σ = 3	81.4	81.3	82.1	43.5	46.7	45.6
RBF kernel, σ = 4	79.9	78.5	79.6	43.9	47.2	47.2
RBF kernel, σ = 5	76.7	74.3	74.9	42.5	45.2	45.2

Table 4.5: Experimental results for different types of kernel function.

The experimental results show that the linear kernel and polynomial kernel of degree 1 have the highest classification rate. However, they take almost double period of time compared to the RBF kernel with σ value 1. Under such situation, it may not be worth to sacrifice such significant amount of time, just for the sake of increasing 2 - 5% accuracy. Therefore, in terms of both accuracy and time, the RBF kernel with σ value 1 should be the best candidate for texture classification. Take note that the calculated time includes both feature extraction and classification processes.

4.3.3 Final Experiments

After going through a series of experiments on different factors which will influence the wavelet transform and SVM, a final experiment is carried out on the whole system by using the parameter settings gathered from all previous experiments. The purpose of this experiment is to observe the best performance that can be achieved by the developed texture classification system in terms of accuracy and time.

4.3.3.1 Experiments for Highest Accuracy

First, the developed system will be tested from the accuracy aspect, which means time factor is not considered. Table 4.6 states the parameter settings used in the experiment:

Parameters	**Value**
Wavelet transform approach	Discrete wavelet frame transform (DWFT)
Wavelet family	Haar (haar), BiorSplines (bior1.5)
Level of decomposition	4
SVM approach	One-versus-one for multiclass classification
SVM's kernel function	Linear kernel, polynomial kernel of degree 2
Number of training sample	1024
Number of testing sample	1024

Table 4.6: Parameter settings used to observe the highest classification rate.

In this experiment, both haar and bior1.5 wavelet basis are used since both of them obtained a promising results from previous experiments. This also applies to the selected SVM's kernel functions, which are linear kernel and polynomial kernel of degree 2. Polynomial kernel of degree 1 is not selected because it tends to give a same result as linear kernel, but take longer time in the whole process. Since there are more than one value for some parameters, several combinations of experiment are carried out.

Factors \ **Experiments**	**Accuracy (%)**					
	1	**2**	**3**	**4**	**5**	**Average**
bior1.5 + linear kernel	86.0	84.7	84.8	83.8	85.6	85.0
bior1.5 + polynomial of degree 2	86.4	83.5	85.4	84.4	85.9	85.1
haar + linear kernel	87.8	87.6	87.4	84.9	87.1	87.0
haar + polynomial of degree 2	85.1	85.4	85.9	86.6	85.4	85.7

Table 4.7: Experimental results for the highest accuracy that can be achieved in the developed system.

Factors \ **Experiments**	**Time (second)**					
	1	**2**	**3**	**4**	**5**	**Average**
bior1.5 + linear kernel	69.0	72.1	71.2	69.8	69.4	70.3
bior1.5 + polynomial of degree 2	75.0	73.2	69.7	72.0	71.4	73.5
haar + linear kernel	67.4	64.2	64.1	65.4	66.9	65.6
haar + polynomial of degree 2	70.7	66.5	66.2	67.6	68.4	67.9

Table 4.8: Experimental results in terms of time for the experiments run in table 4.7.

The experimental results show that the highest classification rate can be obtained by using a haar wavelet with a linear kernel type of SVM. The average classification rate for this approach is 87.0%, which is slightly higher than other approaches. Besides, the total time spent in this approach is also the lowest among the 4 approaches. This has made the approach become more appealing in texture classification since it produces ideal results in terms of both accuracy and time.

4.3.3.2 Experiments for Fastest Time

Time is also another significant factor that needed to be considered in texture classification. This is true when the texture classification system is implemented on a limited resources device, such as embedded device and mobile device. In this experiment, the time will be the main test factor rather than accuracy. Therefore, all parameter settings are selected according to the time aspect from previous experiments, as shown in the table 4.9:

Parameters	**Value**
Wavelet transform approach	Discrete wavelet frame transform (DWFT)
Wavelet family	Haar (haar)
Level of decomposition	2, 3
SVM approach	One-versus-one for multiclass classification
SVM's kernel function	RBF kernel, σ = 1
Number of training sample	1024
Number of testing sample	1024

Table 4.9: Parameter settings used to observe the fastest classification approach.

The options for parameter settings in this experiment are quite clear, except the number of decomposition level. In order to get more accurate experimental results, both 2 and 3 levels of decomposition are included in the experiment.

Experiments / **Factors**	**Time (second)**					
	1	**2**	**3**	**4**	**5**	**Average**
Level of decomposition = 2	32.3	30.0	31.6	31.9	32.0	31.6
Level of decomposition = 3	31.7	34.0	34.6	35.3	35.6	34.2

Table 4.10: Experimental results for the fastest classification time that can be achieved in the developed system.

Factors \ Experiments	Accuracy (%)					
	1	**2**	**3**	**4**	**5**	**Average**
Level of decomposition = 2	80.0	81.3	81.0	81.5	80.2	80.8
Level of decomposition = 3	81.9	82.1	82.5	84.1	81.3	82.6

Table 4.11: Experimental results in terms of accuracy for the experiments run in table 4.10.

From the experimental results, it is very obvious that the number of decomposition level in a wavelet transform is the main factor that affecting the time. The total time calculated in this experiment includes the time spent in feature extraction, training, and testing processes for whole dataset. The fastest time that can be achieved in the system is 30 seconds. In other words, it only takes 0.015 second in processing a texture sample. Therefore, it seems to be no significant difference between 2 and 3 levels of decomposition. Under such situation, a haar wavelet with 3 levels of decomposition will be suggested to be applied with RBF kernel with scaling factor 1, in order to get a fast classification time but not to sacrifice too much accuracy at the same time.

4.4 Analysis and Findings

After going through a series of experiment on the developed texture classification system, it can be concluded that the application of wavelet transform together with SVM in texture classification is able to produce a promising classification rate in terms of accuracy and time. Although 87% for the highest accuracy cannot be considered as a very good result in texture classification, the ability of processing 2048 texture samples within 40 seconds has made the system worth to be applied in the real life applications.

Some investigations have been conducted after the experiments, in order to find out the behind reasons that affect the accuracy, other than the factors that have been discussed above. The results show that some classes of texture images are very difficult to be classified into the correct classes, such as D5 and rice paper. Both of them can only get around 14 correct classified samples over a total number of 32 samples respectively, regardless of what factors have been used in the experimental approaches. Both of them share a similarity, that is, their texture patterns are stochastic. In other words, they have no constant texture primitives over the whole image. This may conclude that wavelet transform is not able to perform well in extracting a stochastic texture's features. Figure 4.1 shows some of the samples from the D5 and rice paper classes, in original form.

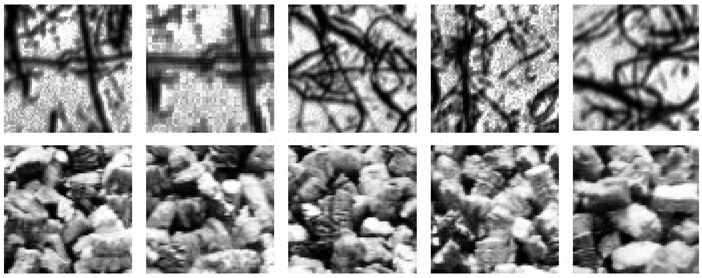


Figure 4.1: Example of texture samples from D5 and rice paper classes.

Considering the texture feature used in this project is just a simple wavelet energy, the classification results may be improved by using other more promising texture features, such as histogram and co-occurrence matrix. Calculating the relevant information between pixels should be better than just measuring the distribution of energy in a texture image.

4.5 Summary

This chapter includes all the information related to experiments conducted in this project. First, the experimental materials, tool, and settings are briefly described. Next, the detail descriptions for a series of experiments on the wavelet transform, SVM, and developed system are stated, including their corresponding experimental results. Lastly, analysis and findings for the obtained experimental results are briefly summarized.

CHAPTER 5.0

CONCLUSION AND FUTURE WORK

5.1 Findings of Research

This thesis has conducted a research study on texture classification, by using wavelet transform and SVM as the feature extraction and classification method. The final experimental results have proved that such texture classification approach is worth to be implemented in real life applications.

The research has been started with studying the properties located in a texture. The study shows that the main problem currently faced in the texture classification is no effective and efficient method to describe a texture, due to its wide variability and complexity. In order to overcome this issue, many researchers have tried to analyze some of the important properties that describe a texture, such as texture primitive, scale, tonal and structure properties. Later, these properties have become the fundamental foundation in discovering a feature extraction method.

A feature extraction method was developed based on the idea of how human capture a texture's descriptions. Researchers have tried to incorporate many solutions in their feature extraction methods in order to overcome the problems limited by texture properties. Wavelet transform is such a feature extraction method with an objective of solving scale-dependent issue in texture classification. By using the DWFT, a texture image is able to be analyzed at multiresolution by decomposing the texture image into detail images, which represent the texture's features.

A series of experiments have been carried out to test such factors that affect the performance of a wavelet transform in describing a texture. For example, an experimental result has showed the downsampling operations in DWT will significantly reduce the classification rate, because translation-variant property will destroy the scale information in a texture image.

Besides, wavelets such as Haar, Daubechies, Coiflets, and BiorSplines wavelets have been proved effective and efficient in describing a texture. The compactly supported characteristic of these wavelets has made them capable to analyze a texture image at multiresolution and hence collect detail texture's descriptions.

Apart from that, experimental results have proved that wavelet transform is able to describe a texture successfully with a small number of texture features, between 9 to 12 features for each texture. This indirectly helps to reduce the time required in a classification process, since not many features are computed.

Classification method is also an essential process in texture classification. SVM can be considered as a modern classification approach which features a lot of benefits, such as kernel trick and soft-margin classifiers. SVM has been proved by many researchers as a very powerful classification approach especially in binary classification problems.

However, texture classification is usually a multiclass classification problem all the time. The developed texture classification system in this thesis is able to prove the combination of several binary SVM classifiers can solve the multiclass classification problem in texture classification. The process can be completed by applying some kernel functions. Experiments on testing kernel functions also are conducted. Results show that the selection on different kernel functions can cause a significant impact in terms of accuracy and time during a texture classification process. Therefore, cautions have to be taken in picking correct kernel function in order to enhance the performance of a classifier.

Although a lot of improvement have been tried to be put into the developed texture classification system, the classification rate of around 80% is still not convincing and satisfied, as many real life applications cannot tolerate such high error rate. However, such problem is able to be overcome by executing the future work stated in the following section.

5.2 Future Work

A lot of improvement for the developed texture classification system in this thesis still can be achieved by doing some future works:

- Texture feature used in the developed texture classification system is wavelet energy. Since wavelet energy only measures the distribution of energy in a detail image, its performance may be not as good as features used in statistical approaches, such as histogram and co-occurrence matrix.

- It is possible to create an own customized wavelet basis to suit the requirements in texture classification. However, this is definitely not an easy task as it requires a strong base of knowledge in many areas, such as mathematics and physics. Since the current wavelets are still not able to analyze well on all types of textures, a new wavelet basis may be required.

REFERENCE

Amgaa, T. (2003). *Wavelet-based Analysis for Object Separation from Laser Altimetry Data*. Master Thesis. International Institute for Geo-information Science and Earth Observation.

Barford, L. A., Fazzio, R. S., Smith D. R. (1992). *An Introduction to Wavelets.* Instruments and Photonics Laboratory, Hewlett-Packard.

Brodatz, P. (1996). *Textures: A Photographic Album for Artists and Designers*. Dover Publications, Inc., Mineola, NY.

Busch, A. & Boles, W. W. (2002). Texture Classification Using Multiple Wavelet Analysis. *DICTA2002: Digital Image Computing Techniques and Applications*.

Duan, K. B. & Keerthi, S. S. (2005). Which is the Best Multiclass SVM Method? An Empirical Study. *Proceedings of the Sixth International Workshop on Multiple Classifier Systems*, pp. 278-285.

Gonzalez, R. C. & Woods, R. E. (2010). Texture. In *Digital Image Processing* (3rd ed.), pp. 849-861. USA: Pearson.

Graps, A. (1995). An Introduction to Wavelets. *IEEE Computational Science & Engineering*, 2(2), pp. 50-61.

Hamel, L. (2009). *Knowledge Discovery with Support Vector Machines*. USA: Wiley.

Haykin, S. (2009). *Neural Networks and Machine Learning* (3rd ed.). USA: Pearson.

Haralick, R. M. (1979). Statistical and Structural Approaches to Texture. *Proceeding of The IEEE*, 67(5), pp. 786-804.

Hiremath, P. S., & Shivashankar, S. (2006). Wavelet Based Features for Texture Classification. *GVIP Journal*, 6(3), pp. 55-58.

Li, S. T., Kwok, J. T., Zhu, H. L. & Wang, Y. N. (2003) Texture Classification Using the Support Vector Machines. *Pattern Recognition 36*, pp. 2883-2893.

Liu, J. F. & Lee, J. C. M. (1996). An Efficient and Effective Texture Classification Approach Using a New Notion in Wavelet Theory. *Proceedings of the 13th International Conference on Pattern Recognition (ICPR'96)*. Vol. 2, pp. 820 - 824.

Mallat, S. G. (1989). A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), pp. 674-693.

Mathworks (2010). Wavelet Toolbox. In *Matlab R2010b Documentation*. Retrieved April 6, 2011, from http://www.mathworks.com/help/toolbox/wavelet

Noble, S. W. (2006). What is A Support Vector Machine? *Nature Biotechnology*, 24(12), pp. 1565-1567.

Ojala, T. & Pietikainan, M. (2004). Texture Classification. *Machine Vision and Media Processing Unit, University of Oulu, Finland*. Retrieved April 5, 2010, from http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OJALA1/texclas.htm

Olszewski, R. T. (2009). *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. Doctor Thesis. Carnegie Mellon University.

Petrou, M. & Sevilla, P. G. (2006). *Image Processing Dealing with Texture*. West Sussex, England: Wiley.

Picard, R. W., Kabir, T. & Liu, F. (1993). Real-time Recognition with the Entire Brodatz Texture Database. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 638-639.

Polikar, R. (1996). The Engineer's Ultimate Guide to Wavelet Analysis: The Wavelet Tutorial. *Iowa State University of Science and Technology*. Retrieved August 16, 2010, from http://users.rowan.edu/~polikar/WAVELETS/WTtutorial.html

Polikar, R. (1999). The Story of Wavelets. *Physics and Modern Topics in Mechanical and Electrical Engineering*. pp. 192-197.

Ruiz, L. A. & Recio, J. A. (2004). Texture Feature Extraction for Classification of Remote Sensing Data using Wavelet Decomposition: a Comparative Study. *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXV, pp. 1109-1114.

Sebe, N. & Lew, M. S. (2000). Wavelet Based Texture Classification. *International Conference on Pattern Recognition*, pp. 959-962.

Srinivasan, G. N. & & Shobha, G. (2008). Statistical Texture Analysis. *Proceeding of World Academy of Science, Engineering and Technology*, pp. 1264-1269.

Sonka, M., Hlavac, V., & Boyle, R. (2008). *Image Processing, Analysis, and Machine Vision* (3rd ed.). Toronto: Thomson.

Tou, J.Y. (2009). *Texture Classification on Wood Images for Species Recognition*. Master Thesis. Universiti Tunku Abdul Rahman.

Tou, J. Y., Tay, Y. H., & Lau, P. Y. (2009). Recent Trends in Texture Classification: A Review. *Proceedings Symposium on Progress,* in *Information and Communication Technology 2009 (SPICT 2009), Kuala Lumpur*, pp. 63-68.

Tuceryan, M. & Jain, A. K. (1998). Texture Analysis. In *The Handbook of Pattern Recognition and Computer Vision* (2nd ed.), pp. 207-248, Singapore: World Scientific Publishing.

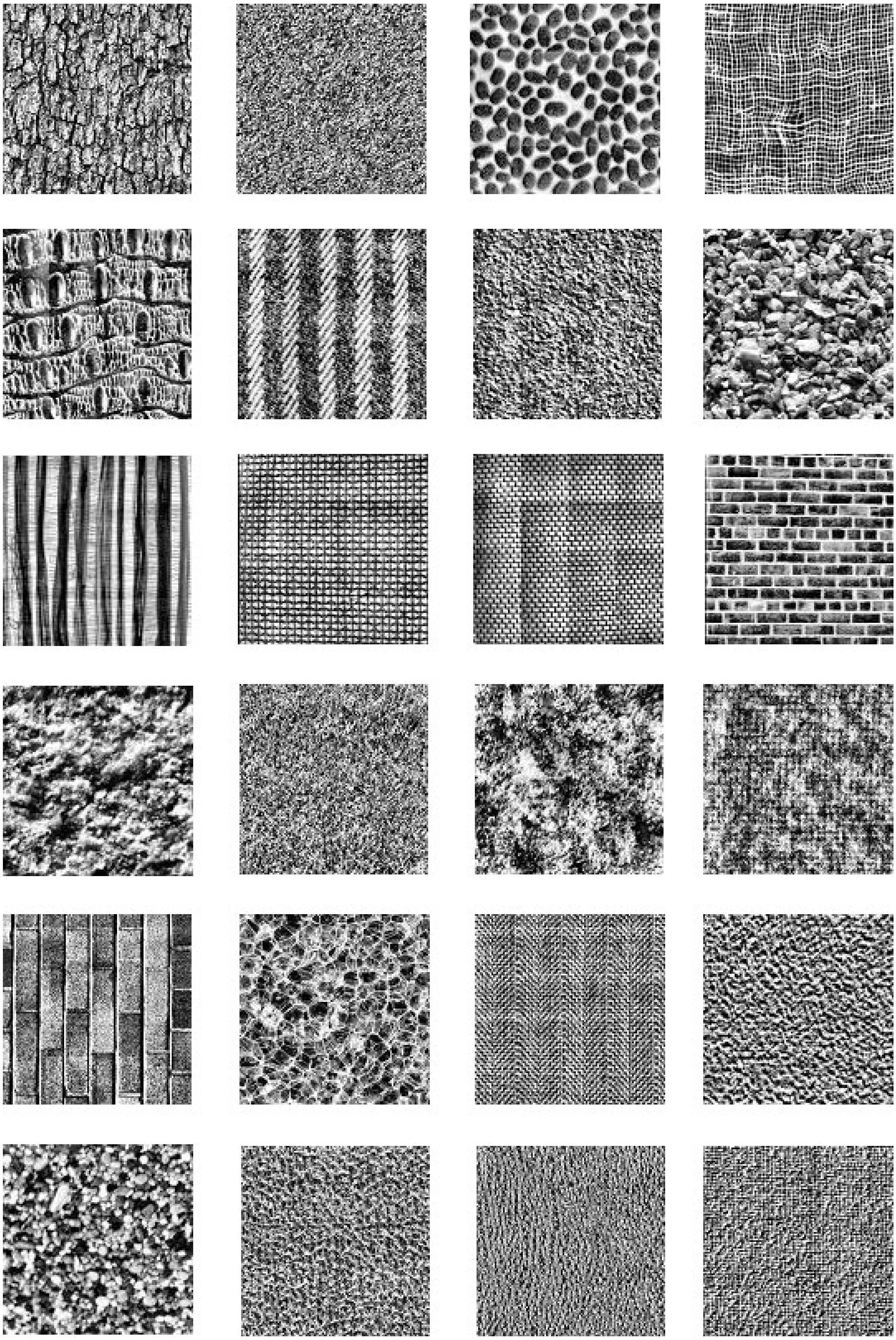
Van de Wouwer, G., Scheunders, P. & Van Dyck, D. (1999). Statistical Texture Characterization from Discrete Wavelet Representations. *IEEE Transactions on Image Processing*, 8(4), pp. 592-598.

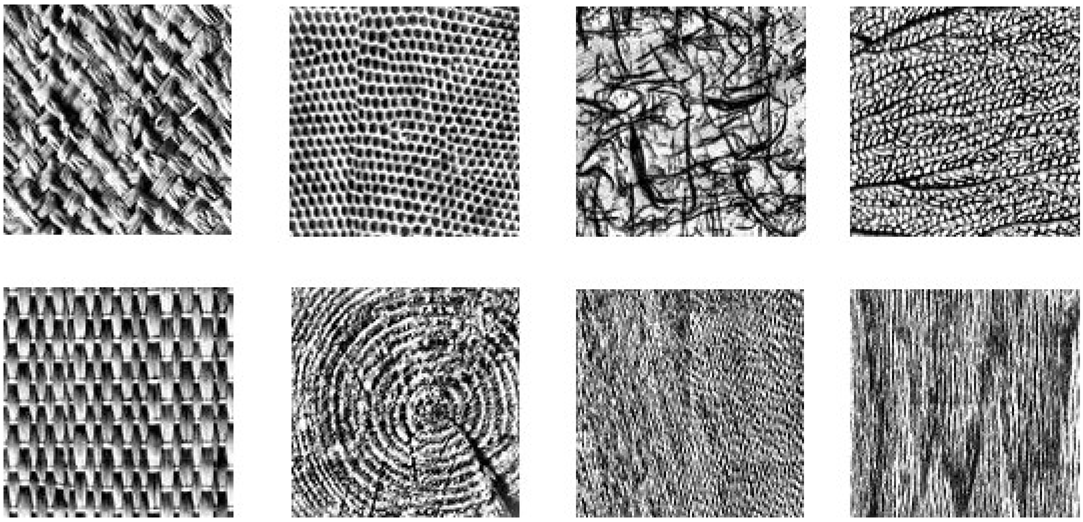
Wagner, T. (1999). Texture analysis. In *The Handbook of Pattern Recognition and Computer Vision*, Vol. 2, pp. 275-308, USA: Academic Press.

Zhou, D. (2006). *Texture Analysis and Synthesis using a Generic Markov-Gibbs Image Model*. Thesis. University of Auckland. Retrieved April 1, 2010, from http://www.cs.auckland.ac.nz/~georgy/research/texture/thesis-html/thesis.html

APPENDIX

32 BRODATZ TEXTURES USED IN EXPERIMENTS





EXAMPLES OF DATASET SAMPLES