LICENSE PLATE RECOGNITION USING WAVELET TRANSFORM AND CLUSTER-K-NEAREST NEIGHBOR CLASSIFIER

RAJA SYAHIRA BT RAJA ABDUL AZIZ

ELECTRICAL AND ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

LICENSE PLATE RECOGNITION USING WAVELET TRANSFORM AND CLUSTER-K-NEAREST NEIGHBOR CLASSIFIER

RAJA SYAHIRA BT RAJA ABDUL AZIZ

ELECTRICAL AND ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2010

License Plate Recognition Using Wavelet Transform and Cluster-k-Nearest Neighbor Classifier

By

Raja Syahira binti Raja Abdul Aziz

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Engineering (Hons)
(Electrical and Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

License Plate Recognition Using Wavelet Transform and Cluster-k-Nearest Neighbor Classifier

By

Raja Syahira Raja Abdul Aziz

A project dissertation submitted to the
Electrical & Electronics Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved by,

DR SAMIR BRAHIM BELHAOUARI
Senior Lecturer
Fundamental & Applied Sciences Department
Universiti Teknologi PETRONAS, PERAK

(Pn. Hanita Daud)	(Dr. Brahim Belhoauari Samir)
Project Supervisor	Project Co-Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
June 2010

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Raja Syahira Raja Abdul Aziz

ABSTRACT

Vehicles, especially cars have been an important asset in our life, and each vehicle has its own unique identity displayed on the license plate. Therefore, many vehicles related applications can be developed if the plate number can be recognized. Such applications are to be used in access control to a facility, traffic surveillance, toll booth system and parking system. This project proposes an implementation of License Plate Recognition in improving the current system of control access for Universiti Teknologi PETRONAS (UTP). The Plate Recognition methods utilized in this project are Wavelet Feature Extraction and Cluster-k-Nearest Neighbours Classifier. All the programs are carried out in MATLAB. Generally, the project is carried out by creating image database and the database of vehicles and owners information. A classifier using Cluster-k-Nearest Neighbour will train and test the plate number image features, which is obtained from wavelet transform. After the classifier achieves good result of testing, it is integrated with the database of the vehicle information. The system was said to be completely successful when an input image is inserted, the program will assign the right car and display all the information stored in the database corresponds to the car.

ACKNOWLEDGEMENT

With my sincere appreciation, I would like to express my utmost gratitude to my supervisor, Mrs. Hanita Daud for her detailed and constructive comments along with continuous help, support, motivation and guidance from the initial to the final level of the project.

My greatest gratitude and appreciation is extended to Dr. Brahim Belhoauari Samir especially for the entire efforts and contribution of his ideas in this project, and also for the guidance in MATLAB programming. His wide knowledge and logical way of thinking have been of great value for me.

Above all, I thank my beloved parents, Wan Rohaya bt Wan Abdul Ghani and Raja Abdul Aziz bin Raja Ismail, for always standing there beside me and continuously supporting throughout my life.

Last but not least, I express my regards and blessings to all lecturers and colleagues who supported me in any respect during the completion of the project.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Control access using License Plate Recognition is designed to improve the efficiency of the current system of car control access used in UTP. Since plate number is a unique mandatory identifier for vehicles, License Plate Recognition technology has been chosen to identify registered vehicles. This project's goal is to build a prototype, which is capable of recognizing a registered license plate number with a standard license plate in Malaysia in the database.

The heart of the system is the image processing module, which consists of the feature extraction and classification modules. The classifier will recognize and classify the vehicle. The final element is to compare the resulted plate number with the list of registered vehicles' plate number. A Graphic User Interface is built for the application to be easily operated.

1.2 Problem Statement

UTP is currently still using manual gate entrance which is lack of security efficiency. For the current system, every students and staffs using cars are given car stickers after registering their vehicle. The usage of car stickers is seen as unreliable as they can be easily removed and used for other unregistered vehicles.

With the rapid growth of the number of vehicles in UTP, there is a need to improve the present system for the identification of the vehicles. Uncontrolled vehicles entering UTP may threat the safety of students and staffs, especially those who are staying in the campus. To improve the security system, a new system of control access is proposed to replace the manual gate system. This system utilizes the license plate as an identification of registered vehicles. It consists of image acquisition module, image processing module and database. The system will check the vehicle registration plate against the database of registered vehicle.

1.3 Project Objectives

- To improve security level and efficiency of UTP access control by using a license plate classifier integrated with system database.
- To develop a reliable license plate classifier with a fast processing time. In this project the classifier is built using Cluster-k-Nearest Neighbors.
- To reduce work load of security personnel. With the implementation of this project, security personnel could focus more on the main entrance's visitors, where security is a higher concern.

1.4 Scope of Study

i. To develop a program that can process and recognize car plate number by using Wavelet Transform and Cluster-k-Nearest Neighbor classifier. This is to identify the vehicles approaching UTP main entrance.

ii. To design a control access system that holds the information of the registered vehicles.

To successfully implement the whole system, milestones that have to be achieved are:

- To fully understand the basic principles of wavelet transform algorithms and Cluster-k-Nearest Neighbor implementation.
- To develop a software as an interface between the recognition plate classifier and the database in a computer.

CHAPTER 2
LITERATURE REVIEW

2.1 License Plate

A vehicle registration plate is a metal or plastic plate attached to a motor vehicle or trailer for official identification purposes. The registration identifier is a numeric or alphanumeric code that uniquely identifies the vehicle within the issuing region's database [1]. Most governments require a registration plate to be attached to both the front and rear of a vehicle. National databases relate this number to essential information includes the manufacturer, model, color, year of manufacture, engine size, type of fuel used and the name and address of the vehicle's registered owner or keeper [1].

Figure 1: Sample of registration plate from different countries

2.1.1 Malaysian Car License Plate

In standard regulation, registration plate of Malaysian vehicles differs with other countries by having white characters with black background. A typical Malaysian number plate is depicted in an ABC 1234 format. The first letter determines the vehicle's registered location. For example: DH 3333 is a vehicle registered in Kelantan, because it begins with letter D. All vehicle license plates in Malaysia does not use the letters "O" and "I" because it can be confused with "0" and "1". Meanwhile "Z" is used for the military only [1].

2.2 License Plate Recognition

License Plate Recognition (LPR) is the method used by a computer to convert digital images of vehicle license plates into electronic text. Typical applications of LPR include private parking lot management, traffic monitoring, automatic traffic ticket issuing, automatic toll payment, surveillance, and security enforcement. Real time LPR plays a major role in automatic monitoring of traffic rules and maintaining law enforcement on public roads. There are a number of techniques used so far for recognition of number plates using image processing technology such as BAM (Bi-directional Associative Memories) neural network character recognition, pattern matching etc [2].

There are multiple license plate recognition systems available; however a majority of the systems are available for foreign plates only. License Plate Reading systems are nowadays considered as off-the-shelf systems in the panorama of image and video processing applications. Working systems that are being commercialized include the all-in-one system VEGA produced by Tattile, SeeCar by HI Tech Solutions, Adaptive Recognition Hungary. A sophysticated example of such systems is Autodetector, developed by Elsag s.p.a., that couples a robust system for LPR with a motion analysis suite, thus obtaining a system that can be mounted on a car [2].

2.3 Common License Plate Recognition Techniques

At present, the most commonly used car license plate recognition methods include segmentation and recognition process. Several methods have been proposed for the two tasks. For segmentation, techniques such as neural network [3], mathematical morphology [4], color analysis [5] and histogram analysis have been applied. While these methods are claimed to perform well, there are still limitations.

(i) Neural network method has self-learning, strong adaptability, and other advantages, but through a lot of exploration to determine the appropriate neural network model, algorithm and the set of parameters is difficult to achieve The learning algorithm of Artificial neural network (ANN) has disadvantages such as local optimization, slow convergence speed and long iteration time and so on, which influenced the accuracy of forecast seriously [10].

(ii) Morphological processing is a useful method to compensate for illumination variations. The basic procedure to estimate a bright background is to use the closing operation that consists of a sequential application of dilation and erosion using the same structuring element. The dilation operation fills in darker regions of an image as the brighter regions enlarge. If a given structuring element spans a dark region the dark region will disappear. This technique yields generally superior results; however, some artifacts are introduced to the non-uniform illumination of the license plate. Due to the closing operation, stroke width of the character increases, leading to touching characters in the string [4].

(iii) Color-based method may work well in controlled condition (such as light), light and background color will cause ambiguity [9].

(iv) Histogram based method is fast: but it is also sensitive to noise and rotation[9]. This type of segmentation process used includes threshold selection to extract the license plate region. In [22], a bimodal histogram segmentation is introduced. The line whose number of black pixels exceeds the threshold is selected. If the adjacent selected regions are sufficiently close, they are merged into one. If it is too large, they are discarded. However, the performance of this method is not quite satisfactory without some optimizations.

For recognition, there exist three main approaches: a) template matching. b) structural analysis, c) Neural network. Structural analysis uses a decision-tree to assess the geometric feature of each symbol's contour. The process is time-consuming. Neural network works by learning from examples and it works well if properly trained.

Among them, the template matching method works by matching the large target symbol with predefined standards. This method has the character of rather fast learning speed, simple principle, and be applied in some specific situations, such as under the condition of fixed size, number plate location level having no circumrotating, and so on. When the license plate images had been slightly twisted, strokes been uneven or even damage and rupture, it would easily lead to wrong identification [7, 8].

2.4 Wavelet

The fundamental idea behind wavelets is to analyze according to scale. Wavelets are functions that satisfy certain mathematical requirements and are used in representing data or other functions. In fact, this is not a new idea. Approximation using superposition of functions has existed since the early 1800's, when Joseph Fourier discovered that he could superpose sinus and cosines to represent other functions [14]. Fourier analysis has been a traditional and efficient tool in many fields of science and engineering, in past two hundred years. However, Fourier analysis has its own deficiency. It has two major problems [12], namely:

- Fourier analysis can not characterize the signals locally in time domain. In transforming to the frequency domain, time information is lost. This makes the Fourier transforms less than optimal representations for analyzing signals, images and patterns containing transient or localized components.
- Fourier expansion can approximate the stationary signals well, but unable do so for the non-stationary signals.

Unlike the Fourier transform, whose basis functions are the sinusoids, wavelet transforms are based on small waves, called wavelets, of varying frequency and limited duration. One major advantage afforded by wavelets it the ability to perform local analysis, that is, to analyze a localized of a larger signal.

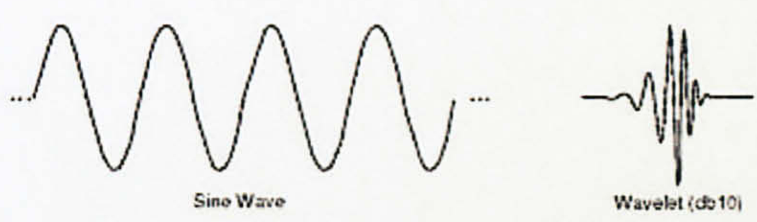


Figure 2: Sine wave and wavelet

2.4.1 The Wavelet Functions

In wavelet transformation, there are many basic functions can be considered as mother wavelet. Since the mother wavelet produces all wavelet functions used in the transformation through translation and scaling, it determines the characteristics of the resulting wavelet transform. There are several types of wavelet families used in wavelet transform. Each of them hold special properties and give different wavelet functions. Therefore, a wavelet is chosen depending on the application.

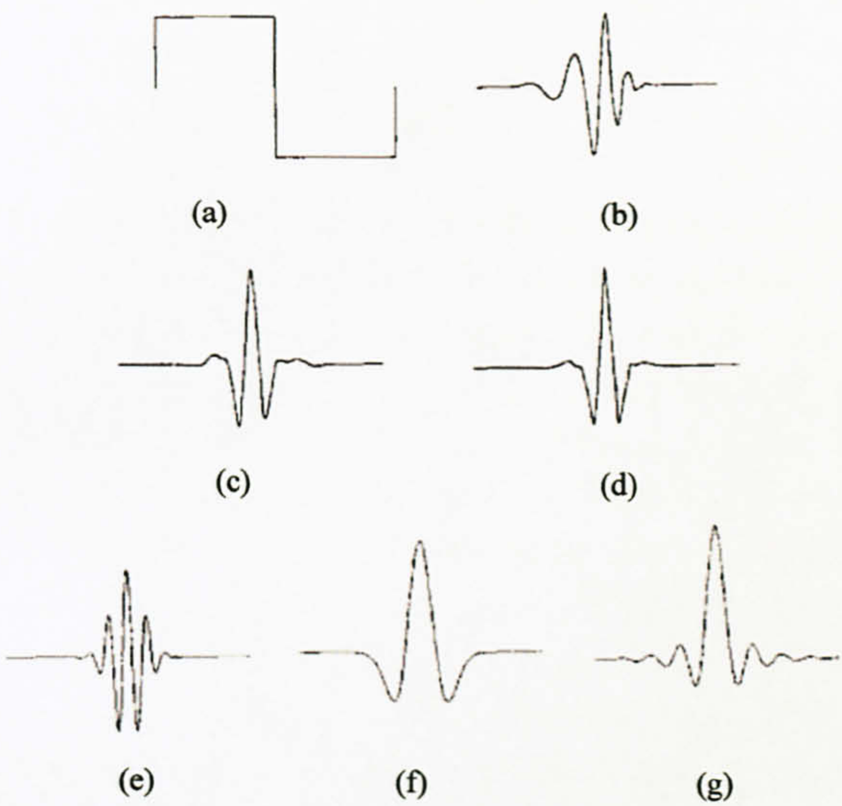


Figure 3: Example of wavelet from different wavelet families (a) Haar (b) Deubechies (c) Symlet (d) Coiflet (e) Morlet (e) Mexican (f) Meyer

Figure 3 illustrates some of the commonly used wavelet functions. Daubechies wavelets represent the foundations of wavelet signal processing and are used in numerous applications. Among them, Haar wavelets belonging to daubechies wavelet family are most commonly used wavelets in database literature because they are easy to comprehend and fast to compute [23]. Haar transform can be viewed as a series of averaging and differentiating operations on a discrete function. The Haar, Daubechies, Symlets and Coiflets are compactly supported orthogonal wavelets[24]. These wavelets along with Meyer wavelets are capable of perfect reconstruction. The Meyer, Morlet and Mexican Hat wavelets are symmetric in shape.

2.4.2 Wavelet-Based Texture Feature Set

Pattern recognition is concerned with the design and development of methods for the classification or description of patterns, objects, signals, and processes. In the case of image classification, because of the image size involved, it is essential to extract the features of the image, so that the pattern or image can be then classified based on the features extracted. It is important to select features that represents the image uniquely and contains most discriminatory information for robust classification. The features extracted should reflect global properties of the image[11].

Huang Wei [15] proposed feature extraction method based on wavelet packet for license plate characters. Through wavelet packet decomposition of the normalized character images, the feature vector is composed of wavelet packet decomposition coefficients. The finding came from Lee [16] with his research on a new multi-resolution recognition scheme for recognizing handwritten numerals using two-dimensional wavelet transform and Multilayer Cluster Neural Network to classify the characters. In [17], bank note images are first pre-processed by performing edge detection in order to facilitate the wavelet feature, and classified using the Euclidean minimum distance matching.

2.4.3 Two-Dimensional Wavelet Transform

A two-dimensional (2-D) wavelet transform, which is a separable filter bank of row and column directions, decomposes an image into four sub images, one approximate sub band, and three detail sub bands (horizontal, vertical, and diagonal) are shown in Figure 4. A single forward wavelet of an image is accomplished by two separate 1-D transforms. The image f(x, y) is first filtered along the x direction, resulting in a low pass image and high pass image. Since the bandwidth of the two images along the x direction is now half of f, each of the filtered images can be down sampled in the x direction by two without loss of information. The down sampling is accomplished by dropping every other filtered value. Both images are then filtered along the y direction[11].

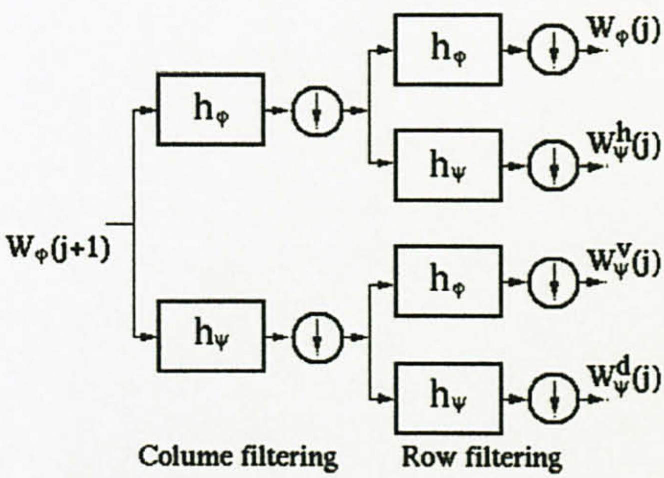


Figure 4: 2-D Wavelet decomposition structure

Where h_ψ is a low pass filter and h_φ is a high pass filter which split a signal's bandwidth into half. The impulse response of h_ψ and h_φ are mirror images.

In 2-D discrete wavelet transform, a scaling function, $\varphi(x, y)$and three wavelets $\psi^H(x,y)$, $\psi^V(x,y)$, and $\psi^D(x,y)$ are required[25]. Each is the product of two-dimensional functions. These wavelet measures variation along different directions: ψ^H measures variations along columns, ψ^V responds to variations along rows and ψ^D corresponds to variations along diagonals[25]. First, the scaled and translated basis functions are defined:

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\,\varphi(2^j x - m,\ 2^j y - n)$$

$$\psi^i{}_{j,m,n}(x,y) = 2^{j/2}\,\psi^i(2^j x - m,\ 2^j y - n) \quad i = \{H,V,D\}$$

Where index *i* indentifies the directional wavelets.

The 2-D discrete wavelet transform of image *f(x,y)* of size $M \times N$ is then

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\varphi_{(j_0,m,n)}(x,y)$$

$$W_{\psi^i}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\psi^i{}_{(j,m,n)}(x,y), \quad i = \{H,V,D\}$$

2.5 Reducing Number of Feature Extraction from Wavelet Transform

A statistical method was introduced in [21] in order to select and reduce wavelet transform coefficients to which represent most significant effect on the classification accuracy. This problem refers to the properties of wavelet functions (families) that are lead to produce high number of coefficients represent the features which are redundant for the purpose of classification. With implementation of this method, only coefficients with higher variance between classes stay. The variance of the class compared to the overall variance is represented as

$$Var\_mod = Min\left\{\frac{(m_i - m_T)^2}{Var_i}\right\}$$

Where, Var_i is variance of the class, m_T is total mean, m_i is class mean, i is index of class, and n is number of classes. If the *Var_mod* is less than the value of threshold, they delete all coefficients belong to this column and they didn't taken in our consideration, otherwise we keep it for further process. Figure 5 shows three classes with overlapping coefficients, with the total mean and each class mean.

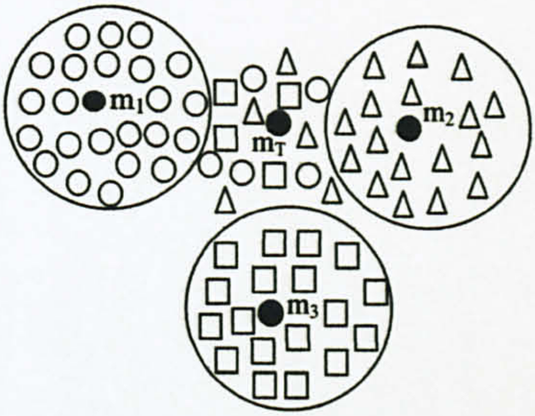


Figure 5: Three classes with overlapping coefficients

2.6 Image Classification

Classification process can be categorized into two types [12] :

- The classification with supervised learning. In this type of classification, there is a supervisor to teach the recognition system how to classify a known set of patterns, and thereafter, it let the system go ahead freely to classify other patterns. In this way, a priori information is needed to form the basis of the learning [13].
- The classification with non-supervised learning. The classification process is not depending on prior information, which means generating classes without any prior knowledge about the patterns; neither can the proper training pattern sets be obtained.

2.6.1 K-Nearest Neighbors

A classification task usually involves with training and testing data which consist of some data instances. K-nearest neighbor is a supervised learning algorithm where the result of new instance query is classified based on majority of k-nearest neighbor category. The purpose of this algorithm is to classify a new object based on attributes and training samples. The classifiers do not use any model to fit and only based on memory. Given a query point, we find k number of objects or (training points) closest to the query point. The classification is using majority vote among the classification of the k objects. Any ties can be broken at random. K Nearest neighbor algorithm used neighborhood classification as the prediction value of the new query instance[18]. The k-nearest-neighbor searching problem is to find the k nearest points in a dataset containing n points to a query point, usually under the Euclidean distance. It has applications in a wide range of real-world settings, in particular pattern recognition, machine learning [19] and database querying [20].

2.6.2 K-Means Clustering

K-means clustering algorithm was developed by J. MacQueen (1967) and then by J. A. Hartigan and M. A. Wong around 1975 [18]. Basically, k-means clustering is an algorithm to classify or to group your objects based on attributes or features into *k* number of group. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

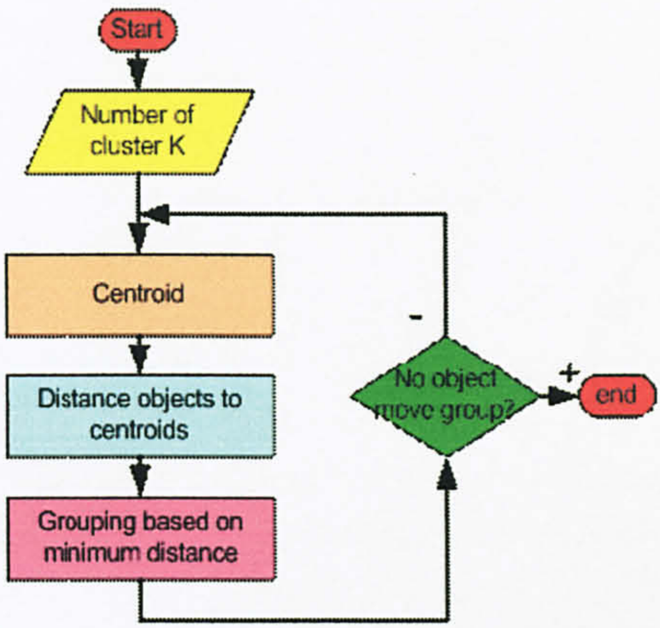


Figure 6: K-means clustering algorithm

Based on Figure 6, the step by step k-means clustering algorithm listed below[18] :

1. Begin with a decision on the value of *k* = number of clusters.
2. Put any initial partition that classifies the data into *k* clusters. The training samples may be assigned randomly, or systematically as the following:
   (i) Take the first *k* training sample as single-element clusters

(ii) Assign each of the remaining training samples to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster.

3. Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.
4. Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

2.6.3 Cluster k-Nearest Neighbors (C-k-NN)

C-k-NN is a new technique which combines k-NN with k-Means Clustering to reduce the classification time for k-NN. With C-k-NN, the training data is clustered into subclasses, so that the classification time will depend just on the number of subclasses m_i, with m_i the number of subclasses in the class C_i. Each subclass contains a random number of data, which is relatively close to each other[26].

Each class, C_i, should be a cluster to several subclasses, $C_{i,j}$, with $1 \leq j \leq mi$, and each subclass will be represented by it mean, $\mu_{i,j}$. To find the best number of subclasses, the number of subclasses shall be iterated starting from 1 until two conditions meet to stop the iteration[26]:

(a) All the representatives or centroid $\mu_{i,j}$ have to be closer to their classes C_i or subclasses $C_{i,j}$ than to any other classes. This is to reduce the misclassification (i.e. reduce error in the classification of our own training data).

(b) The variance of each class C_i, Var_i, does not decrease drastically in comparison to the previous iteration. We may use $\frac{\Delta Var}{Var} \leq \alpha$ as a criteria to quantify if there is a decrease or it is still approximately remain constant. In certain cases, it is better to stop the iteration if the condition has been checked twice or more (i.e., after which the variance will be smoothen).

After that, classification algorithm of k-NN is used to classify by using the representative data. Common k-NN classifier is generally based on the Euclidean distance between a test sample x and the specified training samples but in C-k-NN, we introduce other metrics in order to better estimate the density probability. Let $x_{i,j}$ belongs to the subclass j of the class i, denoted by $C_{i,j}$, and for all positive number s we may define the following metric[26]:

$$d(x, x_{i,j}) = \frac{d^s_{euclidean}(x, \hat{x}_{i,j})}{n_{i,j}}$$

Where $\hat{x}_{i,j}$ represents $C_{i,j}$ and $n_{i,j}$ is the variance of the set $C_{i,j}$.

CHAPTER 3
METHODOLOGY

3.1 Procedure Identification

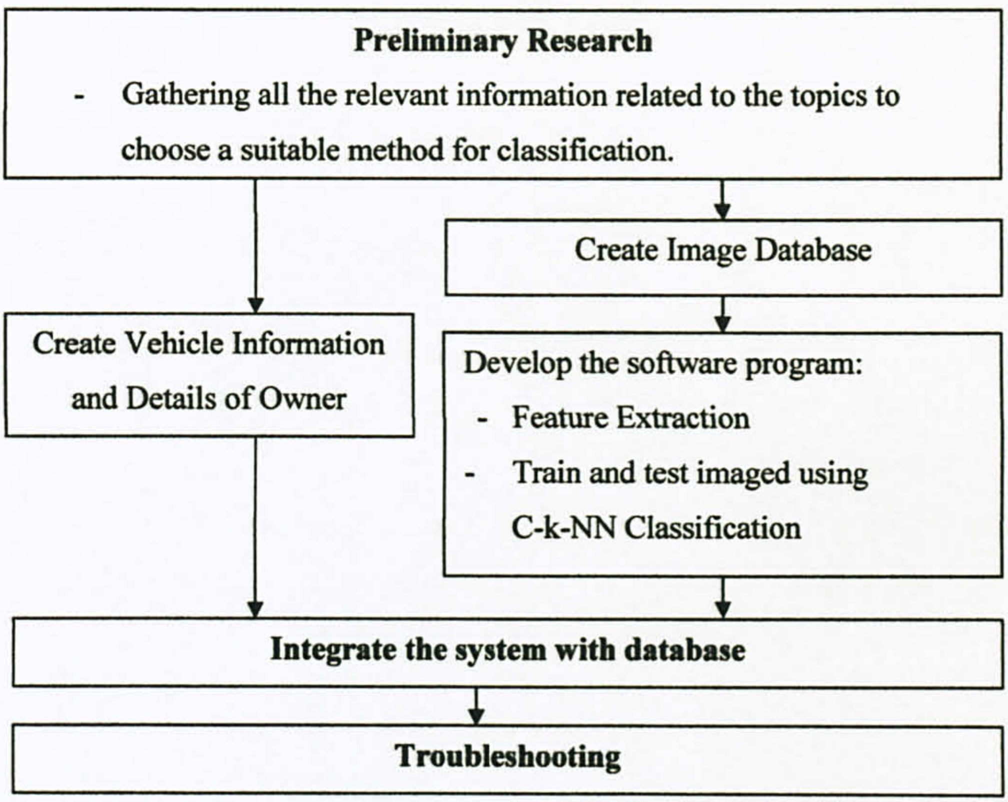


Figure 7: Flow chart research methodology

3.2 Overall System Overview

The objective of this project is to develop an access control system with robust license plate recognition technology. The system is capable to recognize car plate number then it will be compared with the list of registered vehicles plate numbers in the database. If the plate number is inside the list, the system will allow the particular car enter the facility.

The software design is involving the following things:

- Image acquisition module which will capture the image of a car.
- Image processing module that will process the image of a car and return the plate number. This includes the wavelet transform and C-k-NN neighbor classifier.
- Database system that will handle the personal information of the cars' owner and the several plate images of the car.

2-D wavelet transform will be applied on the image which results in four sub images as shown in Figure 4. For the purpose of image classification, the three detail images and the Approximate sub image is combined and converted into a vector by concatenating the columns. The selected wavelet feature will be used as input to a C-k-NN Classifier. Therefore, each car with several images of plate number builds one class. In response to a set of input features, the output of the classifier is actually a set of numbers representing the probability estimate of each class. The final classification is obtained by choosing the class with the highest probability estimate. The final output will be the information of the car matched in the list of the database. If the probability not reaching certain pre-defined value, for example 0.5, the input is considered out of record, or unregistered.

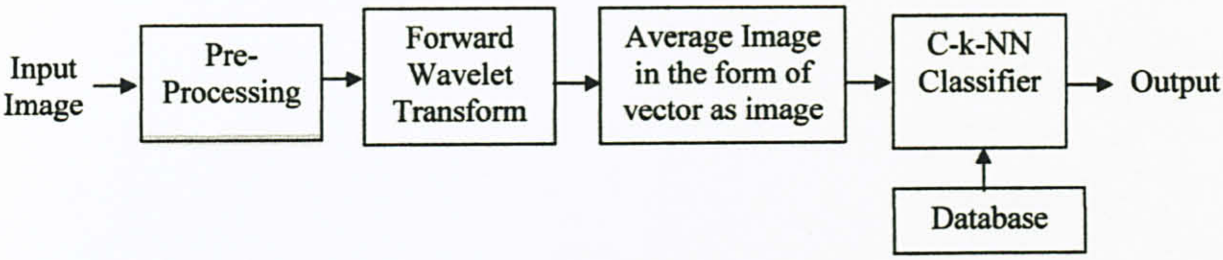


Figure 8: Block diagram of the overall system

3.3 Image Acquisition Module

Figure 9: The illustration of image acquisition setup

The above illustration shows the main idea of the proposed security access system and the position license plate images captured. The system is quite similar with the existing typical access-control system which using vehicle identification. The only difference is that this system will utilize the license plate number as the passing code word and is integrated with the database.

To produce the image database, several images need to be captured for each car. This includes different angle view and different light exposure but with fixed camera position. Once the images are ready, the plate regions are cropped manually from the images. Then, they are converted to gray-scale before undergo further processes.

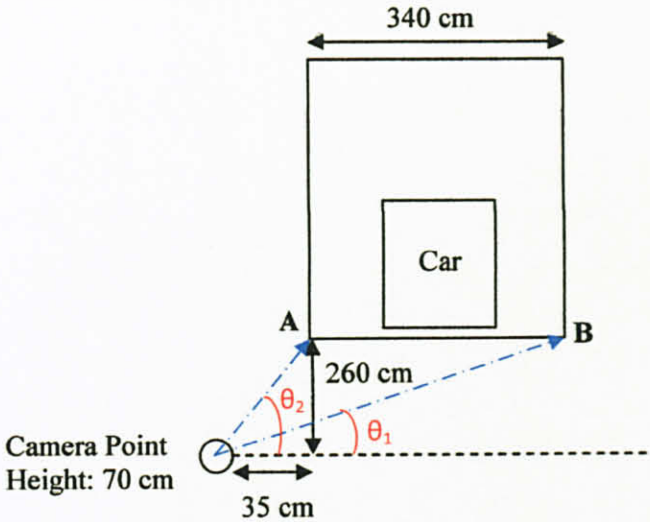


Figure 10: Camera and car position setting

The distance between the camera and the car follows the setup as in Figure 9. The subtraction of θ_2 by θ_1, which represents the angle between point A and B are fixed, while the angle of the car itself may vary. The values for θ_1 and θ_2 are calculated as below.

$$\tan\theta_1 = \frac{260\ cm}{375\ cm}$$
$$\theta_1 = 34.73°$$

and

$$\tan\theta_2 = \frac{260\ cm}{35\ cm}$$
$$\theta_2 = 82.33°$$

3.4 Tools

The major tool for this project is MATLAB 2007 with a requirement Image Processing Toolbox and Wavelet Toolbox. Other equipment needed is a camera to capture the image. It is better to use a high resolution camera as it can produce high quality images. In this project, a camera digital with resolution 3264 x 2448 (8 Mega pixels) is used.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Image Database Collection

For this project 20 images are captured for each class of 4 different cars. From the original image, the plate number region is cropped out manually to take out plate number region only. Figure 11 and 12 show an example of pictures prepared for one car. Images are resized in same size 158 x 60 pixels.

Figure 11: The original image

Figure 12: Image database for Class 1

All the images can be classified to the four classes as shown in table below.

Table 1: Details of classes used in database

Classes	Image No.	Plate No.
Class 1	1-20	BHB 2783
Class 2	21-40	NBH 3160
Class 3	41-60	DAU 9843
Class 4	61-80	PEK 7224

4.2 Image Enhancement

For plate recognition, image sharpness is very important aspect since the quality of a plate image can vary significantly. For example, the quality of the image depends very much on the lighting or brightness level, and noise levels. Before further processing, all the images are converted to gray scale. Then, a sharpening filter has been applied, where the images are sharpened by subtracting a blurred (unsharp) version of the image from itself. A sample of image before and after enhancement is shown in figure below. In Figure 13(b), the plate numbers edge becomes clearer and sharper than in Figure 13(a).

Figure 13(a): Original image before image enhancement

Figure 13(b): Original image after image enhancement

4.3 Feature Extraction

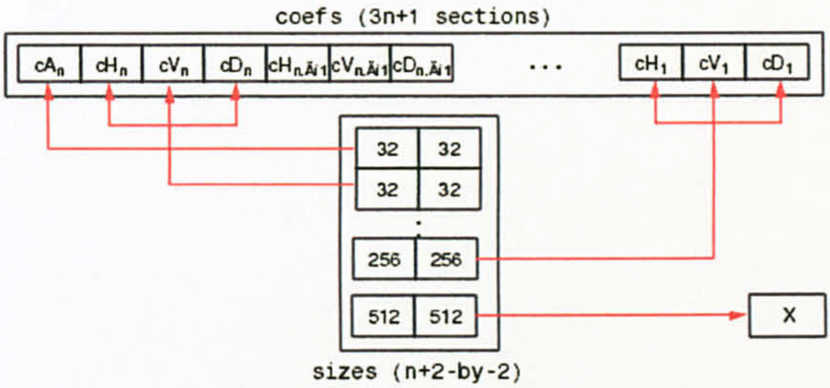


Figure 14: Wavelet Transform coefficients

Figure 14 shows how a vector of concatenated DWT coefficients is formed from the variable *coefs*. The *coefs* vector for a n-level decomposition with matrix size specified by variable size, which consists of approximation coefficients, followed by the horizontal, vertical, and diagonal detail coefficients. To visualize the wavelet decomposition, the author has made an analysis as seen in Figure 15. The original image, synthesized image, and all the sub-band images produced from the decomposition can be observed.

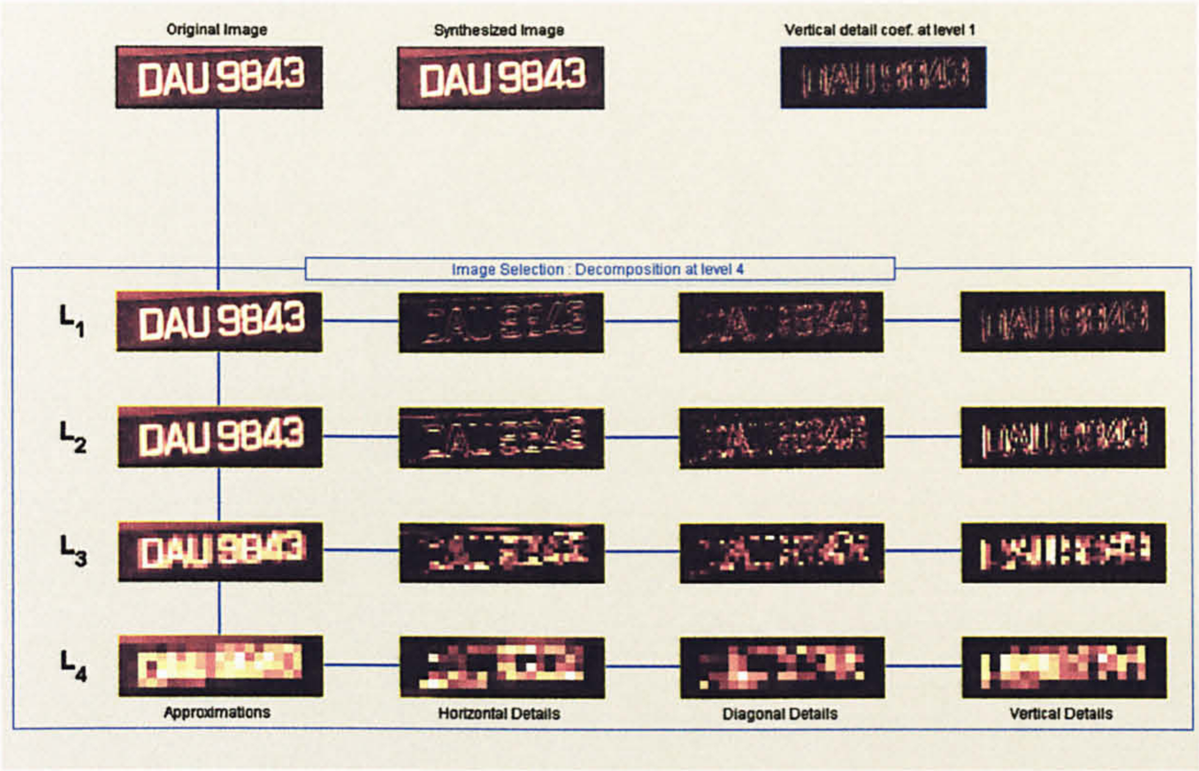


Figure 15: Image decomposition using dB1 wavelet at level 4.

After performing the wavelet transform, a statistical method as in [21] was applied to reduce the features extracted number which are represent the plate images. Some coefficients of wavelet transform are sometimes redundant and do not contribute information for the classification are deleted. After applying the threshold of the coefficients, the selected feature extracted is then proceed to training and testing procedure for classification. The main idea using this method is with higher threshold, more coefficients with small variance between classes are eliminated. Therefore smaller number of coefficients is being processed, which results in smaller processing time.

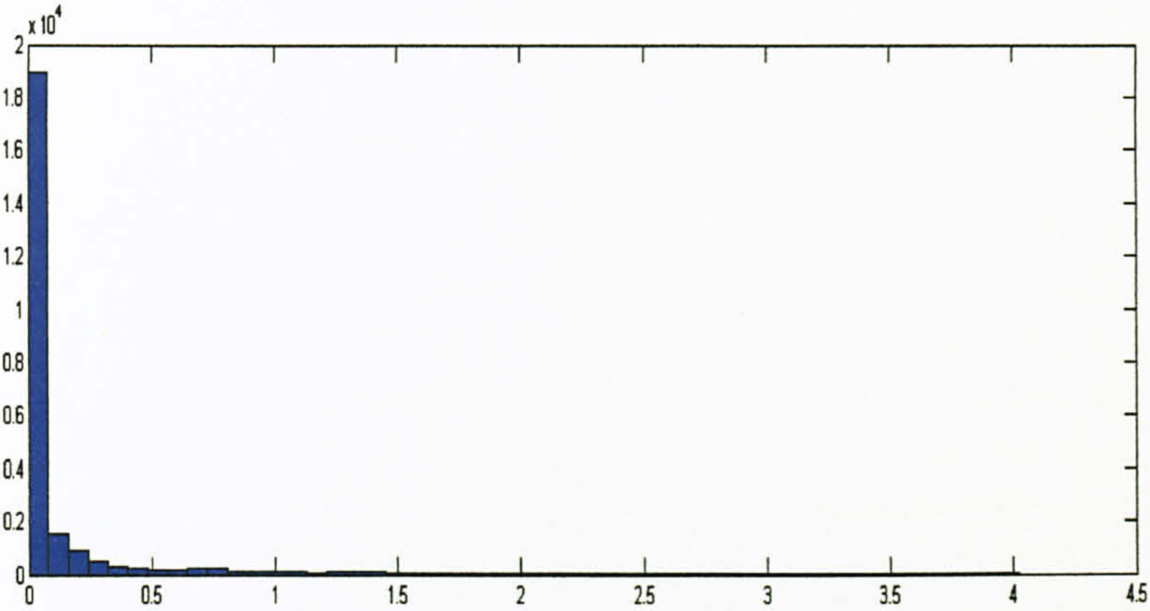


Figure 16: Plot of wavelet coefficients variance at level 5

The above figure shows the plot of the variance for a level 5 wavelet decomposition. The variable on x-axis represents *Var_mod (Refer section 2.5)* and the variable on y-axis refers to the frequency of the coefficients with the corresponding *Var_mod.* Notice that large number of coefficients falls into the region with low variance between the classes. In the next section, the author has performed the classification method with different threshold value of the *Var_mod* and the result was compared.

4.4 Proposed Classification Method

From the 20 images per class, 12 of the images were used for training and the other 8 images were used for testing to assess preliminary classifier performance. In classifying the visual texture, the image features are extracted and then using the wavelet decomposition to obtain the transformation coefficients of the wavelet functions. For the proposed method, two types of wavelet functions had been used, which are dB1 and sym8. dB1 is actually identical to Haar wavelet. Both wavelet functions are trained and tested using the proposed Cluster-k-Nearest Neighbors method and compare with the Euclidean distance method. The results of the testing are shown in tables below.

Table 2: Level 5 decomposition testing result using sym8 wavelet

Threshold	Coefficients	Euclidean	CKNN
0.0	23268	45.00%	71.88%
0.1	4587	60.00%	72.57%
0.2	3307	91.25%	89.93%
0.3	2543	95.00%	89.24%
0.4	2146	96.25%	86.81%
0.5	1888	93.75%	91.67%
0.6	1714	92.50%	90.97%
0.7	1497	90.00%	87.15%
0.8	1193	70.00%	76.39%
0.9	1061	67.50%	75.35%
1.0	957	70.00%	95.14%

Table 3: Level 6 decomposition testing result using sym8 wavelet

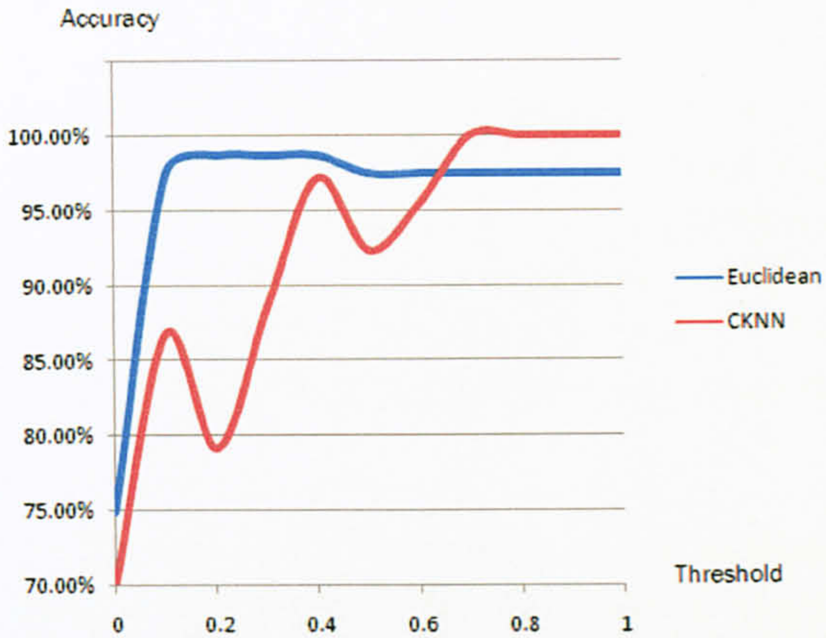
Threshold	Coefficients	Euclidean	CKNN
0.0	24288	45.00%	69.00%
0.1	4831	46.25%	72.57%
0.2	3388	91.25%	89.93%
0.3	2572	95.00%	90.28%
0.4	2149	95.00%	86.81%
0.5	1889	93.75%	90.63%
0.6	1714	92.50%	89.93%
0.7	1497	90.00%	87.15%
0.8	1193	70.00%	76.39%
0.9	1061	67.50%	76.74%
1.0	957	70.00%	96.53%

Table 4: Level 5 decomposition testing result using dB1wavelet

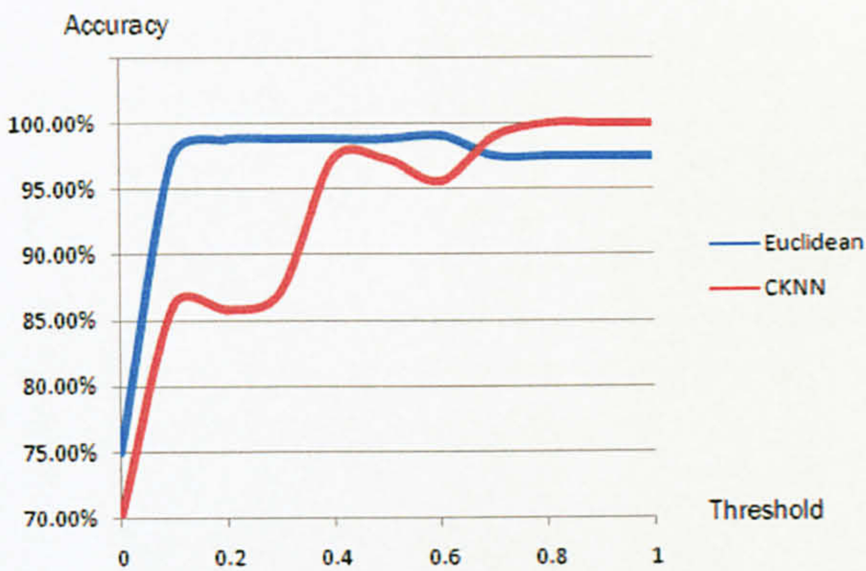
Threshold	Coefficients	Euclidean	CKNN
0.0	12720	75.00%	70.00%
0.1	1243	97.50%	86.81%
0.2	685	98.75%	79.17%
0.3	486	98.75%	88.54%
0.4	352	98.75%	97.22%
0.5	232	97.50%	92.36%
0.6	122	97.50%	95.49%
0.7	39	97.50%	100%
0.8	31	97.50%	100%
0.9	27	97.50%	100%
1.0	19	97.50%	100%

Table 5: Level 6 decomposition testing result using dB1wavelet

Threshold	Coefficients	Euclidean	CKNN
0.0	12732	75.00%	70.00%
0.1	1245	97.50%	86.11%
0.2	686	98.75%	85.76%
0.3	487	98.75%	87.15%
0.4	353	98.75%	97.22%
0.5	233	98.75%	97.20%
0.6	123	99.00%	95.49%
0.7	40	97.50%	98.96%
0.8	31	97.50%	100%
0.9	27	97.50%	100%
1.0	19	97.50%	100%



(a)



(b)

Figure 17: The performance of C-k-NN compared to Euclidean Distance classifier for dB1 wavelet from (a) Level 5 decomposition and (b) Level 6 decomposition

4.4.1 Discussion

From the results, we can see that dB1 wavelets give better result compared to sym8 wavelet function. In addition, the decomposed wavelet coefficients size number for dB1 is less than that of symlet wavelet for the two levels. Smaller number of coefficients is preferred because it requires less time to process. At both level 5 and 6 decomposition of dB1 wavelet, higher threshold value gives a result of 100% of the testing results. This means that all 8 images for testing are assigned to their classes correctly.

Figure 17 (a) and (b) shows graph developed from the result tabulated in Table 4 and 5. From the graph, we can see that at the first, with low threshold value, the result of C-k-NN is lower compared to Euclidean distance method. However the accuracy starts to increase and C-k-NN give the highest result at threshold 0.7 or 0.8 and above. Based on this finding, dB1 wavelet had been used for the classification process since it is simpler and faster.

After achieving good result for testing, all the trained data for the selected threshold was then saved to be used for classification process. These data can be called back in the program. The classification processing time for a given single input image takes about 0.2 seconds, excluding the training and testing procedure. The time refers to the time taken to assign the class to the input image.

4.5 Graphic User Interface (GUI)

A user-friendly Graphic User Interface (GUI) was built to integrate the classification module with the database of the registered vehicles. Through the GUI the user can insert an image of license plate. By entering the image, it will work as an input to the classification program. The status of the incoming vehicles will be stated either it is registered or not. If the plate number inserted is in the list of the registered vehicles database, the status will appear to be registered. The program will assign the image to a respective class, and display all the information retrieved for that particular class as shown in Figure 18.

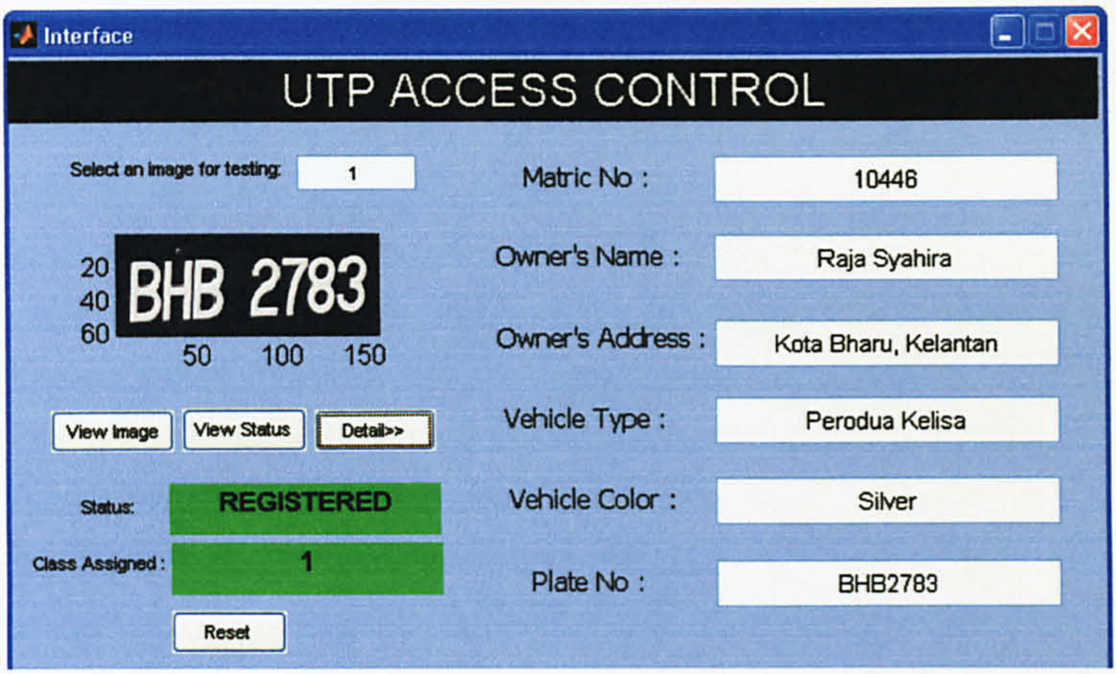


Figure 18: Graphic User Interface

4.6 Trust Confidence of the Images

From the previous result, the Cluster k-Nearest Neighbors Classifier have successfully assigned any given test image into a class. However, problem arises when a test image is not from the registered database, or in other words the input is non-registered vehicles plate images. The classifier output will always assign the input images a predicted class. Therefore, a method has been carried out to distinguish the input images are registered or not by calculating the trust confidence. The trust confidence, with value between 0 and 1, represents the probability of confidence level of the classification result. If the value of trust confidence is 1, this means that the classifier is 100% confident with the classification result. On the other hand, zero trust confidence means the classifier is not confident at all with the class assigned for the input image. Referring to Figure 19, assuming the test image is assigned to class 1, the trust confidence of a test image is calculated using the following equation.

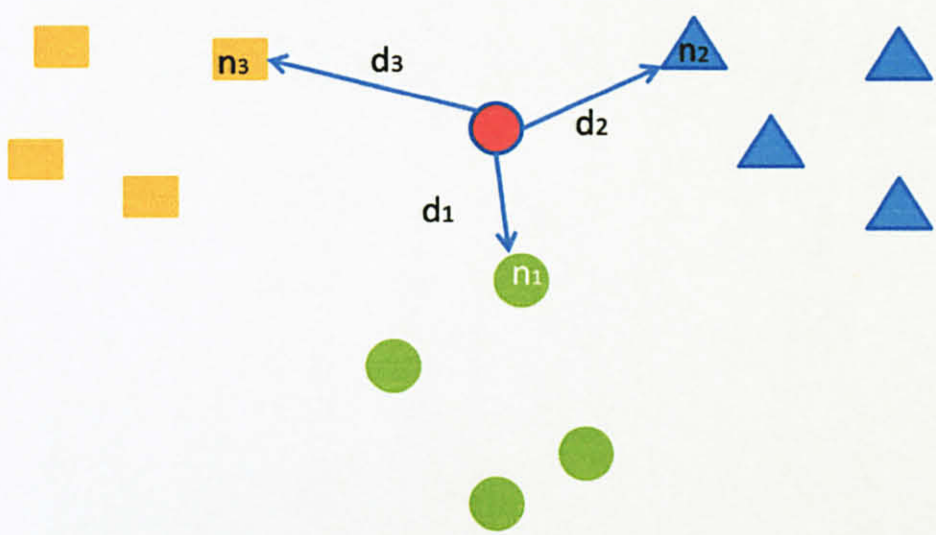


Figure 19: A test image to be classified into one of three classes with their subclasses.

$$Trust\ Confidence = 1 - \left(\frac{d_1 n_1}{d_1 n_1 + d_2 n_2 + d_3 n_3}\right)$$

Where,

d_1= Euclidean distance between the test image and the nearest subclass in Class 1

d_2= Euclidean distance between the test image and the nearest subclass in Class 2

d_3= Euclidean distance between the test image and the nearest subclass in Class 3

n_1= Number of data in the nearest subclass in Class 1

n_2= Number of data in the nearest subclass in Class 2

n_3= Number of data in the nearest subclass in Class 3

8 images for 4 classes of registered images and 3 images for 5 classes of non-registered images have been tested and the trust confidences for each of the tested image are calculated. Table 6 and 7 below show the trust confidences for each of the tested images. The average, maximum and minimum values of the trust confidence for tested images are shown in Figure 19.

Table 6: List of trust confidence for registered images

Images	BHB2783	NBH3160	DAU9843	PEK7224
1	0.9995	0.9917	0.9475	1
2	0.9758	0.9976	0.9962	1
3	0.9989	0.991	0.9298	1
4	0.984	0.9623	0.8678	1
5	0.9869	0.9788	0.8554	1
6	0.9672	0.9721	0.8329	1
7	0.9493	0.9996	0.8592	0.9967
8	0.966	0.9741	0.9178	1
Average :	**0.97845**	**0.9834**	**0.900825**	**0.999588**

Table 7: List of trust confidence for non-registered images

Images	DBM4222	DAX6955	CCH7003	WQW8177	TV9269
1	0.5885	0.072	0.2982	0.451	0.5726
2	0.5653	0.1575	0.211	0.5334	0.376
3	0.2212	0.793	0.403	0.2225	0.7789
Average :	**0.458333**	**0.340833**	**0.304067**	**0.4023**	**0.575833**

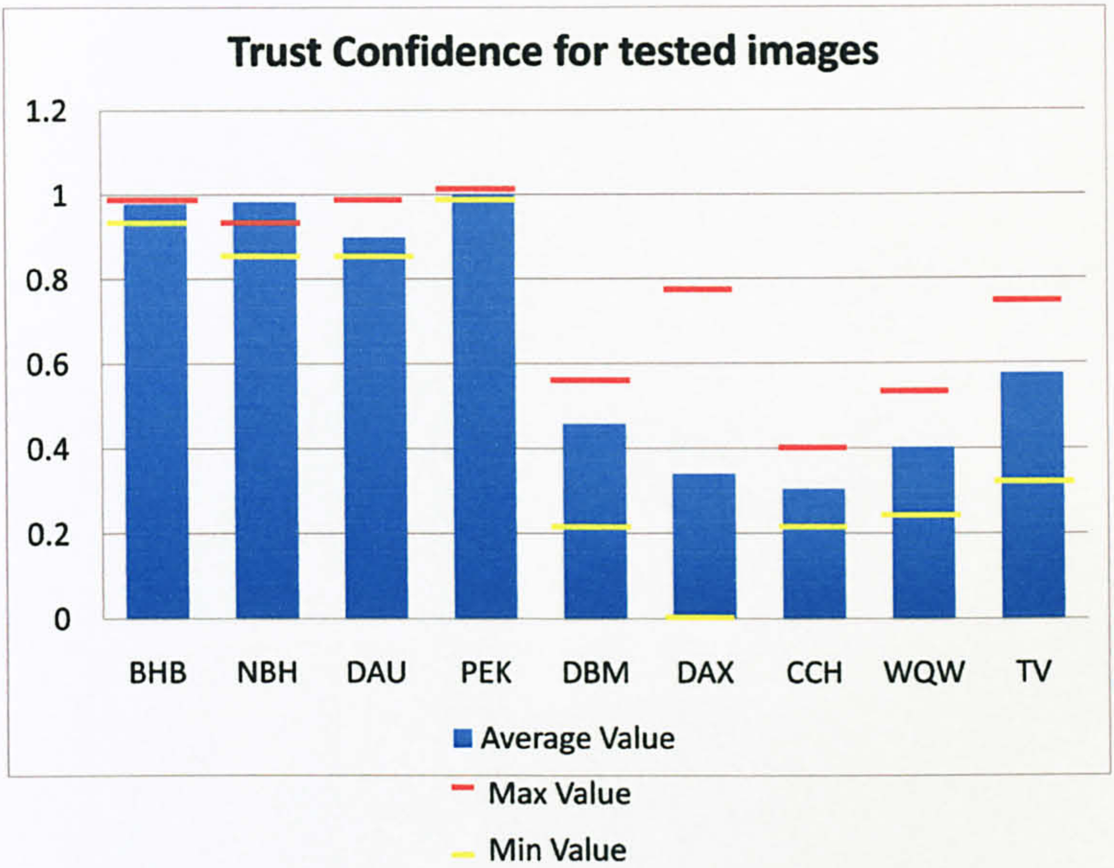


Figure 20: The average, maximum and minimum value of the trust confidence for tested images

Based on the graph in Figure 19, the highest maximum trust confidence value for the registered images is 1 for PEK7224, while the lowest minimum value is 0.8329 for DAU9843. For the non-registered images, the highest maximum trust confidence value is 0.793 and the lowest minimum value is 0.072, both from

DAX6955. All these values are analyzed to set a threshold value for the trust confidence boundary between registered and non-registered images. Based on these findings, a threshold value of 0.8 is used. If the trust confidence value is greater or equal than 0.8 the input image is accepted as registered vehicles. Trust confidence lower that 0.8 means the confidence level is low, therefore it is considered as non-registered vehicles.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Plate Recognition using Wavelet Transform and Cluster-k-Nearest Neighbor classifier is proposed to improve security efficiency of access-control for UTP and at the same time reduce work load of security personnel. A classification method was introduced for plate recognition which works under various lighting, shadow and angle of rotation. The working system was tested successfully and the system is still need to improve to meet the security requirement of control access for campus.

5.2 Recommendation

For optimal implementation of this project, there is some room for improvement to be made. To increase the performance of the classifier, more images per class can be added. For real implementation, more classes are stored in the database. Since the existed system is offline, the author recommends that this method is incorporated with a real-time application system with video stream. This would be very useful for 24 hours control access for the residence. Besides, the program can be linked with an automatic detection of vehicles and automatic gate entrance. It is not impossible that the system is capable of being implemented in future.

REFERENCES

REFERENCES

[1] 2010, Wikipedia, Vehicle registration plate
http://en.wikipedia.org/wiki/Vehicle_registration_plate

[2] Francesca Odone, "Experiments on a License Plate Recognition System", DISI, Università degli Studi di Genova, Technical Report number: DISI-TR-06-17

[3] J.W. Lee, I.S.Kweon, 1998, *"Automatic number-plate recognition: neural network approach"*, IEEE Vehicle Navigation and Information Systems Conference, vol.3.1. no.12, pp. 99-101,

[4] M.Shridhar, J.W. Miller, G.Houle and L.Bijinagte, 1999, *"Recognition of license plate: issues and perspectives"*, IEEE Processing of International Conference on Document Analysis Recognition, pp. 17-20,

[5] E.R. Lee, P.K. Kim, H.J. Kim, *"Automatic Recognition of a car license plate using color image processing"*, IEEE International Conference, vol.2, pp.301-305, 1994.

[6] Nai-ping H., Li W., and Yan-ping Z., 2006, *"The Design of Developed BP Arithmetic and Its Application in Identifying The Number Plate of The Vehicle"*, Control & Automation, vol.22(9-1), pp.313-314.

[7] Wu W., Qi-sen Z., Ming-jun W., 2001, *"A method of number-plate recognition templates matching"*, China Journal of Highway and Transport, vol14(1), pp104-106.

[8] Xiang-hua X., Guo-hua G., 2003, "*Method of Quickly Recognizing Vehicle Plate Based on Pattern Matching and Characteristic Dot Matching*", OptoelectronicTechnology, vol 23(4), pp 268-270.

[9] Cheng Y, Lu J., Yahagi T., 2004, "*Car license plate recognition based on the combination of principal components analysis and radial basis function networks*", Signal Processing, Proceedings. ICSP '04. 7th International Conference , vol.2, pp. 1455-1458

[10] Dongxiao N., Zhihong G., Xing M. 2007, "*Research on Neural Networks Based on Culture Particle Swarm Optimization and Its Application in Power Load Forecasting*" Natural Computation, ICNC 2007. Third International Conference, vol.1, pp. 270-274.

[11] Sarlashkar, A.N.; Bodruzzaman, M.; Malkani, M.J. 1998, "*Feature Extraction Using Wavelet Transform For Neural Network Based Image Classification*" System Theory, Proceedings of the Thirtieth Southeastern Symposium on 1998, pp. 412-416

[12] Tang, Y. Y., Yang, L.H., Liu, J., and Ma, H., 2000, *Wavelet Theory and Its Application to Pattern Recognition.* World Scientific Publishing Co. Pte. Ltd.

[13] Bow, S. T. 1992, *Pattern Recognition and Image Processing.* Marcel-Dekker, Marcel-Dekker: New York.

[14] Graps. A. "*An introduction to wavelets*" Computational Science & Engineering, IEEE, vol.2 , Issue 2, pp. 50-61, 1995

[15] Huang Wei, Lu Xiaobo, Ling Xiaojing, 2005, *"Wavelet packet based feature extraction and recognition of license plate characters"*, Chinese Science Bulletin Vol. 50, No. 2, pp. 97-100

[16] Lee, S.W., Kim, C. H., Ma, H. et al. 1996, *Multiresolution recognition of unconstrained handwritten numerals with wavelet transform and multi-layer cluster neural network*, Pattern Recognition, Vol. 29 (12), pp. 1953-1961.

[17] Choi E., Lee J., Yoon J., 2006, *"Feature Extraction for Bank Note Classification Using Wavelet Transform"* IEEE Pattern Recognition, ICPR 2006. 18th International Conference, Vol. 2, pp. 934 - 937

[18] Teknomo, K., 2006, K-Nearest Neighbors Tutorial. http:\\people.revoledu.com\kardi\tutorial\KNN\

[19] Duda R. O., Hart P. E., 1973, *Pattern Classification and Scene Analysis*. John Wiley & Sons.

[20] P. Indyk and R. Motwani, 1998, *Approximate nearest neighbors: towards removing the curse of dimensionality.* In *STOC*, pages 604–613.

[21] Samir Brahim B., N. Nourain, 2010, *"Statistical Model and Wavelet Function for Face Recognition"*, First International Conference on Fundamental and Applied Sciences, Kuala Lumpur.

[22] Shi, X., Zhao, W. and Shen, Y., *"Automatic License Plate Recognition System Based on Color Image Processing"*, International Conference on Computational Science and Its Applications 2005, pp. 1159-1168

[23] Ramsey FL, Schafer DW, 2002, *The Statistical Sleuth, a course in methods of data analysis.* Duxbury Learning Publishing.

[24] S. Prabakaran, R. Sahu, S.Verma, *"An analysis of Microarray data using Wavelet Power Spectrum"*, Engineering Letters, Vol. 13, Issue 3, EL_13_3_17

[25] Rafael C. G., Richard E. W., 2010, *Digital Image Processing*, Third Edition, Pearson Prentice Hall, Pearson Education, Inc.

[26] Samir Brahim B., 2009, *"Fast and Accuracy Control Chart Pattern Recognition using a New cluster-k-Nearest Neighbor"*, World Academy of Science, Engineering and Technology, Issue 49, pp. 1022-1026.

APPENDICES

APPENDIX A

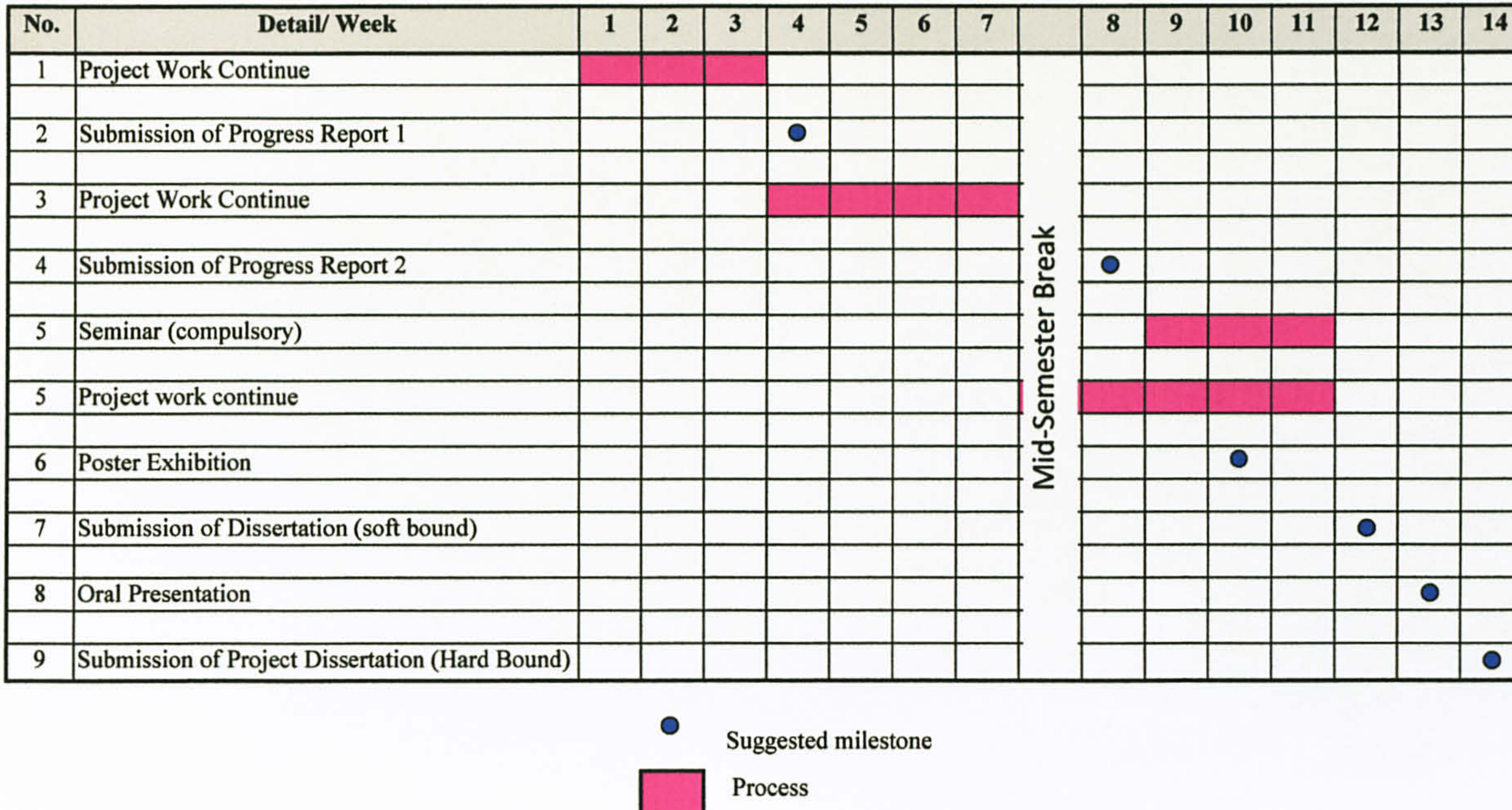
GANNT CHART FOR FYP 1

No.	Detail/ Week	1	2	3	4	5	6	7		8	9	10	11	12	13	14
1	Selection of Project Topic	■	■						Mid-semester break							
2	Preliminary Research Work		■	■	■											
3	Submission of Preliminary Report				●											
4	Seminar 1 (optional)					■	■	■								
5	Project Work					■	■	■								
6	Submission of Progress Report									●						
7	Seminar 2 (compulsory)										■	■	■	■		
8	Project work continues									■	■	■	■	■		
9	Submission of Interim Report Final Draft														●	
10	Oral Presentation															●

● Milestone
■ Process

APPENDIX B

GANNT CHART FOR FYP 2



APPENDIX C

IMAGE DATABASE

Class 1 images

Class 2 images

Class 3 images

Class 4 images

APPENDIX D

WAVELET FEATURES EXTRACTION

MATLAB Code
level5.m

```
threshold_v=0.7
save('data1','threshold_v');%clear all the variables except data_all
clear all;
load data1;
delete('*.txt');
Image_number=80
for images=1:Image_number
h = fspecial('unsharp',0.5);
str = strcat(int2str(images));
eval('imge=imread(str);');
imge=rgb2gray(imge);

imge = filter2(h,imge)/255;
size(imge)
imge=imge/max(max(max(imge)));
images
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%wave transformation
[C,S]=wavedec2(double(imge),5,'sym8'); % sym8 or dB1 wavelet

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%APPROXIMTION
cA5 = appcoef2(C,S,'sym8',5);
cA4 = appcoef2(C,S,'sym8',4);
cA3 = appcoef2(C,S,'sym8',3);
cA2 = appcoef2(C,S,'sym8',2);
cA1 = appcoef2(C,S,'sym8',1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cA1);
cA11=sort(reshape(cA1,1,n1*m1),'descend');
[n1 m1]=size(cA2);
cA21=sort(reshape(cA2,1,n1*m1),'descend');
[n1 m1]=size(cA3);
cA31=sort(reshape(cA3,1,n1*m1),'descend');
[n1 m1]=size(cA4);
cA41=sort(reshape(cA4,1,n1*m1),'descend');
[n1 m1]=size(cA5);
cA51=sort(reshape(cA5,1,n1*m1),'descend');
cA_ALL=[cA11 cA21 cA31 cA41 cA51];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HORIZONTAL
cH5 = detcoef2('h',C,S,5);
cH4 = detcoef2('h',C,S,4);
cH3 = detcoef2('h',C,S,3);
cH2 = detcoef2('h',C,S,2);
cH1 = detcoef2('h',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cH1);
```

```
cH11=sort(reshape(cH1,1,n1*m1),'descend');
[n1 m1]=size(cH2);
cH21=sort(reshape(cH2,1,n1*m1),'descend');
[n1 m1]=size (cH3);
cH31=sort(reshape(cH3,1,n1*m1),'descend');
[n1 m1]=size (cH4);
cH41=sort(reshape(cH4,1,n1*m1),'descend');
[n1 m1]=size(cA5);
cH51=sort(reshape(cH5,1,n1*m1),'descend');
cH_ALL=[cH11 cH21 cH31 cH41 cH51];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VERTICAL
cV5 = detcoef2('v',C,S,5);
cV4 = detcoef2('v',C,S,4);
cV3 = detcoef2('v',C,S,3);
cV2 = detcoef2('v',C,S,2);
cV1 = detcoef2('v',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cV1);
cV11=sort(reshape(cV1,1,n1*m1),'descend');
[n1 m1]=size(cV2);
cV21=sort(reshape(cV2,1,n1*m1),'descend');
[n1 m1]=size(cV3);
cV31=sort(reshape(cV3,1,n1*m1),'descend');
[n1 m1]=size(cV4);
cV41=sort(reshape(cV4,1,n1*m1),'descend');
[n1 m1]=size(cV5);
cV51=sort(reshape(cV5,1,n1*m1),'descend');
cV_ALL=[cV11 cV21 cV31 cV41 cV51];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DIAGONAL
cD5 = detcoef2('d',C,S,5);
cD4 = detcoef2('d',C,S,4);
cD3 = detcoef2('d',C,S,3);
cD2 = detcoef2('d',C,S,2);
cD1 = detcoef2('d',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cD1);
cD11=sort(reshape(cD1,1,n1*m1),'descend');
[n1 m1]=size(cD2);
cD21=sort(reshape(cD2,1,n1*m1),'descend');
[n1 m1]=size(cD3);
cD31=sort(reshape(cD3,1,n1*m1),'descend');
[n1 m1]=size(cD4);
cD41=sort(reshape(cD4,1,n1*m1),'descend');
[n1 m1]=size(cD5);
cD51=sort(reshape(cD5,1,n1*m1),'descend');
cD_ALL=[cD11 cD21 cD31 cD41 cD51];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Data_whole=[cD_ALL cV_ALL cH_ALL cA_ALL];
Data3=size(Data_whole)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if images==1 data_all=Data_whole;
else
    data_all=[data_all;Data_whole];
end

Data1=size(Data_whole)
```

```
Data2=size(data_all);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Features reduction and selection

[n1 n2]=size(data_all)
dimm=n2
size(data_all)
for i=1:4
   meann(i,:)=mean(data_all((1+(i-1)*20):(20*i),:));
   varr(i,:)=var(data_all((1+(i-1)*20):(20*i),:));
end
meann_size=size(meann)

vect_removeS=0;
for j=1:dimm
var_mod(j)=0;
min_var(j)=100^100;
for k=1:4
   if min_var(j)>=(meann(k,j)-mean(meann(:,j)))^2/(0.0000001+varr(k,j))
      min_var(j)=(meann(k,j)-mean(meann(:,j)))^2/(0.0000001+varr(k,j));
   end

end
var_mod(j)=min_var(j);
if var_mod(j)<threshold_v   %Insert defined threshold value
vect_remove=[vect_removeS j];
end
end
vect_removeS(1)=[];
data_all2=data_all;

data_all2(:,vect_removeS)=[];
data_all=data_all2;
eucledian_all;
```

APPENDIX E

EUCLIDEAN DISTANCE CLASSIFIER

MATLAB Code
euclidean_all.m

```
train=data_all;
one_core=mean(train(1:2:20,:));
two_core=mean(train(21:2:40,:));
three_core=mean(train(41:2:60,:));
four_core=mean(train(61:2:80,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
one=0;
two=0;
three=0;
four=0;

for i=1:80
dist1=sqrt(sum((one_core-train(i,:)).^2));
dist2=sqrt(sum((two_core-train(i,:)).^2));
dist3=sqrt(sum((three_core-train(i,:)).^2));
dist4=sqrt(sum((four_core-train(i,:)).^2));

   mindist=min([dist1 dist2 dist3 dist4]);

if mindist==dist1
       class = 'one';
         if 1<=i && i<=20 one=one+1;
         end
   elseif mindist==dist2
        class = 'two';
         if 21 <= i && i <= 40
            two=two+1;
         end
        elseif mindist==dist3
        class = 'three';
         if 41 <= i && i <= 60 three=three+1; end
        elseif mindist==dist4
        class = 'four';
        if 61 <= i && i <= 80 four=four+1; end
        end

end

accuracy=(one + two + three + four )/80
```

APPENDIX F

CLUSTER K-NEAREST NEIGHBORS CLASSIFIER

main_nnn_2.m

```
threshold_v=0.7
level5; % perform level 5 wavelet feature extraction from level5.m
save ('data','data_all'); % clear all the variables except data_all
save vect_remove vect_removeS % save variable vect_removeS
size(vect_removeS)

clear all;

load data;
accurrency=0.55;
class_number=4

accu=0;
h1=1;
Max_Accu=0;
for co=0.01:0.05:0.99
 % co
  for cor=0.1:0.1:0.99


    h1=1;
    coefficient=co*ones(class_number,1);
    coefficient_reduction=cor;
    while(accurrency<0.99 & h1<10)



    NNN4;
    [class_assig, accurrency ,erreur_position, time_spend_NN,
trust_coefficient,trust_coefficient2]=classify_control_data(repre_class,subclass_number,data_control,class_size_t-
class_size,size_subclass,1);

   % accurrency
     if accurrency>=Max_Accu
        Max_Accu=accurrency
        subclass_number
        save test repre_class subclass_number size_subclass ; %save the trained data in file name 'test'
        co_max=co
        cor_max=cor
     end

    h1=h1+1;
    accu=accu+accurrency;


    end
```

NN4.m

```
class_number=4;
class_size=[12 12 12 12];
class_size_t=[20 20 20 20];
dim=16;
%var1=sum(std(x')'.^2);
k=1;
control=0;
size_subclass=zeros(class_number,max(class_size));                    %%%%%% class_size become max class_size


hole_data=data_all;
%%%%%% paratger les donnees entre training data et control data

data_t=hole_data(1:class_size(1),:);                                 %%%%%%111  50 have to be change
for tr=1:(class_number-1)
   data_t=[data_t;hole_data(1+sum(class_size_t(1:tr)):class_size(tr+1)+sum(class_size_t(1:tr)),:)];
%%%%%%111  50 have to be change

end

%%%%%% it better to take 0.34 as begginer

%coefficient=0.6*ones(class_number,1);


data_control=hole_data(class_size(1)+1:class_size_t(1),:);
for tr=1:(class_number-1)

data_control=[data_control;hole_data(1+sum(class_size_t(1:tr))+class_size(tr+1):class_size_t(tr+1)+sum(class_si
ze_t(1:tr)),:)];
end

dim=size(data_t,2);

incream=1;
check_part=1;
while(check_part>0 & incream<10) % check that the corfficient is never negative
      var1=sum(sum((data_t-ones(sum(class_size),1)*mean(data_t)).^2));  %%%%%%111  class_zize have to be
change
       var2=0.5*var1;

 for cn=1:class_number
   control=0;
   x=data_t( sum(class_size(1:cn-1))+1:sum(class_size(1:cn)),:);
   var1=sum(sum((x-ones(class_size(cn),1)*mean(x)).^2));
   var_class_i(cn)=var1;
   var1=2*var1;
   %%%%%%%%%%%%%%%%%%%%%%
   for k=1:(class_size(cn)-1)
      [y,b,c]=kMeansCluster(x,k,1);
   %%%%%%%%%%%%%%%%%%%%%%
      size_subclass(cn,:)=zeros(1,max(class_size));
      for n=1:k
         [Rf,Cf,Vo]=find(b==n);
         size_subclass(cn,n)=sum(b==n);
       contro_var(n)=sum(sum((x(find(b==n),:)-ones(sum(Vo),1)*c(n,:)).^2));
      end
```

```
        var2=var1;
        var1=sum(contro_var(1:k));
        var_list(k)=var1;
        variation_var_list(cn,k)=abs(((var1-var2)/max(var2,var1)));
        if abs((var1-var2)/max(var2,var1))<coefficient(cn)
            control=control+1;
        end
        if abs((var1-var2)/max(var2,var1))>= coefficient(cn)   % choose the treashold
            control=0;
        end

       if (control>=2| var1==0)  % better to take control>=2
          cn;
          k;
          break
       end
    end
     subclass_number(cn)=k;

       var_class_f(cn)=var1;
   if cn==1
      diction_class=[y,cn*ones(class_size(cn),1)];
      repre_class=c;

   else
   diction_class=[diction_class;y,cn*ones(class_size(cn),1)];
   repre_class=[repre_class;c];

   end


  end
  [class_assig,
occur,erreur_position,time_spend_NN,trust_coefficient,trust_coefficient2]=classify_control_data(repre_class,subc
lass_number,data_t,class_size,size_subclass,1);
  class_aff=ones(1,class_size(1));
  for p=2:class_number
    class_aff=[class_aff, p*ones(1,class_size(p))];
  end
  [r c v]=find(class_assig~=class_aff);
  if sum(v)~=0
    erreur_location=unique(class_aff(c));
   coefficient(erreur_location)=coefficient_reduction*coefficient(erreur_location);
  end

  check_part=sum(v);

  incream=incream+1;
end
```

classify_control_data.m

```
function [class_assig,trust_coefficient,trust_coefficient2]=
classify_control_data(repre_class,subclass_number,data_control,class_size_C,size_subclass,n);

 m1=sum(subclass_number);
[ert class_number]=size(subclass_number);
[control_number dim]=size(data_control);

for ind=1:control_number
   if n==1
         size(repre_class)
         size(data_control)
         erreur=sum(((repre_class-ones(m1,1)*data_control(ind,:)).^2)');
        [val posi1]=min(erreur);
        posi=posi1;
        for l=1:class_number
          if posi-subclass_number(l)<=0
            class_assig(ind)=l;
            break
          end
            posi=posi-subclass_number(l);
        end

              [value_min posi_min]=sort(erreur);

              for ind_t=2:m1
                  posi_min2=posi_min(ind_t);
                  for l3=1:class_number
                     if posi_min2-subclass_number(l3)<=0
                       secd_posi=l3;
                       break
                     end
                    posi_min2=posi_min2-subclass_number(l3);
                  end
                  if secd_posi~=class_assig(ind)
                    break
                  end
              end
trust_coefficient(ind)=erreur(posi1)/erreur(posi_min(ind_t));

             xp=posi1-sum(subclass_number(1:l-1));
             N_subclass_min=xp;
             xp=posi_min(ind_t)-sum(subclass_number(1:(l3-1)));
             N_subclass_min2=xp;
           trust_coefficient2(ind)=size_subclass(l3,N_subclass_min2)/size_subclass(l,N_subclass_min);
    end

    if n==2
       erreur=sum( ( ((repre_class-ones(m1,1)*data_control(ind,:)).^2)./ (max(0.1*abs(repre_class),
0.1*abs(ones(m1,1)*data_control(ind,:))).^2) )' );
       [val posi]=min(erreur);
       for l=1:class_number
         if posi-subclass_number(l)<=0
           class_assig(ind)=l;
           break
         end
         posi=posi-subclass_number(l);
       end
```

```
    end
%stop time
end
%average of stop time
occur=0;

class_number
```

APPENDIX G

MAIN PROGRAM

TEST.m

```
load vect_remove
load test
testinglevel5
[class_assig,
trust_coefficient,trust_coefficient2]=classify_control_data(repre_class,subclass_number,data_all,size(data_all,1),s
ize_subclass,1)
confedence1=1-trust_coefficient2
```

testinglevel5.m

```
load test; % Load the saved trained data
X=imread('02.bmp'); % Read the input image
h = fspecial('unsharp',0.5);
imge=rgb2gray(X);

imge = filter2(h,imge)/255;
size(imge);
imge=imge/max(max(max(imge)));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%wave transformation
[C,S]=wavedec2(double(imge),5,'dB1');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%APPROXIMTION
cA5 = appcoef2(C,S,'dB1',5);
cA4 = appcoef2(C,S,'dB1',4);
cA3 = appcoef2(C,S,'dB1',3);
cA2 = appcoef2(C,S,'dB1',2);
cA1 = appcoef2(C,S,'dB1',1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cA1);
cA11=sort(reshape(cA1,1,n1*m1),'descend');
[n1 m1]=size(cA2);
cA21=sort(reshape(cA2,1,n1*m1),'descend');
[n1 m1]=size(cA3);
cA31=sort(reshape(cA3,1,n1*m1),'descend');
[n1 m1]=size(cA4);
cA41=sort(reshape(cA4,1,n1*m1),'descend');
[n1 m1]=size(cA5);
cA51=sort(reshape(cA5,1,n1*m1),'descend');
cA_ALL=[cA11 cA21 cA31 cA41 cA51];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% HORIZONTAL
cH5 = detcoef2('h',C,S,5);
cH4 = detcoef2('h',C,S,4);
cH3 = detcoef2('h',C,S,3);
cH2 = detcoef2('h',C,S,2);
cH1 = detcoef2('h',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cH1);
```

```
cH11=sort(reshape(cH1,1,n1*m1),'descend');
[n1 m1]=size(cH2);
cH21=sort(reshape(cH2,1,n1*m1),'descend');
[n1 m1]=size (cH3);
cH31=sort(reshape(cH3,1,n1*m1),'descend');
[n1 m1]=size (cH4);
cH41=sort(reshape(cH4,1,n1*m1),'descend');
[n1 m1]=size(cA5);
cH51=sort(reshape(cH5,1,n1*m1),'descend');
cH_ALL=[cH11 cH21 cH31 cH41 cH51];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%VERTICAL
cV5 = detcoef2('v',C,S,5);
cV4 = detcoef2('v',C,S,4);
cV3 = detcoef2('v',C,S,3);
cV2 = detcoef2('v',C,S,2);
cV1 = detcoef2('v',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cV1);
cV11=sort(reshape(cV1,1,n1*m1),'descend');
[n1 m1]=size(cV2);
cV21=sort(reshape(cV2,1,n1*m1),'descend');
[n1 m1]=size(cV3);
cV31=sort(reshape(cV3,1,n1*m1),'descend');
[n1 m1]=size(cV4);
cV41=sort(reshape(cV4,1,n1*m1),'descend');
[n1 m1]=size(cV5);
cV51=sort(reshape(cV5,1,n1*m1),'descend');
cV_ALL=[cV11 cV21 cV31 cV41 cV51];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DIAGONAL
cD5 = detcoef2('d',C,S,5);
cD4 = detcoef2('d',C,S,4);
cD3 = detcoef2('d',C,S,3);
cD2 = detcoef2('d',C,S,2);
cD1 = detcoef2('d',C,S,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[n1 m1]=size(cD1);
cD11=sort(reshape(cD1,1,n1*m1),'descend');
[n1 m1]=size(cD2);
cD21=sort(reshape(cD2,1,n1*m1),'descend');
[n1 m1]=size(cD3);
cD31=sort(reshape(cD3,1,n1*m1),'descend');
[n1 m1]=size(cD4);
cD41=sort(reshape(cD4,1,n1*m1),'descend');
[n1 m1]=size(cD5);
cD51=sort(reshape(cD5,1,n1*m1),'descend');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
cD_ALL=[cD11 cD21 cD31 cD41 cD51];

Data_whole=[cD_ALL cV_ALL cH_ALL cA_ALL];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

data_all=Data_whole;

Data1=size(Data_whole);
Data2=size(data_all);
```

```
[n1 n2]=size(data_all);
dimm=n2;
size(data_all);
size(data_all)
data_all(:,vect_removeS)=[];
size(data_all)
```