REAL-TIME VIDEO FOR ROAD LANE MARKER (URBAN TYPE) USING IMAGE PROCESSING

LIANG JIN CHUAN

A report submitted in partial fulfilment of the requirements for the degree of Bachelor of Mechatronics Engineering

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2018

"I hereby declare that I have read through this report entitled "**Real-Time Video for Road Lane Marker (Urban Type) Using Image Processing**" and found that it complies the partial fulfilment for awarding the degree of Bachelor of Mechatronics Engineering.

Signature : ....

Supervisor's Name : Zamani Bin Md. Sani

Date : ....

I declare that this report entitled “**Real-Time Video for Road Lane Marker (Urban Type) Using Image Processing**” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently in candidature of any other degree.

Signature : ....

Name : LIANG JIN CHUAN

Date : ....

To my beloved mother and father

ACKNOWLEDGEMENT

First, I would like to express my deepest gratitude and infinite appreciation to my main project supervisor, Mr Zamani Bin Md. Sani, for guidance critics, encouragement and friendship during my FYP. I also feel thankful to him for giving comment and feedback to me during the draft report submission. Without his continued support and interest, this FYP would not have been same as presented here.

At the same time, I also wanted to thank my FYP panels, Miss Nursabillilah Binti Mohd. Ali and Miss Irma Wani Binti Jamaludin, for giving feedback and suggestion during my FYP presentation. Besides, I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for funding and providing the relevant literatures.

Last but not least, I wanted to thank my course mates as they offered their knowledge and opinion to me when I had problems in my project. My postgraduate friend should also be recognised for his support. I am also grateful to all my family members for their care and love which is motivating me to complete my FYP during the entire period.

ABSTRACT

Road markers provide road information to driver to ensure road and their safety. Different type of markers indicate different kind of information. For example, double lane marker indicates that drivers are not allowed to overtake due to dangerous road condition. It can lead to road accident if the drivers do not follow the rule of the road marker or the road markers are not seen clearly by the drivers. Nowadays, road accidents become a concerning issue all over the world. In order to avoid the tragedy continues to happen, vision based system is developed to detect and classify those markers. Besides, the existing features are insufficient to allow a robust recognition. In this report, real-time video of three types of marker is developed using image processing and artificial neural network (ANN). The video is recorded in urban area at afternoon. The image processing and classification are done using MATLAB software. The image processing techniques include grayscale conversion, image sharpening, noise removal, cropping, resizing and feature extraction. The features selected are the length of the feature vector on HOG and LBP. These feature vectors are later to be trained using neural network pattern recognition tool, to classify the data getting from the feature. The validation for the sample is 15% while the testing for the sample is 15% and the rest is used for training. The result shows an accuracy of 99.4% with HOG and LBP features as input vectors.

ABSTRAK

Penanda jalan memberikan maklumat jalan kepada pemandu untuk memastikan keselamatan mereka dan keamanan jalan raya. Penanda yang berbeza menunjukkan maklumat yang berlainan. Sebagai contoh, penanda yang mempunyai dua barisan menunjukkan bahawa pemandu tidak dibenarkan memotong kerana berada di kawasan yang merbahaya. Hal ini dapat mengakibatkan kemalangan jalan raya jika pemandu tidak mengikuti peraturan penanda ataupun pemandu tidak dapat melihat penanda jalan dengan jelas. Kini, kemalangan jalan raya menjadi isu yang penting di seluruh dunia. Untuk mengelakkan tragedy terus berlaku, sistem berasaskan visi telah dibangunkan untuk mengesan dan mengklasifikasikan penanda-penanda tersebut di jalan raya. Selain itu, terdapat ciri-ciri yang tidak mencukupi untuk membolehkan pengiktirafan yang teguh. Dalam laporan ini, video dalam masa nyata dengan tiga jenis penanda jalan telah dilakukan penyelidikan menggunakan pemprosesan imej dan jaringan saraf tiruan (ANN). Video telah dirakamkan di kawasan bandar pada masa tengah hari. Pemprosesan imej dan klasifikasi pula dilakukan dengan menggunakan perisian MATLAB. Teknik pemprosesan imej termasuk penukaran warna kepada kelabu, pengasahan imej, penyingkiran hingar, pemotongan imej, saiz semula imej dan pengekstrakan ciri. Ciri-ciri yang dipilih adalah kepanjangan vektor ciri pada HOG dan LBP. Vektor ciri ini kemudiannya akan dilatih menggunakan alat rangkaian saraf dalam MATLAB iaitu alat pengenalan corak rangkai neural, untuk mengklasifikasikan data yang diperoleh dari ciri tersebut. Pengesahan sampel adalah 15% manakala ujian bagi sampel adalah 15% dan selebihnya digunakan untuk proses perlatihan. Hasil daripada proses perlatihan menunjukkan ketepatan 99.4% dengan HOG dan LBP.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

FYP – Final Year Project

UTeM – Universiti Teknikal Malaysia Melaka

ANN – Artificial Neural Network

ROI – Region of Interest

FYP- Final Year Project

ADAS – Advanced Driver-Assistance Systems

MIROS – Malaysian Institute of Road Safety Research

GPS – Global Positioning System

IPM – Inverse Perspective Mapping

RPROP – Resilient Back Propagation

HOG – Histogram of Oriented Gradients

POI – Point of Interest

MLP – Multi-Layer Perceptron

USB – Universal Serial Bus

SSD – Solid State Drive

HDD – Hard Disk Drive

LBP – Local Binary Pattern

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

In Malaysia, road accident happens every year and has become a concerning issue. The statistic on the road accidents in Malaysia increases from 2014 to 2016. Based on the road accident data in Malaysia, there is a relation between the population and the death rate. The amount of road deaths increases from year to year due to the growth population in Malaysia [1].

Road marker is a kind of hint or indicator that is used on a road surface in order to give information. They are widely placed with the road marking machines. Besides, they are applied in other facilities used by vehicles to mark parking spaces or designate areas for other purposes. It is also used to show regulation for parking and stopping.

They are used to control and guide the traffic. Besides, they play an important role on urban roads because they ensure the road safety and make the flow of travel paths smooth. They are also used on roadways to provide guidance to the road users such as drivers and pedestrians. Different types of marker indicates different kind of information to minimize the confusion and uncertainty about their meaning. For example, most common types of road markers are single dash line and double solid lines. Single dash line road marker indicates that overtaking of vehicles is allowed whereas double solid line road marker indicates that overtaking of vehicles is not allowed.

Today, road markers are used to send and give information to the driver spanning navigational, safety and enforcement issues leading to their use in road environment understanding within advanced driver-assistance systems and implementation for future use in autonomous road vehicles.

Year	Registered Vehicles	Population	Road Crashes	Road Deaths	Serious Injury	Slight Injury	Index per 10,000 Vehicles	Index per 100,000 Population	Indeks per billion VKT
1997	8,550,469.00	21,665,600.00	215,632.00	6,302.00	14,105.00	36,167.00	7.37	29.10	33.57
1998	9,141,357.00	22,179,500.00	211,037.00	5,740.00	12,068.00	37,896.00	6.28	25.80	28.75
1999	9,929,951.00	22,711,900.00	223,166.00	5,794.00	10,366.00	36,777.00	5.83	25.50	26.79
2000	10,598,804.00	23,263,600.00	250,429.00	6,035.00	9,790.00	34,375.00	5.69	26.00	26.25
2001	11,302,545.00	23,795,300.00	265,175.00	5,849.00	8,680.00	35,944.00	5.17	25.10	23.93
2002	12,068,144.00	24,526,500.00	279,711.00	5,891.00	8,425.00	35,236.00	4.90	25.30	22.71
2003	12,819,248.00	25,048,300.00	298,653.00	6,286.00	9,040.00	37,415.00	4.90	25.10	22.77
2004	13,828,889.00	25,580,000.00	326,815.00	6,228.00	9,218.00	38,645.00	4.52	24.30	21.10
2005	15,026,660.00	26,130,000.00	328,264.00	6,200.00	9,395.00	31,417.00	4.18	23.70	19.58
2006	15,790,732.00	26,640,000.00	341,252.00	6,287.00	9,253.00	19,885.00	3.98	23.60	18.69
2007	16,813,943.00	27,170,000.00	363,319.00	6,282.00	9,273.00	18,444.00	3.74	23.10	17.60
2008	17,971,907.00	27,730,000.00	373,071.00	6,527.00	8,868.00	16,879.00	3.63	23.50	17.65
2009	19,016,782.00	28,310,000.00	397,330.00	6,745.00	8,849.00	15,823.00	3.55	23.80	17.27
2010	20,188,565.00	28,910,000.00	414,421.00	6,872.00	7,781.00	13,616.00	3.40	23.80	16.21
2011	21,401,269.00	29,000,000.00	449,040.00	6,877.00	6,328.00	12,365.00	3.21	23.70	14.68
2012	22,702,221.00	29,300,000.00	462,423.00	6,917.00	5,868.00	11,654.00	3.05	23.60	13.35
2013	23,819,256.00	29,947,600.00	477,204.00	6,915.00	4,597.00	8,388.00	2.90	23.10	12.19
2014	25,101,192.00	30,300,000.00	476,196.00	6,674.00	4,432.00	8,598.00	2.66	22.00	10.64
2015	26,301,952	31,190,000	489,606	6,706	4,120	7,432	2.55	21.5	9.6
2016	27,613,120	31,660,000[e]	521466[a]	7152[a]	NA	NA	2.59	22.6	NA

e = estimated value from Department of Statistics Malaysia
a = media statement
NA = Not available (The official figures are not available yet)

Figure 1.1: General Road Accident Data in Malaysia (1997-2016)[1]

There are several causes leading to the road accidents. For example, human error, condition of the road and vehicle problem [27][28]. According to Transport Minister Datuk Seri Liow Tiong Lai, a total of 7152 people died in road accidents in Malaysia in the year 2016 and he believed that 80.6% of the road accidents are caused by human error [2]. This was proven when a research by the MIROS shows that the main reason of road accidents was caused due to people driving recklessly and ignoring the traffic rules. Even though there is law enforcement and camera installation on the road to punish and fine those who break the rules, this incident still often happen. In order to reduce this tragedy continues happening, researchers have been conducting research towards the autonomous industry.

1.1 Motivation

Road accidents caused almost 40,000 people died during 2008 in European Union [31]. Dropping of death rate for over 30% from 2001, the European Transport Commission is working to revise and update the goals proposed in 2001 [32]. The aim is to reduce the number of accidents.

From the automotive industry, several development have been done in the last years. The cruise control (CC) that allows the driver to set a speed driving or its extension to the adaptive cruise control (ACC) where the vehicle is able to follow a leading car in highways are two of the most popular advanced driving assistance system (ADAS) developed by the car manufacturers to make the driving task easier and convenient.

The era for the future is for there to be an improvement from simple driving aids to automatic driving controls. Among all the different solutions to these problems that have been proposed, the advances of autonomous vehicles is currently an open field of research. Research on developing automated vehicles to improve the road safety and efficiency of road is indeed one of the most popular studied topics in the field of intelligent transportation systems (ITS) [30].

In this case, vision based system for road markers classification could be developed and implemented on the vehicle to assist the driver when they are driving. Road marker recognition and classification is a crucial research for people in the application and implementation of autonomous vehicle system. It is a system attached to a vehicle installing a camera to recognize the types of road marker. The road markers which mark the lane will help the drivers to avoid any accidents and keep track on the road within the detected markers.

Speaking of autonomous vehicle, the concept was first exposed at GM's Futurama exhibit at the World's Fair in 1939. Autonomous vehicle can be defined as self-driving car, robotic car or large sized unmanned ground vehicle that has the ability to navigate itself according to its environmental issues without the assistance of human. Autonomous vehicles use many kinds of techniques to sense their environment. Some of the examples are radar, GPS, computer vision and etc. In 1977, there was first autonomous vehicles in Japan which used camera and analogue computing to process the signals. However, in the year of 1987, the research project had been directed from signal based technologies toward vision-based technologies. In 2004, first Grand

Challenge was held and there were 15 autonomous cars competing in 150-mile challenge to promote the development of autonomous cars [29][33][34]. After six years of time, Google had launched a Google Driverless Car program to rack up more than 140,000 miles with a fleet of autonomous Toyota Prius hybrids. One year later, the state of Nevada approved the first law of allowing driverless cars whereas other nations begun to introduce the legislation law.

Apart from that, according to the statistical report from Boston consulting group which is shown in Figure 1.2, there would be a huge demand of autonomous cars in the year 2035.

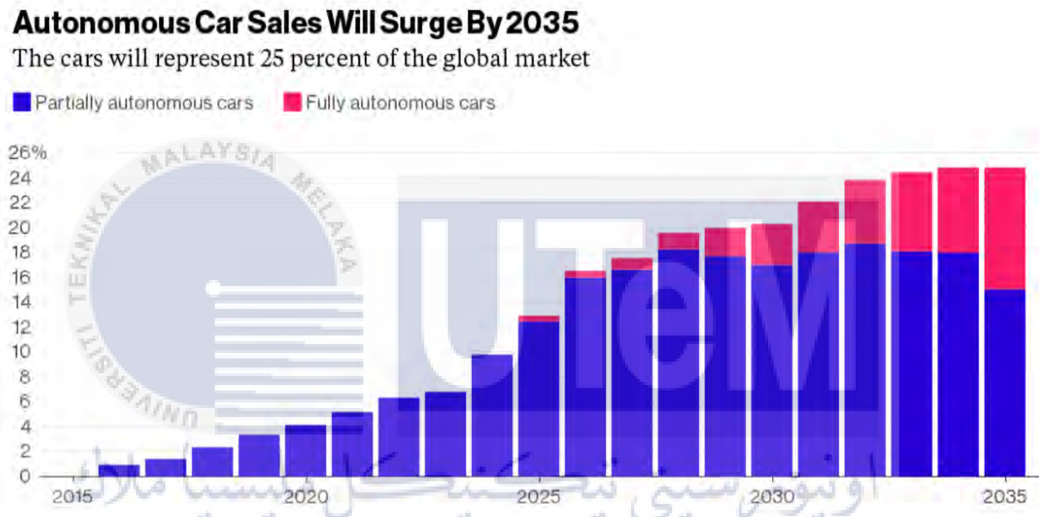

Figure 1.2: Statistical report of autonomous cars sales

In order to achieve the target of fully autonomous industry, the system applied to the autonomous cars are required to be improved. This is the reason and the motivation that is decided to develop a real-time video system of road lane marker.

1.2 Problem Statement

Many people are killed in car accidents each year. According to the statistical report on car accidents in Malaysia, the cases of road deaths increased from year 2014 to 2016. Most of these accidents are caused by the drivers carelessness, road condition and vehicle defection. Besides, the bad condition of the road markers also confused the drivers' mind as they did not know the type of the road markers [1]. Therefore, a vision based system is required to overcome the problem in order to reduce the rate of road accidents in Malaysia. The existing recognition system is not accurate as the existing features are insufficient to allow a robust recognition [16][20][21][22][23].

1.3 Objectives

The objective of this project is to:

i. To classify single dash line, double solid line and double dash line road markers on the urban road using neural network method.

ii. To develop a real-time recognition using image processing method.

iii. To analyse and improve the classification accuracy getting from the video using MATLAB software.

1.4 Scope

This research is focused on three types of road markers which are double solid line, double dash line and single dash line. An USB camera, Logitech C310, was used to capture the real time video. The video was recorded at Jalan Bukit Beruang in Ayer Keroh, Melaka, at the time of afternoon from 2.00 pm to 2.30 pm which had good illumination. The vehicle speed will be fixed around 40 km/h to 60 km/h to prevent crashing on the video captured due to the vibration while recording. Any shadows that causing the glaring effect will not be used in the training process.

a: Double solid line

b: Double dash line

c: Single dash line

Figure 1.3: Types of road marker (a), (b) and (c)

CHAPTER 2

LITERATURE REVIEW

2.1 Theory and Basic Principles

In this research, most theory and principles will be covered on machine vision and image processing. Machine vision is a system that recovers useful information from two-dimensional projections. The recovery requires inversion of a many-to one mapping. Therefore, knowledge about the scene and geometry of projection are needed for recovering the information. Machine vision has been now used widely for various applications such as advanced driver-assistance systems (ADAS) and industry purposes [3]. For example, Figure 2.1 shows the application of ADAS.

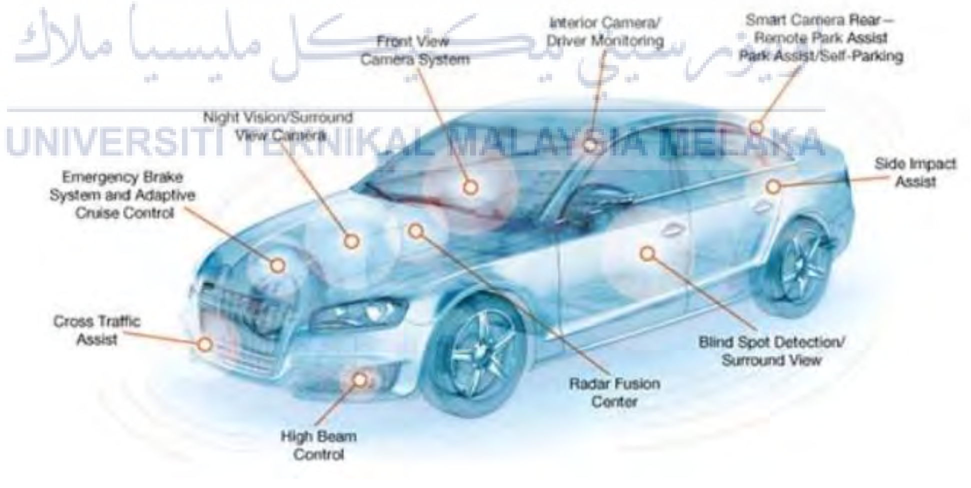

Figure 2.1: ADAS application

As we can see, due to various application of machine vision in different fields, different techniques for recovering information from images have been discovered.

One of the field which is related closely to machine vision is image processing. Image processing processes the original image into another image to retrieve the desired information easier [3]. An image can be defined as a 2D function, f(x,y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x,y) is called the intensity or gray level of the image at that point. A digital image is formed when the amplitude values of f, x and y are all finite. These finite number of elements which has particular elements and value are referred as image elements or pixels. Pixel is the term to denote the digital image elements. These pixels form an original image and consists of varied intensity [4]. For example, in colour image processing, a colour is usually represented by 3 components which are red, green and blue.

Generally, there are many types of image processing techniques included in this field. For examples, image enhancement, image segmentation, feature extraction and image classification [5].

2.1.1 Image Enhancement

In digital image pre-processing, image data that collected from the sensors restrain errors related to geometry and pixels intensity value. Appropriate mathematical models are used to correct these errors. Image enhancement is the method that modify the image by adjusting the pixel intensity values to enhance its visual effect. Image enhancement involves many techniques to improve the image or convert it to be suitable for human or machine interpretation. Sometimes images obtained from camera can be lacked in contrast and brightness due to illumination condition when capturing images. These images are considered having different kind of noise. Therefore, image enhancement aims to accentuate certain image features for subsequent analysis or for image display [12]. For example, some of the enhancement techniques are contrast stretching, noise filtering and histogram modification.

2.1.1.1 Constrast Stretching

Some images are homogeneous. They do not have any difference in their levels. In terms of histogram representation, they are characterized as the occurrence of very narrow peaks. The homogeneity can be due to the illumination condition [11]. Thus, the images are difficult for interpretation because only a narrow range of gray-levels in image having provision for wider range of gray-levels. In this case, contrast stretching can be used to stretch the narrow range to a new dynamic range.

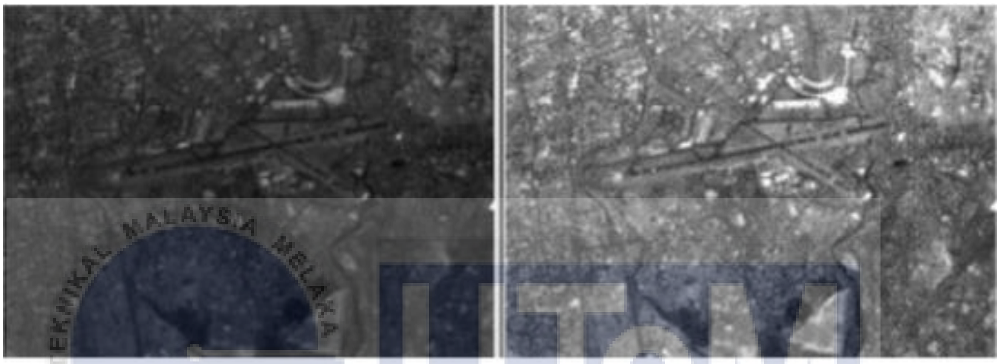

Figure 2.2: Contrast Stretching [11]

Assume an image I with an intensity value range [0,1], the energy functional is established:

$$E(I) = 1/2\ (I(x, y) - 1/2\ )\ 2dxdy - 1/4\ |I(x, y) - I(u, v)|dxdydudv \quad (1)$$

The above equation is solved using the steepest descent with an auxiliary variable t as follows:

$$\frac{\partial I(x,y,t)}{\partial t} = [1 - \frac{I(x,y,t)}{Imax}]A\Omega \; - \; A(I(x,y,t)) \quad (2)$$

where AΩ is the image area and A(·) is the function of the area. Suppose that the above equation and all other evolutionary equations have initial conditions:

$$I(x, y, 0) = I0(x, y) \quad (3)$$

and Neumann boundary conditions:

$$\frac{\partial I}{\partial n} = 0 \quad (4)$$

where n is the orientation perpendicular to the boundary. Equation (2) has a unique steady-state solution:

$$I(x, y, \infty) = Imax \times H(I) \quad (5)$$

where H(I) is the cumulative distribution histogram of image I. This solution is the result of traditional HE method for the image. More generally, the equation (2) can be rewritten as follows [24]:

$$\partial I \, \frac{(x,y,t)}{\partial t} = f(I(x,y,t)) - I(x,y,t) \tag{6}$$

where f(I(x, y, t)) is an arbitrary image enhancement transformation.

2.1.1.2 Noise Filtering

Noise filtering is used to filter the unnecessary or undesired information from an image. It also removes noises from the images. There are many types of filters such as low pass, high pass, mean, median and so on [11].

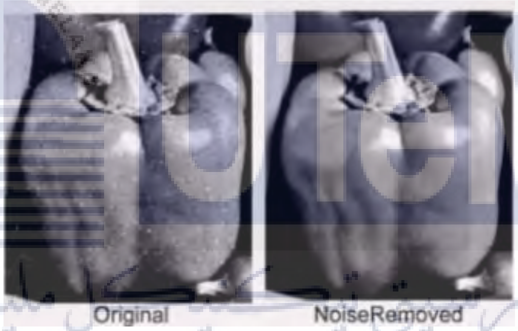

Figure 2.3: Noise Filtering [11]

There is statistical methods for image filtering, though they are infrequently used as they are computationally demanding. For Gaussian noise, it can model the pixels in a greyscale image as auto-normally distributed, where each pixel's "true" greyscale value is normally distributed with mean equal to the average greyscale value of its neighbouring pixels and a given variance.

Let δ_i denote the pixels adjacent to the *i*th pixel. Then the conditional distribution of the greyscale intensity (on a [0,1] scale) at the *i*th node is:

$$P(x(i) = c|x(j)\forall j \in \delta i) \propto e^{-\frac{\beta}{2\lambda}\sum_{j\epsilon\delta i}(c-x(j))^2} \tag{7}$$

for a chosen parameter ß ≥ 0 and variance λ. One method of noise filtering that uses the auto-normal model uses the image data as a Bayesian prior and the auto-normal density as a likelihood function, with the resulting posterior distribution offering a mean or mode as a noise removed image [25].

Besides, there is other noise filtering method which is the combination of salt and pepper noise and median filtering. Salt-and-pepper noise is a common noise that can usually be seen on images. It is also known as impulse noise. Sharp and sudden disturbances in the image signal can cause this type of noise. The noise can be seen as sparsely occurring white and black pixels. An effective removal method for this type of noise is a median filter. Median filter is a non-linear spatial filter. It is commonly used because it is very effective at removing noise while preserving edges. The median is calculated by first sorting all the pixel values from the window into numerical order, and then replacing the pixel being considered with the middle (median) pixel value. If window elements are even, the middle is computed using the mean of the adjacent two pixels [37].

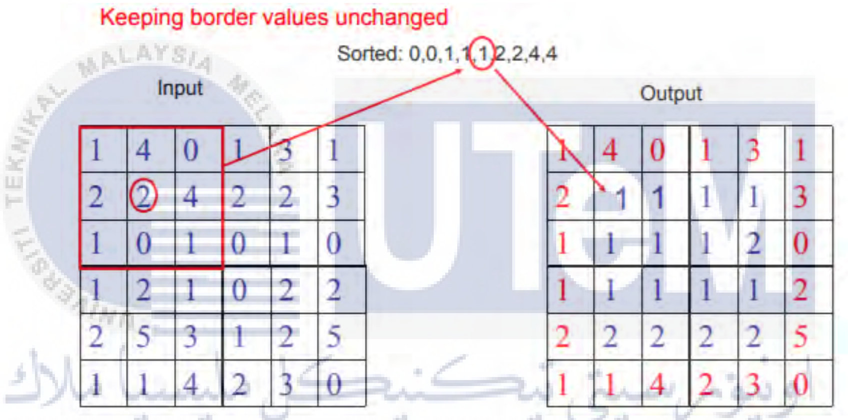

Figure 2.4: 2D Median filtering using a 3 x 3 sampling window

2.1.1.3 Histogram Modification

Histogram plays an important roles in image enhancement. It shows the image traits and characteristics. The traits and characteristics of image can be edited and modified via modifying the histogram. One of the method is Histogram Equalization. This method is a non-linear stretch that redistributes pixel values so that the pixels are identical with each other within a range. The result approximates a flat histogram which contrast is increased at the peaks and decreased at tails [11].

Consider a discrete grayscale image {x} and let n_i be the number of occurrences of gray level i. The probability of an occurrence of a pixel of level i in the image is

$$p_x(i) = p(x = i) = \frac{n_i}{n}, 0 \leq i \leq L \quad (8)$$

L being the total number of gray levels in the image (typically 256), n being the total number of pixels in the image and $p_x(i)$ being the image's histogram for pixel value i, normalized to [0,1].

Cumulative distribution function is defined corresponding to p_x as

$$cdf_x(i) = \sum_{j=0}^{i} p_x(j) \qquad (9)$$

Which is also the image's accumulated normalized histogram.

A transformation of the form $y = T(x)$ is created to produce a new image {y}, with a flat histogram. Such an image would have a linearized cumulative distribution function (CDF) across the value range,

$$cdf_y(i) = iK \qquad (10)$$

For some constant K, the properties of the CDF allow us to perform such a transform which is defined as

$$cdf_y(y') = cdf_y\big(T(k)\big) = cdf_x(k) \qquad (11)$$

Where k is in the range [0, L]). Notice that T maps the levels into the range [0, 1], since a normalized histogram of {x} is used. In order to map the values back into their original range, the following simple transformation needs to be applied on the result which is shown as [35][36]:

$$y' = y \cdot (\max\{x\} - \min\{x\}) + \min\{x\} \qquad (12)$$

Figure 2.4 showed the visualization of image for before equalization and after equalization.

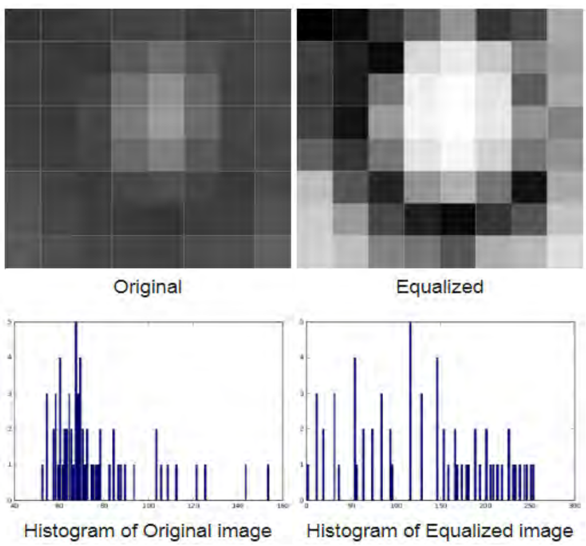

Figure 2.5: Histogram equalization of image [35]

2.1.2 Image Segmentation

Image segmentation is a process that divides an image into several parts or its constituent parts. The problem being solved will decide the level of the segmentation carried out. The segmentation process stops when the region of interest (ROI) is done selected. For example, in this road lane markers research, our region of interest lies on the road lane, which means the first step to be carried out is to segment the road from the image, neglecting the other textures such as trees and building and then to segment the contents of the road down to the road lane markers. In this case, image thresholding techniques are used for image segmentation.

After done thresholding the image, a binary image will be formed where all the object pixels have one gray level whereas the background pixels have another gray level. The gray level can be either 'black' or 'white'. The best threshold for an image segmentation is to select the object pixels and maps them to 'black' to make it easier for processing. Many kind of automatic thresholding techniques have also been proposed. Generally, thresholding can be defined as mapping of the grayscale into the binary set {0,1}:

$$S(x,y) = \begin{cases} 0, if \ g(x,y) < T(x,y) \\ 1, if \ g(x,y) \geq T(x,y) \end{cases} \tag{13}$$

where S(x, y) is the value of the segmented image, g(x, y) is the gray level of the pixel (x, y) and T(x, y) is the threshold value at the coordinates (x, y). Coordinate T(x, y) is not dependent and can be considered as a constant for the whole image. Actually it can be used on the basis of the gray level histogram. When there are 2 gray levels in the histogram of an image, it is possible to select a single threshold for the whole image. However, if there is only a gray level, thresholding is impossible and other techniques must be applied. Overall, segmentation can be considered as separation between different regions.

2.1.3 Feature Extraction

Feature extraction are developed to extract certain related information in an image. This technique is used to extract the desired features in the form of data from the image so as to perform the classification in the later part. Features are elements that describes the characteristics of the target in an image. For example, size, shape, colour, composition, line etc. Feature extraction can be considered as the most difficult and the most important part in image processing because it has relation to the efficiency of the recognition system. The selection of the features affect the performance of the recognition system. Therefore, selection of a feature extraction needs to be cautious as the output obtained from it will be going to be applied as the inputs for training process in the classification [13].

One of the example is single value feature extraction. Any real symmetric matrix can be transformed into a diagonal matrix by means of orthogonal transformation, and similarly for any general rectangular matrix $A_{m\times n}$ by means of so-called Singular Value Decomposition as follows.

Let $A_{m\times n}$ be a real rectangular matrix (suppose m > n, without loss of generality) and rank (A) = k. Then there exist two orthonormal matrices $U_{m\times m,}$ $V_{n\times n}$and a diagonal matrix $\Sigma_{m\times n}$ and the following formula holds,

$$A=U\Sigma V^T \tag{14}$$

Where

$$\Sigma = \mathrm{diag}(\lambda_1, \lambda_2, \dots, \lambda_k, 0, \dots, 0) \tag{15}$$

Superscript T denotes transpose; $\lambda_1 > \lambda_2 > \ldots > \lambda_k$. Each λ_i^2, i = 1, 2, …, k, is the eigenvalue of AA^T as well as $A^T A$, λ_i is called singular value of matrix A and

$$U = (u_1, u_2, \dots u_k, u_{k+1}, \dots, u_m), \tag{16}$$

$$V = (v_1, v_2, \dots v_k, v_{k+1}, \dots, v_m) \tag{17}$$

In which, u_i and v_i , $i = 1,2, \dots, k$, are column eigenvectors of AA^T and $A^T A$ corresponding to eigenvalue λ_i^2, respectively; u_i, $i = k + 1, \dots, m$, can be seen as column eigenvectors of AA^T corresponding to eigenvalue $\lambda = 0$, for the sake of convenience of matrix expression. At the same time, v_i, $i = k + 1, \dots, n$ can also be seen as column eigenvectors of $A^T A$ corresponding to eigenvalue $\lambda = 0$. Formula (14) can be written in the form of product sum expression as below:

$$A = \sum_{i=1}^{k} \lambda_i u_i v_i^T \tag{18}$$

With the view of taking an image as a matrix, formula (18) means that the original image A has undergone orthogonal decomposition. We may construct the following column vector consisting of the principal diagonal entries of matrix Σ, singular values λ_i, i = 1, 2, . . . , k, and the $(n-k)$ remainders are zero of principal diagonal.

$$x_{n\times 1} = \Sigma e = \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_k \\ 0 \\ \dots \\ 0 \end{bmatrix} \tag{19}$$

where, column vector e = $(1, 1, \dots, 1)^T_{n\times 1}$. We call $x_{n\times 1}$ Singular Value (SV) feature vector of image A. For any real rectangular matrix A, under the constraint of $\lambda_1 > \lambda_2 > \lambda_k$, diagonal matrix Σ of singular values in formula (14) is unique . Therefore, the original image A corresponds to a unique SV feature vector [26].

There are many types of feature extraction in image processing. Histogram of oriented gradient (HOG) and Local Binary Pattern (LBP) are the most common types.

2.1.3.1 Histogram of Oriented Gradient (HOG)

HOG is a feature descriptor used in computer vision and image processing. The theory behind the HOG is that local object appearance and shape in an image can be described by the distribution of intensity gradients or edge directions. The image is divided into several small cells, and for the pixels within each cell, a histogram of gradient directions is compiled. The descriptor is the concatenation of these histograms. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

The HOG descriptor has a few strengths over other descriptors. Since it operates on local cells, it is invariant to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. To implement HOG, there are normally 5 steps to be done which are gradient computation, orientation binning, descriptor blocks, block normalization and object recognition [40].

The first step of calculation is the computation of the gradient values. The most common method is to apply one dimensional centered, point discrete derivative mask in one or both of the horizontal and vertical directions. The method needs the color filtering or image intensity data with following kernels:

$$[-1, 0, 1] \ and \ [-1, 0, 1]^T$$

Then, the following step is to creating cell histograms. Each pixel in the cell plays an important role for an orientation based histogram channel based on the values found in the gradient computation. The cells can either be radial or rectangular shape and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is “unsigned” or “signed”. Pixel contribution can either be the gradient magnitude or some function of the magnitude.

To account for changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger, spatially connected blocks. The HOG descriptor is then the concatenated vector of the components of the normalized cell histograms from all of the block regions. These blocks typically overlap, meaning that each cell contributes more than once to the final descriptor. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. Circular HOG blocks (C-HOG) can be found in two variants: those with a single, central cell and those with an angularly divided central cell. In addition, these C-HOG blocks can be described with four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial bins. Gaussian weighting provided no advantages when used in conjunction with the C-HOG blocks. C-HOG blocks appear same to shape context descriptors, but is different in that C-HOG blocks contain cells with several orientation channels, while shape contexts only make use of a single edge presence count in their formulation [40].

In block normalization, let v be the non-normalized vector containing all histograms in a given block, $||v||_k$ be its k-norm for $k = 1, 2$ and e be some small constant. Then the normalization factor can be one of the following:

$$L2 - norm: f = \frac{v}{\sqrt{||v||_2^2 + e^2}} \quad (20)$$

L2-hys: L2-norm followed by clipping (maximum value of v is limited to 0.2) and renormalizing, as [41]

$$L1 - norm: f = \frac{v}{\left(\|v\|_1 + e\right)} \tag{21}$$

$$L1 - sqrt: f = \sqrt{\frac{v}{(\|v\|_1 + e)}} \tag{22}$$

In addition, the scheme L2-hys can be computed by taking the L2-norm, clipping the result, and then renormalizing.

In object recognition, HOG may be used for object recognition by providing them as features to a learning algorithm. HOG descriptors are commonly used as features in a support vector machine [40].

2.1.3.2 Local Binary Pattern (LBP)

LBP is a type of visual descriptor used for classification in computer vision. LBP is a useful feature for texture classification. It is often combined with HOG descriptor in order to improve the dataset performance [42]. LBP feature vector is created by first dividing the examined window into cells. For example, 16 x 16 pixels for each cell size. For each pixel within a cell, the pixel to each of its 8 neighbors is compared. Follow the pixels along a circle either in clockwise or counter-clockwise direction where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, "1" is written. This gives an 8 digit binary number. Then, histogram is computed over the cell, of the frequency of each "number" occurring. For example, each combination of which pixels are smaller and which are greater than the center. The histogram can be seen as a 256 dimensional feature vector. The histogram is optionally normalized and the histogram of all cells are concatenated which will gives a feature vector for entire window. After that, the feature vector can be processed using learning machine or learning algorithm to classify images [43].

The following notation is used for the LBP operator: ${LBP_{P,R}}^{u2}$. The subscript represents using the operator in a (P, R) neighbourhood. Superscript u2 stands for using only uniform patterns and labelling all remaining patterns with a single label. After the LBP labelled image $f_l(x, y)$ has been obtained, the LBP histogram can be defined as

$$H_i = \sum_{x,y} I\{f_l\,(x,y) = i\}, i = 0, \dots, n-1 \qquad (23)$$

In which n is the number of different labels produced by the LBP operator and I{A} is 1 if A is true and 0 if A is false. When the image patches whose histograms are to be compared have different sizes, the histograms must be normalized to get a coherent description as

$$N_i = \frac{H_i}{\sum_{j=0}^{n-1} H_j} \qquad (24)$$

2.1.4 Image Classification

Image classification is the labelling of a pixel or a group of pixels based on its grey value [14]. Classification is one of the most often used methods of information extraction. In classification, multiple features are used for a set of pixels. Most of the information extraction techniques rely on analysis of the spectral reflectance properties of each imagery and employ special algorithms designed to perform various types of 'spectral analysis'. The process of multispectral classification can be performed using two methods which are Supervised or Unsupervised [11].

In Supervised classification, the identity and location of some of the land cover types such as urban and wetland are known as priori through a combination of field works and toposheets. The analyst attempts to locate specific sites in the remotely sensed data that represents homogeneous examples of these land cover types. These areas are commonly referred as TRAINING SITES as the spectral characteristics of these known areas are used to 'train' the classification algorithm for eventual land cover mapping of reminder of the image. Multivariate statistical parameters are calculated for each training site. Every pixel both within and outside these training sites is then evaluated and assigned to a class of which it has the highest likelihood of being a member [15].

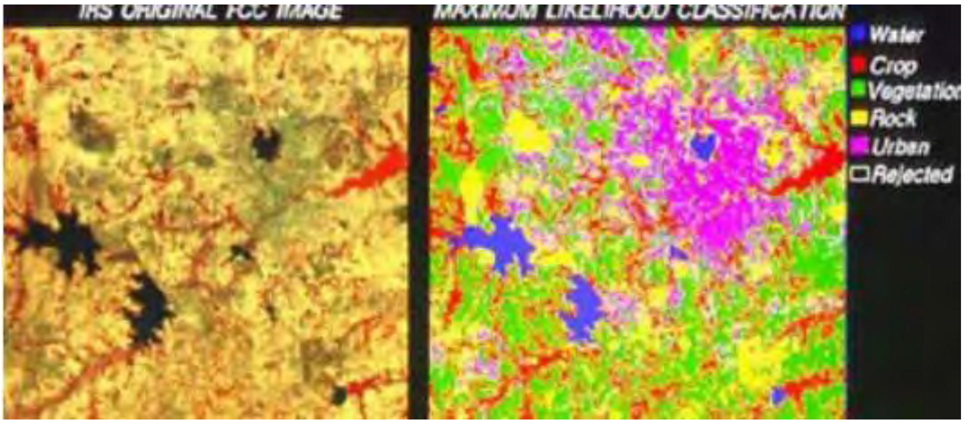

Figure 2.6: Image Classification [11]

On the other hand, the identities of land cover types in an Unsupervised classification has to be specified as classes within a scene are not generally known as priori because ground truth is lacking or surface features within the scene are not well defined. The computer is required to group pixel data into distinct spectral classes according to some statistically determined criteria [11].

2.2 Review of previous related work

Based on Table 2.1, [6] proposed the combination techniques of IPM, adaptive thresholding and feature extraction in the image processing method and using ANN for the classification. IPM was a method used to remove the perspective effect from an image by homogeneously redistributing the information content of the image plane into a new 2D domain. Its application required the knowledge of detailed acquisition conditions such as camera position, orientation, etc. Then, adaptive image thresholding was applied to create binary images for subsequent shape isolation. The threshold was established from the normalised cumulative histogram of the resulting IPM output image. After that, feature extraction method was used to extract the data. In this case, [6] extracted the shape data using own algorithm and classified them using neural network classifier which was a single hidden layer artificial neural network classifier with sigmoid activation based on training with resilient back propagation (RPROP).

[7] also proposed the application of IPM but with other technique such as 8-neighbor Chain Code algorithm and moment feature with rapid computation. In their research, IPM was used to generate binary image. Then, 8-neighbor chain code

algorithm was used to abstract the chain code of a close region. This algorithm was similar like edge detection method which was used to find edges on the images. After that, moment feature and Bhattacharyya were used to select and extract the features. The extracted features were trained using Bayes classifier which was a method from machine learning.

[8] proposed a method that could detect and recognize the road sign. The image processing methods involved including IPM, ROI and feature extraction whereas the recognition method used was multiple candidates feature matching. IPM was first applied to obtain birds-eye view which allowed them to directly compute the gradient on the image to get the Histogram of Oriented Gradients (HOG). ROI was then applied and a set of POI were detected within the ROI image by using FAST corner detection. The HOG would be extracted as input vector for each POI. Then, multiple candidates feature matching was found for each POI and the structural matching algorithm was used to match the POI pairs with the template images.

Meanwhile, [9] proposed a multilevel processing method for image processing and trained binary decision tree for classification. The processing was started by grayscale conversion, then edge detection, followed by multilevel grouping method, closed objects, feature extraction and finally classification. The multilevel grouping was a combination of algorithm working on different levels of object abstraction. It included line extraction, curve detection, parallel detection and closed object finding. It was kind of segmentation method which was used for extracting features more easily. All extracted features were trained using the binary decision tree which was a method from the machine learning.

[10] proposed the combination of Features Vectors and Neural Network technique for real-time daytime road marker recognition. This technique is robust and effective in real time processing as the result data have high accuracy among all the reviews, which about 98%. The research starts with grayscale conversion of the image from the video which changes the RGB image into grayscale image. Then, ROI is used to get the desired information for further processing and remove the unnecessary information in the images. Masking method is used in segmentation of the image to discriminate the lane and the road which lane as 'object' and road as 'background'. In this technique, automatic thresholding is not included. To do so, histogram method is used to decide the threshold value for the discrimination between the lane and the road. After thresholding, erosion algorithm is used to filter the noise of the road surface.

Next, the eroded image is undergoing feature extraction phase. In this case, Hough Transform is used to detect the lines on the image. The detected line will act as input to be used in the neural network classifier. In classification, Multi-Layer Perceptron (MLP) with sigmoid activation based on training with resilient back propagation (RPROP) is applied. After that, simple prediction is applied followed by post-processing. The only limitation of this technique is that it is suitable for daytime application.

Table 2.1: Comparison between previous related research

Title	Type of road marker	Image processing method	Classification method
Automatic real-time road marking recognition using feature driven approach[6]	Arrow and text	-Inverse perspective mapping (IPM) -Adaptive image thresholding -Feature extraction	Artificial neural network with sigmoid activation based on training with resilient back propagation
Road markers recognition based on shape information[7]	Single solid lane, single dash lane and arrow marker	-IPM -8-neighbor chain code -Moment feature and computation	Feature vector, Bayes classifier
A practical system for road marking detection and recognition[8]	Road marker, lane boundary marker and arrow marker	-IPM -ROI and Point of Interest(POI) -Feature extraction	-Multiple candidates feature matching
Applying multi-level processing for robust geometric lane feature extraction[9]	Long wide mark, long thin mark, short wide mark and short thin mark	-Edge detection -Multilevel grouping -Feature extraction	-Trained binary decision tree
Real-Time Daytime Road Marker Recognition Using Features Vectors and Neural Network[10]	Double lane and single dash lane	-Grayscale conversion -ROI -Thresholding -Feature extraction	Artificial Neural Network, multilayer perceptron with backpropagation

2.3 Summary and discussion of the review

Based on the review, the discussion of the technique had been done. In road marker classification, there are 3 main stages which are feature extraction, classification and post-processing. Among all these, feature extraction played the most important role in road marker classification. Many features could be selected to be extracted out as information from the image. For example, features included size, edge, colour and shape.

From the literature review, feature extraction and classification were required in image processing and recognition. Some of the feature extraction method such as Hough Transform was commonly used in recognition. It was useful because most information on the road make solid edges. Based on the literature review, MATLAB software was commonly used as image processing tool. Besides, the training process could also be done in MATLAB using the ANN tool. Besides, there were some additional features that could act as inputs to be trained into the classification process in order to get more accurate result data. For these reason, this research would like to propose a method using feature extraction and neural network.

CHAPTER 3

METHODOLOGY

The proposed technique will be used and explained in detail in this chapter. The technique used is divided into several module in which each module has several tasks need to be achieved in order to get the result data for analysis. The modules are constructed in flowchart shown in Figure 3.1 to show the flow of the proposed technique.

3.1 Research Methodology Flowchart

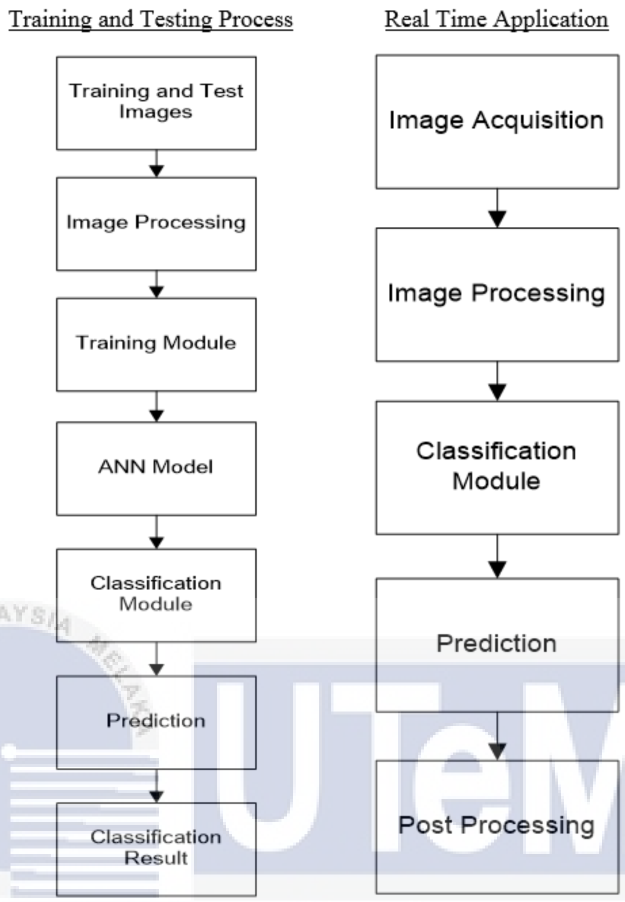

Figure 3.1: Flowchart of research module

Based on the flowchart shown in Figure 3.1, there are a total of 7 modules to be completed in order to achieve the objectives. All the research modules are completed using MATLAB software. It is because MATLAB software consists a lot of toolboxes which are suitable for image processing and pattern recognition. For example, Image Acquisition Toolbox, Image Processing Toolbox, Computer Vision System Toolbox and Neural Network Pattern Recognition Toolbox. In this research, the first module will be the hardware preparation and setup which is important to get the video recording on real time situation. After the video is recorded, image acquisition tool is used to extract frames from the video. The extracted frames will be collected to act as samples for road marker database. The samples from the road marker database will be processed through feature extraction in order to get the feature vector from the sample images. These feature vectors will be acted as input data for the training and classification process. After all the data are completely trained, analysis is done on the accuracy of the classification determined by the features extracted from the image. The highest accuracy of the trained network with the best selected features is saved. Then, real time testing is done by using the trained network to simulate and recognize the test data which is extracted from the test video.

3.2 Discussion on approach

3.2.1 Hardware Preparation and Setup

Initially, camera need to be set up in order to carry out the data collection and real time testing. This can be achieved by setting up all the hardware equipment on a car. The hardware need to be set up included an USB camera and a laptop. An USB camera, Logitech C310, is used to record a video which had three types of road markers and it is stored as WMV file in the laptop. In this case, a laptop with Solid State Drive (SSD) instead of Hard Disk Drive (HDD) is encouraged to be used. This is because when the camera is operating and recording, accidents such as file crash may occur due to some vibration on the road when the car is moving.

Next is the camera positioning. This factor is crucial as incorrect position of camera mounting will result in recording failure. Therefore, focus on the feature that is desired in this research. The feature that this research needs is the road marker. So, the camera will be mounted in front and center of the car which is shown in Figure 3.2. After that, the angle of the mounted camera is also important. If the camera angle is too high, it cannot capture the road frame. On the other hand, if the camera is mounted same angle as the x-axis, there may be some illumination issue on the video frame when strong sunlight is projected into the view of camera. The best angle is to mount the camera slightly lower about 5 to 10 degree with respect to the x-axis. It is optimum because it can capture a good view of road marker and also prevent the flash in recording when there is high intensity of light. After done setting up the camera and laptop, will proceed to image acquisition module.

Figure 3.2: Camera mounted in front of the car

3.2.2 Image Acquisition Module

In image acquisition module, image acquisition tool in MATLAB is used which is shown in Figure 3.3 to get image frames. There are settings and software package need to be done in order to link the Logitech C310 camera to the MATLAB. Before starting the acquisition, there are acquisition parameters that needed to be configured.

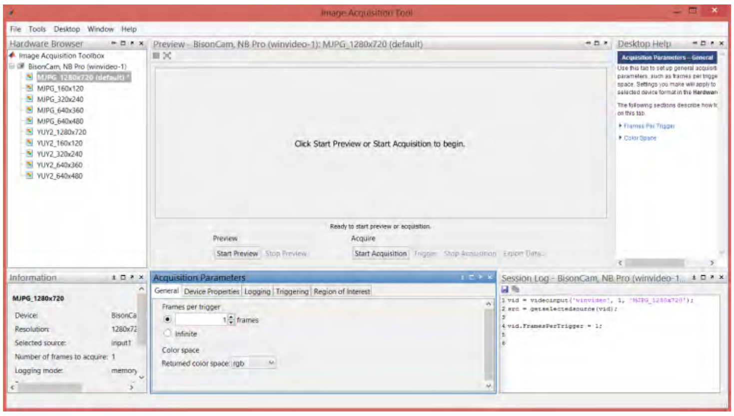

Figure 3.3: Image Acquisition Tool in MATLAB

First, the colour of the image acquisition settings will be in RGB format while the resolution is set to 1024x720. This is to ensure the image captured by the video is colourful and has a clear view due to the high definition resolution. Secondly, the number of frames per trigger determines the number of frames per second to be captured from the video. In this research, the frames per trigger will be set to 30 which meant that the number of frames per second being acquired has 30 samples. Besides, the number of trigger is set to 2 which means the acquisition process run for 2 seconds. The acquisition took place in three different type of road marker situation separately. In total, 60 sample frames for each type of road marker has been acquired and so the total sample frames for three types of road marker is 180. The reason that 60 sample frames for each type of road marker are selected because other sample frames are not suitable to be used due to illumination issue and shadows. Furthermore, these samples will later be used to act as database for training process. Before the training process, feature extraction will be used to extract the feature data out from the sample frames and the feature vector will have many dimensionality. If the sample frames are too many, huge dataset will be used in training process and it will result in error due to many dimensionality on the data. Therefore, 180 sample frames will be collected and stored in database for further preprocessing. The Figure 3.4 shows the frames captured using image acquisition tool.

(a) (b) (c)

Figure 3.4: RGB images acquired from 3 types of road marker (a), (b) and (c)

3.2.3 Preprocessing Module

In this module, there are 5 stages of processing on the sample frames which are grayscale conversion, image sharpening, noise removal, cropping and resizing. This preprocessing module is created in order to make sure the process of feature extraction smooth and easier. The collected frame samples are undergoing the grayscale conversion which are all converted to grayscale image. Next, some of the frame sample may have not so clear. In this case, image sharpening technique is used to make sure the image samples are clear to be seen. Then, "salt and pepper" noise filter is used to remove and filter the noise in the sample frames as the images captured in real time situation have many noise. After that, cropping technique is used to remove other obstacle or unwanted object in the original image samples and crop out the output image samples which has the object that is only desired, road marker. Before going through the training and classification process, all image samples are resized so that they have the same size to make the feature extraction efficient. All the stages of preprocessing are shown in Figure 3.5.

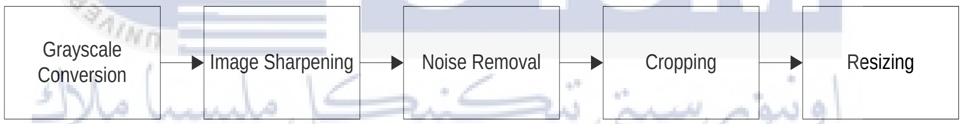

Figure 3.5: 5 Stages of Image Preprocessing

3.2.3.1 Grayscale Conversion

After done storing 180 sample frames in the database, image pre-processing is required. The first step to be done is the grayscale conversion. The reason this technique is used because all the road marker sample frames have the same colour and cannot be differentiated through colour. At the same time, grayscale images are processed faster compared to coloured images. The first step to process the frame is to convert the frame colour from RGB format to grayscale format. The step is known as grayscale conversion and often widely used in image processing. In image processing, colour image consumed more processing time than grayscale image. Most of the colour images have 3 channels which are the red, green and blue whereas the grayscale images have only one channel which is the grayscale level. The RGB images range from [0 0 0] to [255 255 255] whereas the grayscale images range from [0] to [255]. It can be seen clearly that RGB images have total 16, 777, 216 combination colour whereas grayscale images have only total 256 combination colour. This is why most researchers converted the colour images into grayscales images before extracting the features. Similarly, in this experiment, the colour frame is converted into grayscale for efficient processing. This is because this research aims to extract line features not the colour. It will remove all unnecessary information and save a lot of processing time. In order to convert frame into grayscale in the MATLAB coding, rgb2gray() function was used. This command converts the truecolor image RGB to the grayscale intensity image by eliminating the hue and saturation information while retaining the luminance. The Figure 3.6 shows the output image samples after going through grayscale conversion.

(a) (b) (c)

Figure 3.6: Grayscale images acquired from 3 types of road marker (a), (b) and (c)

3.2.3.2 Image Sharpening

Image sharpening is one of the technique in image enhancement. This technique sharpens the images using unsharp masking. Sharpness is actually the contrast between different colours. A quick transition from black to white looks sharp. A gradual transition from black to grey to white looks blurry. Sharpening images increases the contrast along the edges where different colours meet. The unsharp masking technique comes from a publishing industry process in which an image is sharpened by subtracting a blurred (unsharp) version of the image from itself.

In this stage, all the image samples are sharpened and becoming clear compared to before. The principle of it is to extract the high frequency components by subtracting the blurred version from the original as

$$X_{high} = X - h_{\sigma} * X \qquad (25)$$

Where X_{high} is the high frequency components, X is the original image, h_{σ} is the Gaussian kernel with the respective standard deviation σ used for the blurring and operator * is the convolution.

The subtraction is then added to the original:

$$X_{sharp} = X + \alpha X_{high} \qquad (26)$$

Or

$$X_{sharp} = (1 + \alpha)X - \alpha(h_{\sigma} * X) \qquad (27)$$

Where X_{sharp} is the sharpened output image and α is the weighting factor affecting the amount of sharpening. The final result is related to the parameters σ and α. Parameter σ affects the size of neighbourhood in which the sharpening is performed and α is the sharpness of the image [39].

After all the frame samples are being sharpened, the outputs from this stage are processed with the next stage which is noise removal. The output images can be shown in Figure 3.7.

(a) (b) (c)

Figure 3.7: Sharpening of the grayscale images (a), (b) and (c)

3.2.3.3 Noise Removal

For this stage, noise removal technique is used to remove the noise in the image samples from previous stage. This process actually apply some noise to the image and then filter the noise using median filter. In this research, “salt and pepper” noise and median filter are used to remove and filter the sample frames processed from the image sharpening.

The reason that “salt and pepper” noise is used due to its random pixels being set to black or white. For median filter, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values. Median filtering is therefore better able to remove these outliers without reducing the sharpness of the image.

Salt and pepper noise is defined as two-pole impulse noise represents random distributed black points (pepper noise) and white points (salt noise). For an image presented with 8 bit, the probabilistic density function P(z) is given by:

$$P(z) = \begin{cases} a, & z = 0 \\ b, & z = 255 \\ 0, & z = else \end{cases} \tag{28}$$

Where a, b ϵ (0, 1), a + b = r ϵ (0, 1), z is noise grey value, r is noise ratio. When r is low, salt and pepper noise is isolated in some black points and white points, which is distinguished from image details by the gradient easily. When r increases, pepper noise points or salt noise points form mass and are hard to be differentiated with image details. The probability of n pepper noise points adjoined in a line and n salt noise points adjoined in a line is defined as P_n :

$$P_n = \frac{x+y}{M} \tag{29}$$

Where x is the number of n pepper noise points adjoined in a line, y is the number of n salt noise points adjoined in a line and M is the total number of image pixels.

For median filter, it is normally used to reduce noise in an image, almost similar to the mean filter. However, it usually has higher advantage than the mean filter of preserving useful detail in the image. This class of filter belongs to the class of edge preserving smoothing filters which are non-linear filters. This means that for two images A(x) and B(x):

$$median[A(x) + B(x)] \neq median[A(x)] + median[B(x)] \tag{30}$$

These filters smooths the data while keeping the small and sharp details. The median is just the middle value of all the values of the pixels in the neighborhood. Note that this is not the same as the average (or mean); instead, the median has half the values in the neighborhood larger and half smaller. The median is a stronger "central indicator" than the average. In particular, the median is hardly affected by a small number of discrepant values among the pixels in the neighborhood. Consequently, median filtering is very effective at removing various kinds of noise. Figure 3.8 shows an example of median filtering.

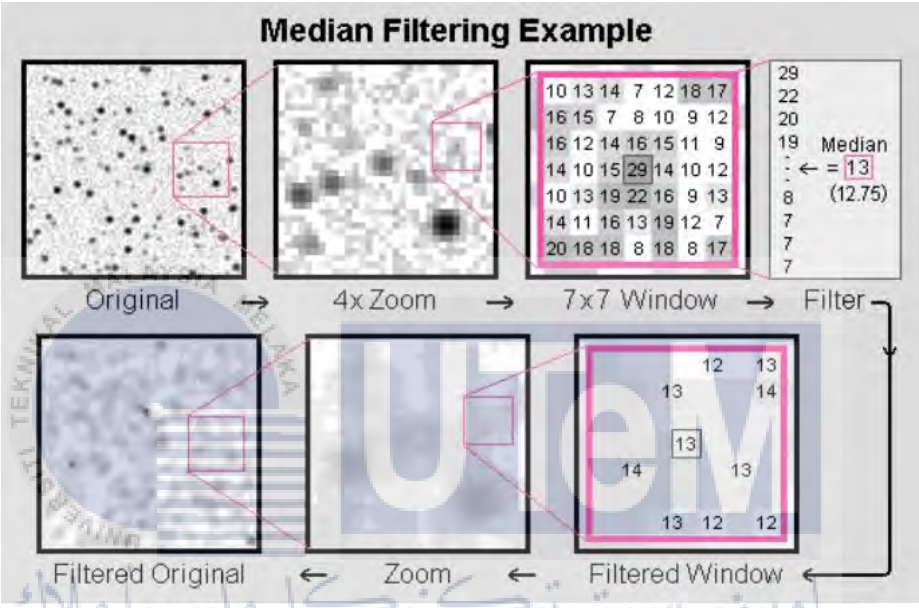

Figure 3.8: Median filtering

Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) Figure 3.9 illustrates an example calculation.

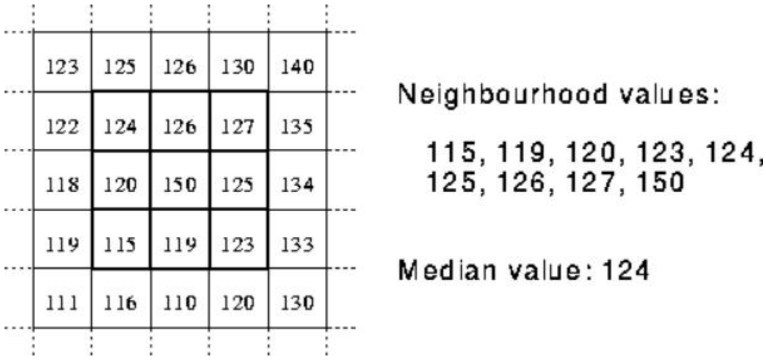

Figure 3.9: Calculation of median value of pixel neighbourhood

As can be seen, the central pixel value of 150 is rather unrepresentative of the surrounding pixels and is replaced with the median value 124. A 3×3 square neighborhood is used here, larger neighborhoods will produce more severe smoothing [38]. The output image samples after removing noise are shown in Figure 3.10.

(a) (b) (c)

Figure 3.10: Noise removal of the sharpened images (a), (b) and (c)

3.2.3.4 Cropping and Resizing

After done sharpening and filtering noise on the image samples, cropping on the image is required. It is because the image samples still contain a lot of things such as cars, trees, buildings and other unwanted curbs or obstacle. The only desired things in the image is the road marker. Thus, the image samples from the previous stage will be cropped manually using imcrop() function in the MATLAB software and the output images are processed with the next stage process, resizing. When all the image samples are cropped manually, the resulted cropped images may have not the same size or dimension. In order to fix the scenario, resizing is necessary to be used on the cropped image samples. Besides, all the image samples must have the similar dimension to avoid high error and low accuracy when going through the feature extraction. The output image samples after cropping and resizing are shown in Figure 3.11.

(a) (b) (c)

Figure 3.11: Cropped and resized image samples (a), (b) and (c)

3.2.4 Feature Extraction

After the sample images going through the preprocessing module, the output image samples are ready for the feature extraction. In this research, two types of feature extraction have been used, Histogram of Oriented Gradient (HOG) and Local Binary Pattern (LBP). This two types of feature extraction are known as common method to extract the feature from the image and the feature data are in the form of feature vector. The only difference between these two methods are the value in the feature vector. For HOG, the value in the feature vector is float number whereas the value in the feature vector for LBP is scalar. Example of the sample frames on HOG visualisation are shown in Figure 3.12, Figure 3.13 and Figure 3.14. The HOG visualisation was done by using the extractHOGFeatures function on the sample images in MATLAB software. Based on the figure below, the HOG visualization shows the total feature vectors corresponding to the certain cell size. The selected cell size 8 by 8 within an images, total number of extracted feature vectors will be 8100. When the selected cell size is 16 by 16 within an images, the total number of extracted feature vectors will be 764. When the selected cell size is 32 by 32 within an image, total number of extracted feature vectors will be 324. It showed that feature length with cell size [8 8] has too much dimensionality while feature length with cell size [32 32] has too less dimensionality. The feature length with cell size [16 16] shows an optimum dimensionality enough to be used in training process and act as input data.

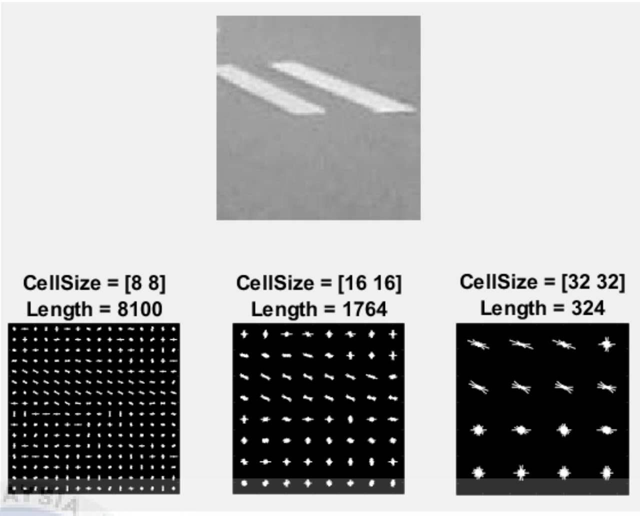

Figure 3.12: HOG visualisation (Double dash)

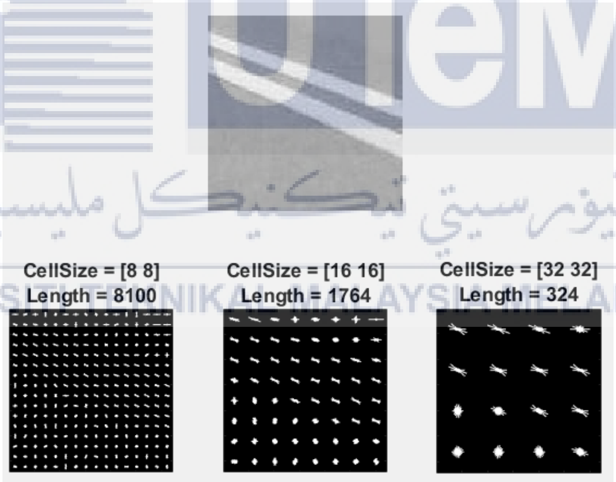

Figure 3.13: HOG visualisation (Double solid)

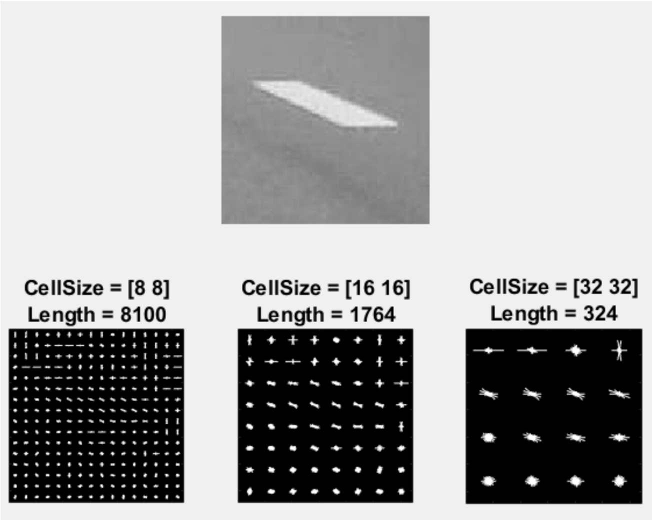

Figure 3.14: HOG visualisation (Single dash)

3.2.5 Training and Classification Module

Once features had been extracted, they might be used to build models for accurate classification. Artificial neural-network (ANN) method was used to classify the type of the road lane markers. ANN was the term used to describe a computer model assumption of the biological brain. It usually consisted of a set of interconnected simple processing units (neurons or nodes) which combined to give an output signal to solve a certain problem based on the input signals it received. There were 2 types of ANN which were supervised learning and unsupervised learning. Supervised learning meant learning with teacher signals or targets whereas unsupervised learning meant learning without the use of teacher signals. In this case, supervised learning was used in this research as it was easier and there were feature data gathered for training purpose. In the MATLAB software, there was a neural network pattern recognition tool which could be used to train and classify the data. The tool could be opened by typing nprtool in the MATLAB command. A graphic user interface (GUI) would show up and data could be inserted to be trained.

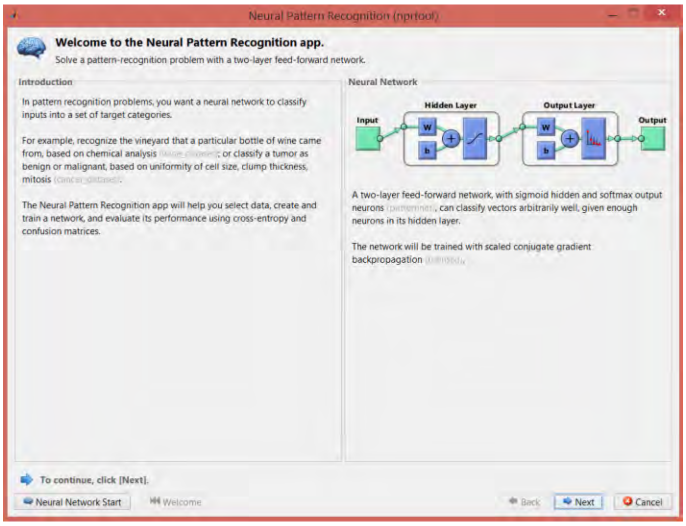

Figure 3.15: Neural Network Regconition Tool

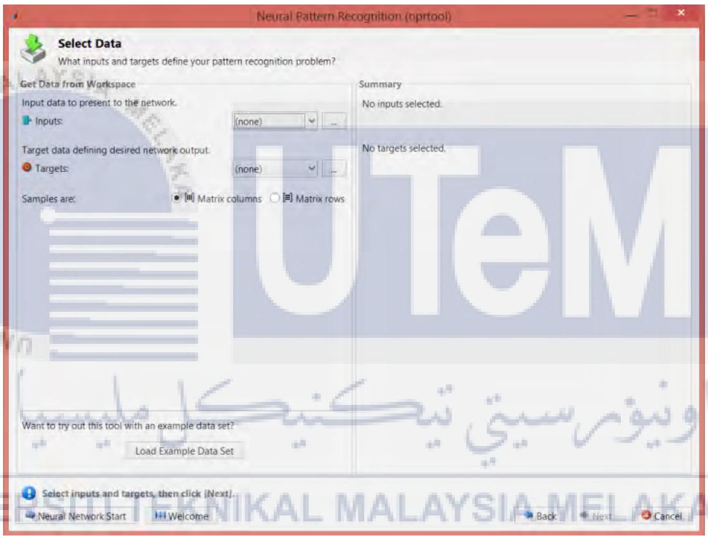

Figure 3.16: Data Selection Interface

Before start training, the data set that gathered from the feature extraction must arrange in a table form which the type of feature as row and the number of sample as column. Besides, the training process also required target data to be trained with the input data. This research aimed to classify 3 types of road marker which meant there were 3 types of target data. The target data could be any number and random but in this research, 1, 2 and 3 were used as the target value to differentiate each group. After completing the data set preparation, the training process could be started and the result would be shown in Chapter 4.

CHAPTER 4

RESULT AND DISCUSSION

All the experimental results are collected and tabulated. An Intel Core-i7 2.60 GHz CPU computer with 12 GB of RAM is used to run this program to recognize road markers. The prototype is developed within the MATLAB environment. The image acquisition toolbox, image processing toolbox, computer vision toolbox and neural network toolbox are used to implement this system.

An USB camera is mounted in front of a moving vehicle to capture video from a real-time environment. The video is segmented frame by frame with one second intervals, and it goes through grayscale conversion, image sharpening, noise removal, cropping and resizing. The extracted images are converted into a grayscale image and normalized to 128-by-128 pixels. The database images pass through the HOG and LBP feature extraction process to extract 1764 and 256 feature vectors respectively. These feature vectors are used to train the artificial neural network for recognition of the road marker.

4.1 Training and Testing Set Images

In this research, several frames are collected. Out of these all frames, a total number of 180 image frames are selected. 60 image frames for each type of road markers are selected. The reason that 180 image frames are selected is due to other image samples are not clear. The rest of the image samples have shadows and the road marker is blurred which cannot be used in training process as mentioned in the scope. Besides, more image frames to be used in training process also required more processing time which will delay the post processing stages. These 180 selected image frames are stored in database in order to be used in training and classification process. Example of 8 different image samples for each type of road markers are shown in Figure 4.1, Figure 4.2 and Figure 4.3.

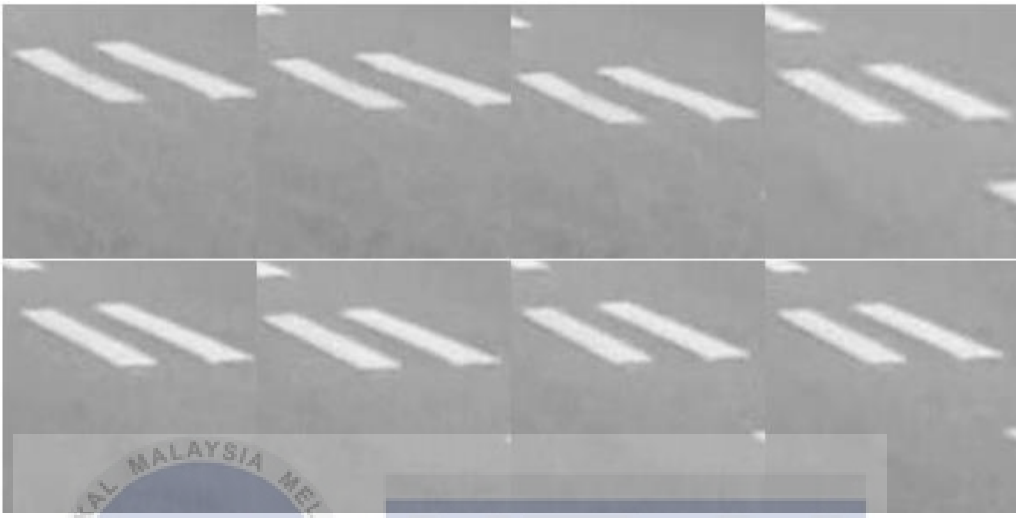

Figure 4.1: Sample of double dash marker in database

Figure 4.2: Sample of double solid marker in database

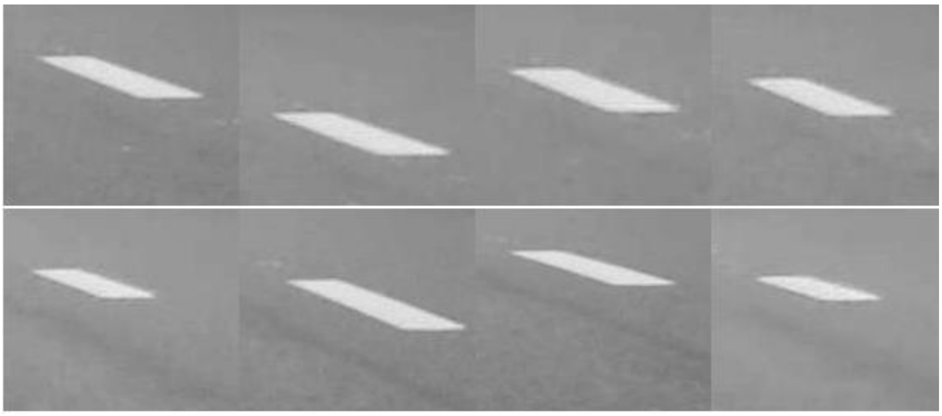

Figure 4.3: Sample of single dash marker in database

4.2 Training, Testing and Validation Data (HOG)

For using HOG as feature, a set of 60 samples are used for each type of road marker and 180 samples for 3 classes of road marker. The extracted 180-by-1764 features vector is used as an ANN input data set for training, testing, and validating the network.

Figure 4.4 shows the HOG as feature for neural network's performance, training state, error histogram, and overall confusion matrix. From the performance plot, the cross-entropy error is maximum at the beginning of training. For this proposed system, the best validation performance is at epoch 18, and at this point the cross-entropy error is very close to zero. On the training state plot, the maximum validation check 6 at epoch 24 and at this point, the neural network halts the training process to give best performance. The error histogram plot represents that the error of this system is very close to zero. An overall confusion matrix is three sets of combined confusion matrices, which are the training confusion matrix, validation confusion matrix, and testing confusion matrix. This overall confusion matrix plot shows 98.9% correct classification for this system.

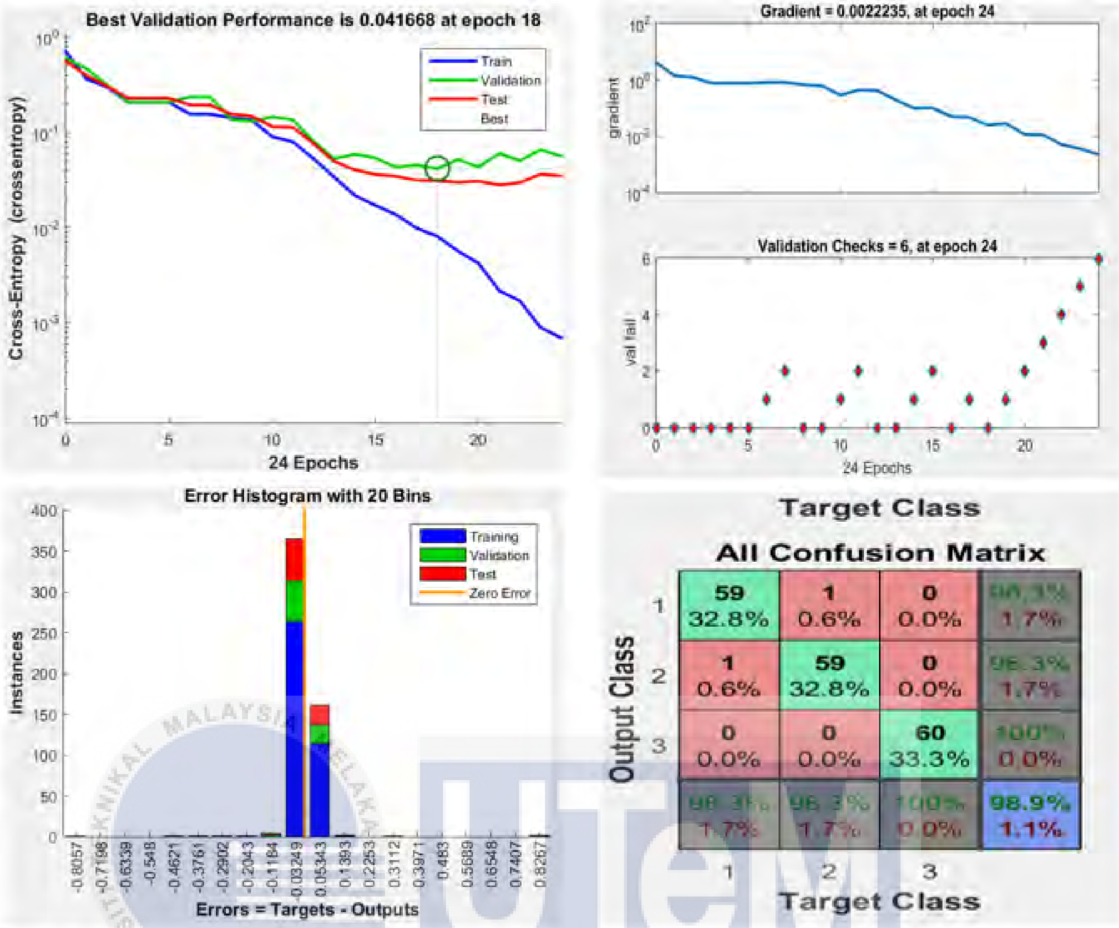

Figure 4.4: Neural Network (HOG): Performance, Training State, Error Histogram, and Overall Confusion Matrix

Receiver Operating Characteristic (ROC) curve of the network which illustrates true positive rate verses false positive rate at various threshold settings of the network, is shown in Figure 4.5. Area under the curve (AUC) shows a slightly result for this proposed system. At the neural network train, test and validation conclusion, this network performs 98.9% correct classification of 3 classes of road marker.

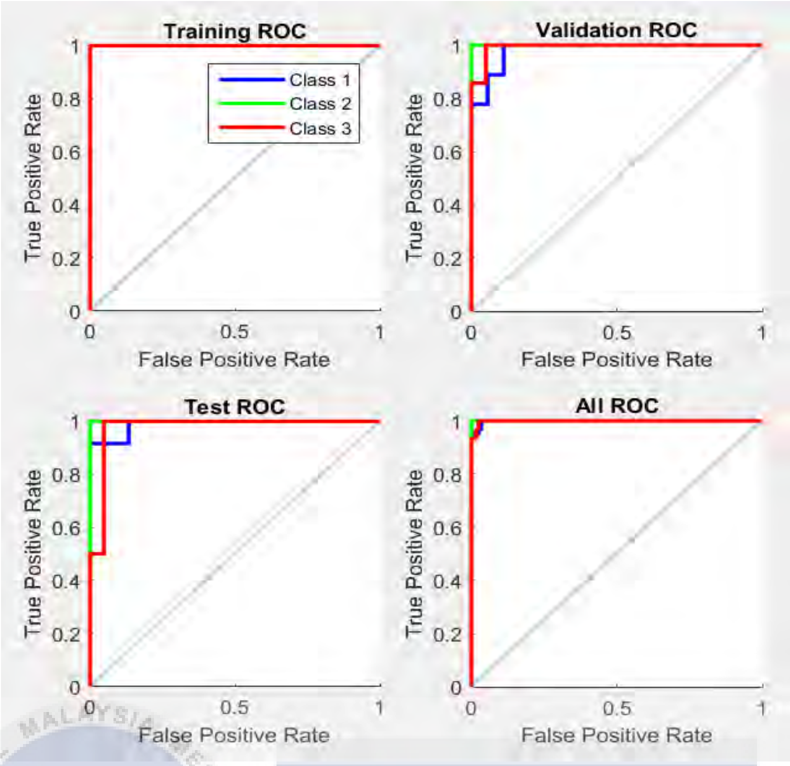

Figure 4.5: HOG Network Receiver Operating Characteristic (ROC)

4.3 Training, Testing, and Validation Data (LBP)

For using LBP as feature, the extracted 180-by-256 features vectors are used as an ANN input data set for training, testing, and validating the network.

Figure 4.6 shows the LBP as feature for neural network's performance, training state, error histogram, and overall confusion matrix. From the performance plot, the cross-entropy error is higher at the beginning of training compared to HOG. For this system, the best validation performance is at epoch 31, and at this point the cross-entropy error is closer and more stable to zero compared to HOG. On the training state plot, the maximum validation check 6 at epoch 37 and at this point, the neural network halts the training process to give best performance. The error histogram plot represents that the error of this system is very close to zero. The overall confusion matrix plot shows 98.3% correct classification for this system.

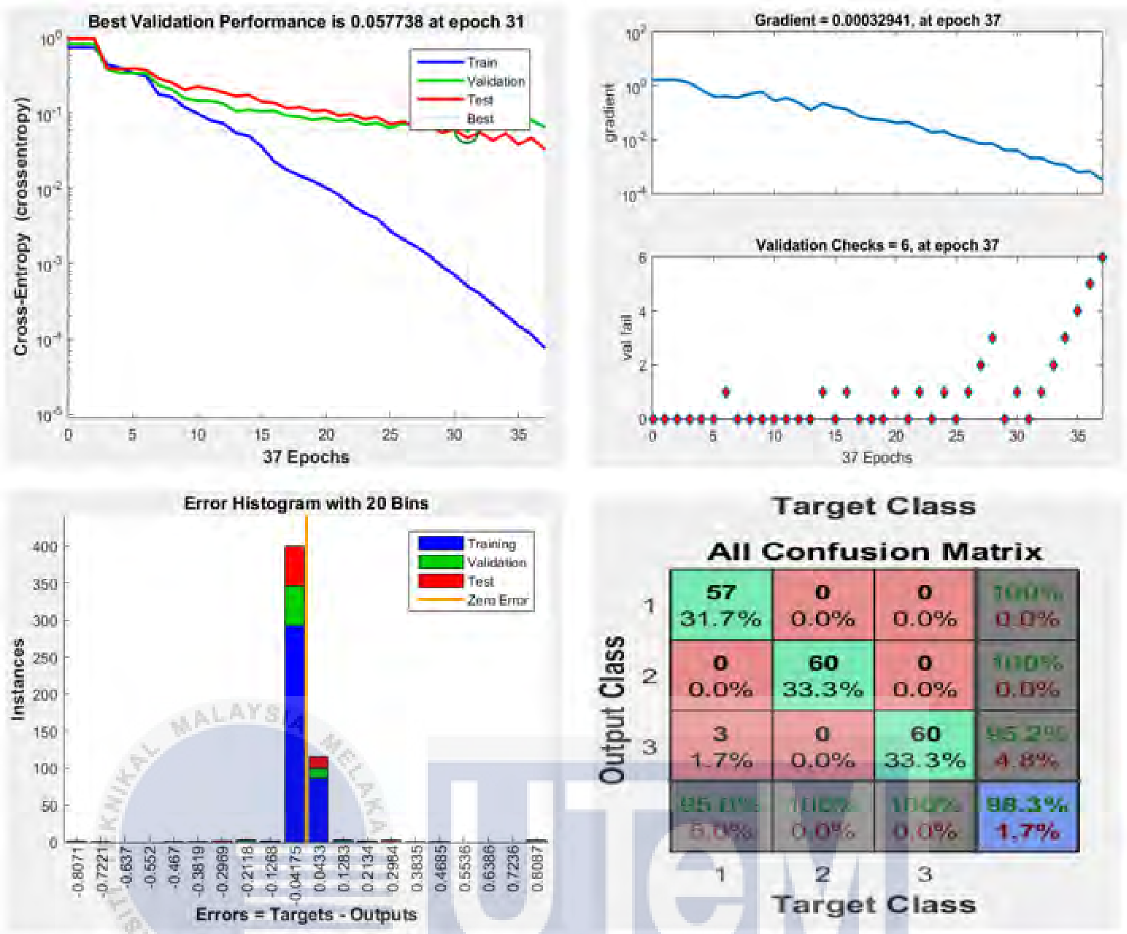

Figure 4.6: Neural Network (LBP): Performance, Training State, Error Histogram, and Overall Confusion Matrix

Receiver Operating Characteristic (ROC) curve of the network which illustrates true positive rate verses false positive rate at various threshold settings of the network, is shown in Figure 4.7. Area under the curve (AUC) shows a slightly perfect result for this proposed system. At the neural network train, test and validation conclusion, this network performs 98.3% correct classification of 3 classes of road marker.

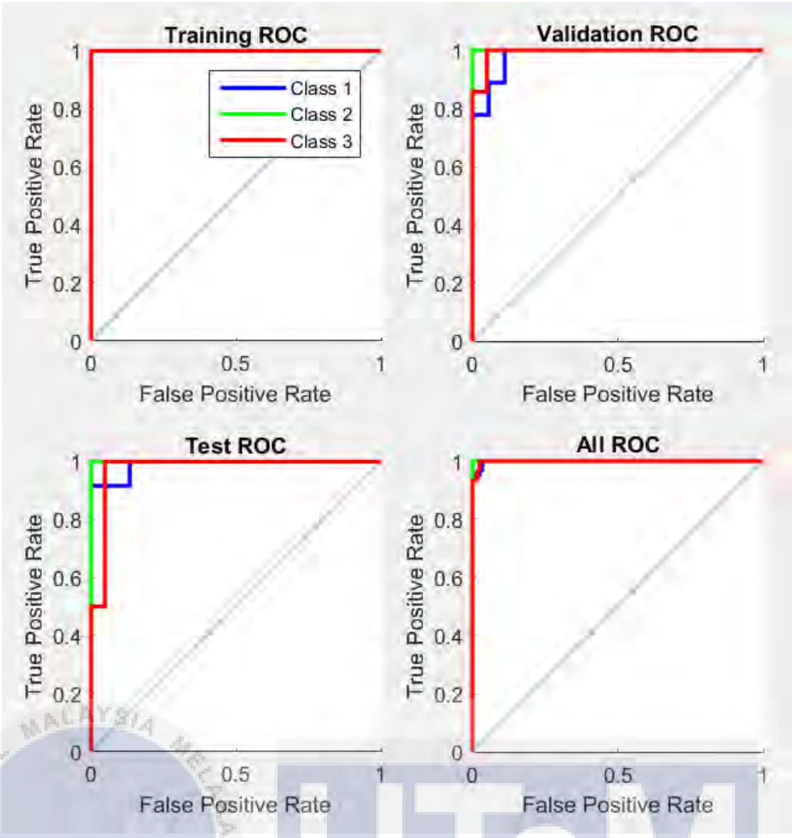

Figure 4.7: Network Receiver Operating Characteristic (ROC)

4.4 Training, Testing, and Validation Data (HOG&LBP)

For using HOG and LBP as feature, the extracted 180-by-2020 features vectors are used as an ANN input data set for training, testing, and validating the network.

Figure 4.8 shows the HOG and LBP as feature for neural network's performance, training state, error histogram, and overall confusion matrix. From the performance plot, the cross-entropy error is higher at the beginning of training compared to HOG. For this system, the best validation performance is at epoch 23, and at this point the cross-entropy error is closer and more stable to zero compared to HOG. On the training state plot, the maximum validation check 6 at epoch 29 and at this point, the neural network halts the training process to give best performance. The error histogram plot represents that the error of this system is very close to zero. The overall confusion matrix plot shows 99.4% correct classification for this system.

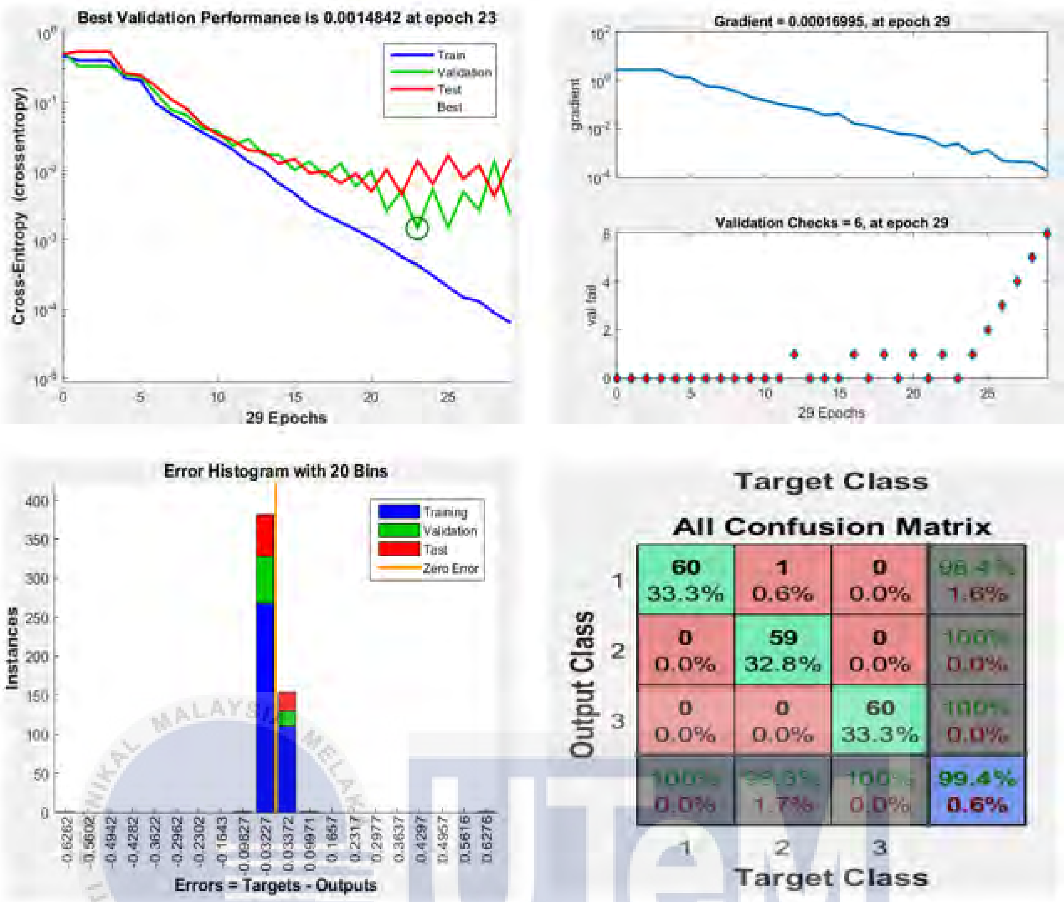

Figure 4.8: Neural Network (HOG and LBP): Performance, Training State, Error Histogram, and Overall Confusion Matrix

Receiver Operating Characteristic (ROC) curve of the network which illustrates true positive rate verses false positive rate at various threshold settings of the network, is shown in Figure 4.9. Area under the curve (AUC) shows a slightly perfect result for this proposed system. At the neural network train, test and validation conclusion, this network performs 99.4% correct classification of 3 classes of road marker.

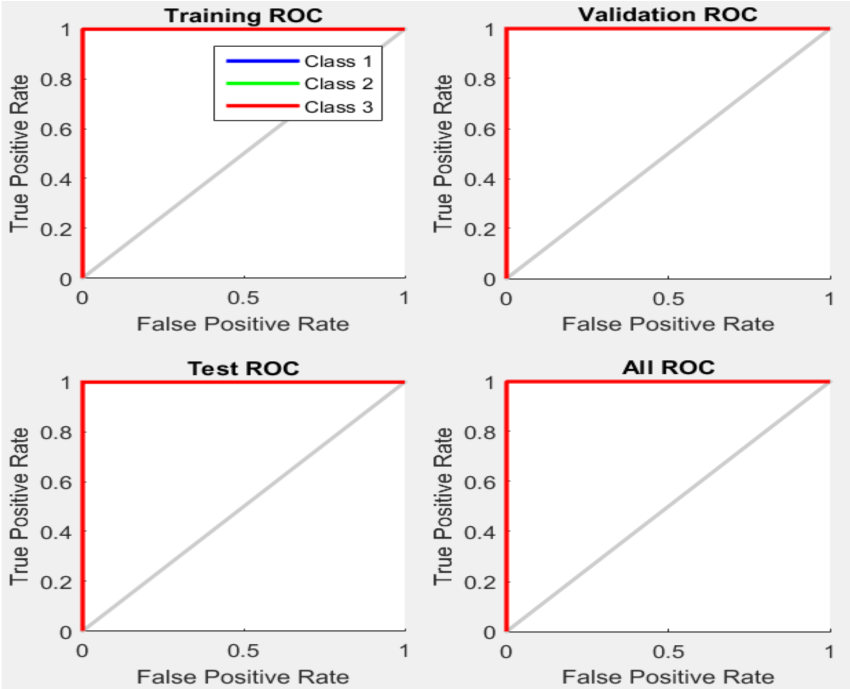

Figure 4.9: HOG and LBP Network Receiver Operating Characteristic (ROC)

Table 4.1: Accuracy of the classification determined by feature extraction method

Feature extraction method	Accuracy (%)
HOG	98.9
LBP	98.3
HOG and LBP	99.4

Based on the Table 4.1, it is clearly that the combination of HOG and LBP feature extraction method will result in higher accuracy in the classification process.

4.5 Experiment with Real Time

After getting the trained network, the system was tested with new video input on the real time situation. Different types of road markers situation were tested with the trained network. The accuracy and result were shown in Table 4.2 and Table 4.3.

Table 4.2: Accuracy of the classification on real time video

Experiment	Road Marker	Number of Frame	Classification Accuracy
1	Double Dash	91	41.76%
2	Double Solid	89	100.00%
3	Single Dash	180	68.33%

Table 4.3: Confusion matrix of the overall accuracy on real time video

	1	**2**	**3**	
1	38 (10.56%)	35 (9.72%)	18 (5%)	41.76%
2	0	89 (24.72%)	0	100.00%
3	48 (13.33%)	9 (2.5%)	123 (34.17%)	68.33%
	44.19%	66.92%	87.23%	69.44%

Based on Table 4.2, three experiments were carried out on real time situation. Each experiment was tested with the trained network on each type of the road marker. In experiment 1, the type of the road marker was double dash and 91 samples of frames were tested and classified. The accuracy of the classification on real time for experiment 1 was 41.76%. In experiment 2, the type of the road marker was double solid and 89 samples of frames were tested and classified. The accuracy of the classification on real time for experiment 2 was 100.00%. In experiment 3, the type of the road marker was single dash and 180 samples of frames were tested and classified. The accuracy of the classification on real time for experiment 3 was 68.33%.

In Table 4.3, the first three diagonal cells showed the number and percentage of correct classifications by the trained network. For example, 38 samples were correctly classified as Class 1 which was double dash road marker. This corresponded to 10.56% of all 360 samples. Secondly, 89 samples were correctly classified as double solid road marker. This leaded to 24.72% of all 360 samples. Similarly, 123 samples were correctly classified as single dash road marker. This corresponded to 34.17% of all 360 samples.

35 samples of the double dash markers were incorrectly classified as double solid marker. This corresponded to 9.72% of all 360 samples in the data. 18 samples of the double dash markers were wrongly classified as single dash marker. This corresponded to 5% of all data. At the same time, 48 samples of single dash markers were wrongly classified as double dash marker and this corresponded to 13.33% of all data. Other 9 samples of single dash marker were incorrectly classified as double solid marker which corresponded to 2.5% of all the samples.

Out of 91 double dash marker predictions, 41.76% were correct and 58.24% were wrong. Out of 89 double solid marker predictions, all 89 samples were predicted correctly which indicated 100% correct. Out of 180 single dash marker predictions, 68.33% were correct and 31.67% were wrong.

On the other hand, out of 86 double dash marker cases, 44.19% were correctly predicted as double dash marker and 55.84% were wrongly predicted. Out of 133 double solid marker cases, 66.92% were correctly classified as double solid marker and 33.08% were classified as other type of road marker. Out of 141 single dash marker cases, 87.23% were correctly classified as single dash marker and 12.77% were classified as other type of road marker. Overall, 69.44% of the predictions were correct and 30.56% were wrong.

In addition, there are also some errors. Example of these errors are shown in Figure 4.13 and Figure 4.14. Figure 4.13 showed that the classification on the road without the marker is classified as double solid marker. This is due to the trained network had only 3 classes of target markers which are double dash marker, double solid marker and single dash marker. The road without marker is not included in the classification process. Due to this reason, the feature extracted from the image in Figure 4.13 may have almost the same value as in feature vector in double solid class and so it is classified in the double solid class even though it is not. For Figure 4.14, the recognition is not accurate where it showed double solid marker but the recognized image is not. This is due to this research is focused on the accuracy of the recognition of the road marker. When the recognized region is placed, it is fixed. When there is a curve on the road in the video, the road marker may not be included in the recognized region. Thus, unable to classify the road marker accurately.

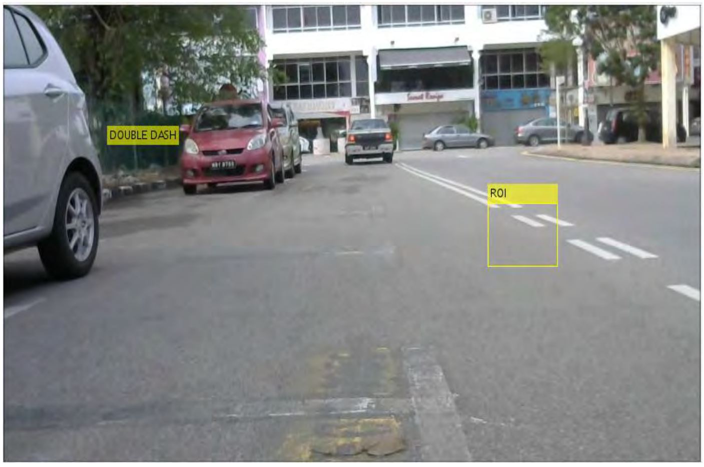

Figure 4.10: Real time test on double dash road marker

Figure 4.11: Real time test on double solid road marker

Figure 4.12: Real time test on single dash road marker

Figure 4.13: Error on real time test

Figure 4.14: Error on real time test

CHAPTER 5

CONCLUSION

In the conclusion, the classification result of the road marker in this FYP 2 were able to be obtained. Among all the feature extraction methods, the result of the combination of HOG and LBP as the feature extraction method showed the highest accuracy which was 99.4%. The testing was also carried out using a trained network against the real time situation. The result of classification accuracy on the real time road marker of double dash, double solid and single dash markers are 41.76%, 100% and 68.33% respectively. The overall accuracy of the classification on real time road marker is 69.44%. However, there are some errors in the real time situation which is due to the region selection. The errors can be solved by implementing detection system on the road marker.

In overall, all the objectives of the project had been achieved which was able to classify three different types of urban road lane marker using neural network method, developing a real time recognition using image processing method and improve the image data accuracy getting from the video using MATLAB software.

In the future work, this system can be improved more accurately by adding the detection system in order to detect the marker automatically before classified.

REFERENCES

1. ASEAN Road Safety Centre, "General Road Accident Data in Malaysia (1997 – 2016)", Malaysian Institute of Road Safety Research (MIROS) official website, 2017.
2. Veena Babulal, "Number of fatal road accidents up in 2016, more than 7000 lives lost", New Straits Times, January 2017.
3. Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision", McGraw-Hill, Inc., ISBN 0-07-032018-7, 1995.
4. Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Prentice Hall, Inc., ISBN 0-13-094650-8, 2002.
5. B. Chitradevi, P. Srimathi, "An Overview on Image Processing Techniques", International Journal of Innovative Research in Computer and Communication Engineering, vol. 2, Issue 11, November 2014.
6. Kheyrollahi, A., & Breckon, T. P. (2012). Automatic real-time road marking recognition using a feature driven approach. *Machine Vision and Applications*, *23*(1), 123-133.
7. Li, Y., He, K., & Jia, P. (2007, June). Road markers recognition based on shape information. In *Intelligent Vehicles Symposium, 2007 IEEE* (pp. 117-122). IEEE.
8. Wu, T., & Ranganathan, A. (2012, June). A practical system for road marking detection and recognition. In *Intelligent Vehicles Symposium (IV), 2012 IEEE* (pp. 25-30). IEEE.
9. Lindner, P., Blokzyl, S., Wanielik, G., & Scheunert, U. (2010, September). Applying multi level processing for robust geometric lane feature extraction. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2010 IEEE Conference on* (pp. 248-254). IEEE.
10. Z. Md Sani, L. W. Sen, H. Abd. Ghani, R. Besar, "Real-Time Daytime Road Marker Recognition Using Features Vectors and Neural Network, 2015 IEEE Conference, 2015, pp. 38-43.

11. K.M.M. Rao, Anil K. Jain, "Overview of Image Processing", Readings in Image Processing Fundamentals Of Digital Image Processing, Prentice-Hall, 1989.
12. Gaurav Kumar, Pradeep Kumar Bhatia, "A Detailed Review Of Feature Extraction In Image Processing System", 2014 Fourth International Conference on Advanced Computing & Communication Technologies.
13. Ernest L. Hal, "Computer Image Processing And Recognition", Academic Press, 1979.
14. Chellappa, "Digital Image Processing", 2nd Edition, IEEE Computer Society Press, 1992.
15. M. Mansourpour, M.A. Rajabi, J.A.R. Blais, "Effects And Performance Of Speckle Noise Reduction Filters On Active Radar And SAR Images", ISPRS, XXXVI/1-W41.
16. P.A. Devijver, J. Kittler, "Pattern Recognition: A Statistical Approach", London: Prentice Hall, 1982.
17. Sivaraman, S., & Trivedi, M. M. (2010). A general active-learning framework for on-road vehicle recognition and tracking. *IEEE Transactions on Intelligent Transportation Systems*, *11*(2), 267-276.
18. Aufrere, R., Chapuis, R., & Chausse, F. (2001). A model-driven approach for real-time road recognition. *Machine Vision and Applications*, *13*(2), 95-107.
19. Estevez, L., & Kehtarnavaz, N. (1996, April). A real-time histographic approach to road sign recognition. In *Image Analysis and Interpretation, 1996., Proceedings of the IEEE Southwest Symposium on* (pp. 95-100). IEEE.
20. Cheng, H., Zheng, N., & Liu, T. (2002, August). An approach of road recognition based mean shift and feature clustering. In *Signal Processing, 2002 6th International Conference on* (Vol. 2, pp. 1059-1062). IEEE.
21. Li, W., Jiang, X., & Wang, Y. (1998). Road recognition for vision navigation of an autonomous vehicle by fuzzy reasoning. *Fuzzy Sets and Systems*, *93*(3), 275-280.
22. Paetzold, F., & Franke, U. (2000). Road recognition in urban environment. *Image and Vision Computing*, *18*(5), 377-387.
23. Maldonado-Bascón, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gómez-Moreno, H., & López-Ferreras, F. (2007). Road-sign detection and recognition based on

support vector machines. *IEEE transactions on intelligent transportation systems*, *8*(2), 264-278.

24. Fu, Y., Yan, T., Zhang, D. W., & Meng, M. Q. H. (2012, June). A partial differential equation algorithm for image enhancement. In *Information and Automation (ICIA), 2012 International Conference on* (pp. 606-610). IEEE.
25. Besag, Julian (1986). "On the Statistical Analysis of Dirty Pictures". *Journal of the Royal Statistical Society. Series B (Methodological)*. **48** (3): 259–302. JSTOR 2345426
26. Hong, Z. Q. (1991). Algebraic feature extraction of image for recognition. *Pattern recognition*, *24*(3), 211-219.
27. Subramaniam, S. K., Ganapathy, V. R., Subramonian, S., & Hamidon, A. H. (2010). Automated traffic light system for road user's safety in two lane road construction sites. *WSEAS Transactions on Circuits and Systems*, *2*(9), 71-80.
28. Gannapathy, V. R., Subramaniam, S. K., Diah, A. M., Suaidi, M. K., & Hamidon, A. H. (2008). Risk factors in a road construction site. *Proceedings of the World Academy of Science, Engineering and Technology*, *46*, 640-643.
29. Jo, K., Kim, J., Kim, D., Jang, C., & Sunwoo, M. (2014). Development of autonomous car—Part I: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, *61*(12), 7131-7140.
30. Milanés, V., Llorca, D. F., Vinagre, B. M., González, C., & Sotelo, M. A. (2010, September). Clavileño: Evolution of an autonomous car. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on* (pp. 1129-1134). IEEE.
31. Report from Community database on accidents on the roads in Europe (CARE). www.ec.europa.eu/transport.
32. ECT, "White paper. european transport policy for 2010. time to decide," European Commission Transport, Tech. Rep., 2001.
33. S. Thrun et al., "Stanley: The robot that won the DARPA Grand Challenge," J. Field Robot., vol. 23, no. 9, pp. 661–692, Sep. 2006.
34. C. Urmson et al., "Autonomous driving in urban environments: Boss and the Urban Challenge," J. Field Robot., vol. 25, no. 8, pp. 425–466, Aug. 2008.
35. Acharya and Ray, *Image Processing: Principles and Applications*, Wiley-Interscience 2005 ISBN 0-471-71998-6.

36. Russ, *The Image Processing Handbook: Fourth Edition*, CRC 2002 ISBN 0-8493-2532-3.
37. Yu, J., Guo, H., & Li, C. (2010, September). A filtering algorithm for removing salt and pepper noise and preserving details of images. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on* (pp. 1-4). IEEE.
38. Lei, P. (2004). Adaptive median filtering. In *Seminar Report, Machine Vision* (Vol. 140).
39. Krasula, L. (2013). Quality assessment of sharpened images. *Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Radio Engineering*.
40. Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.
41. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, *60*(2), 91-110.
42. Wang, X., Han, T. X., & Yan, S. (2009, September). An HOG-LBP human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on*(pp. 32-39). IEEE.
43. Barkan, O., Weill, J., Wolf, L., & Aronowitz, H. (2013). Fast high dimensional vector multiplication face recognition. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1960-1967).

APPENDICES A

Image Acquisition Coding

```
vid = videoinput('winvideo', 1, 'RGB24_640x480');
src = getselectedsource(vid);

vid.FramesPerTrigger = 1;

% TriggerRepeat is zero based and is always one
% less than the number of triggers.
vid.TriggerRepeat = 1;

vid.FramesPerTrigger = 30;

preview(vid);

start(vid);

stoppreview(vid);
diskLogger = VideoWriter('C:\Users\Admin\Desktop\Sem 8\FYP
Matlab\videos.mp4', 'MPEG-4');
open(diskLogger);
data = getdata(vid, vid.FramesAvailable);
numFrames = size(data, 4);
for ii = 1:numFrames
    writeVideo(diskLogger, data(:,:,:,ii));
end
close(diskLogger);
```


Grayscale Conversion

```
for i=1:60
    a=imread([num2str(i) '.jpg']);
    filename=strcat('GS',num2str(i),'.jpg');
    b=rgb2gray(a);
    imwrite(b,filename);
end
```

Image Sharpening

```
clc;
clear;
for i=1:60
    a=imread([num2str(i) '.jpg']);
    filename=strcat('SP',num2str(i),'.jpg');
    b=rgb2gray(a);
    c=imsharpen(b);
    imwrite(c,filename);
end
```

Noise Removal

```
clc;
clear;
for i=1:60
    a=imread([num2str(i) '.jpg']);
    filename=strcat('NR',num2str(i),'.jpg');
    b=rgb2gray(a);
    c=imsharpen(b);
    d = imnoise(c,'salt & pepper',0.02);
    Kmedian = medfilt2(d);
    imwrite(Kmedian,filename);
end
```

Cropped and Resize

```
Count=0;
count=count+1;
  if count== 1
     [j, rect]=imcrop(Kmedian);
     f=imcrop(Kmedian,rect);
  else
      f=imcrop(Kmedian,rect);
  end
  u = imresize(f,[128 128]);
```

LBP Extraction

```
for i=1:60

    im=imread(['ROI' num2str(i) '.jpg']);
   % Extract LBP features
    a(i,:)= lbp(im);
    lbpe=a';

end
```


HOG Extraction

```
for i=1:60

    im=imread(['ROI' num2str(i) '.jpg']);
   % Extract HOG features
    hog = extractHOGFeatures(im,'CellSize',[16 16]);
    a(i,:)=hog;
    hoge=a';
end
```

APPENDICES B

Table 5.1: Final Year Project 1 Gantt Chart

Plan	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Title Selection and Confirmation														
Problem Statement and Objective														
Research on road lane markers in Malaysia														
Review on previously related work														
Proposed suitable method and technique														

Hardware and video preparation														
Image Processing Algorithm using MATLAB														
Presentation														
Finalized Report														

Table 5.2: Final Year Project 2 Gantt Chart

Plan	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
ANN Training using MATLAB														
Troubleshooting error														
Improving accuracy of data														
Real-Time Video Testing														
Presentation														
Finalized Report														