AUTOMATIC CAR LICENSE PLATE RECOGNITION SYSTEM (CLPR)

RABI'ATUL ADAWIYAH BINTI MUSTAFA

This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Hons.) (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

OCTOBER,2008

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : ______________________________

Author : RABI’ATUL ADAWIYAH BINTI MUSTAFA

Date : 28 OCTOBER 2008

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Electronics)"

Signature : ______________________________

Name : REZA EZUAN BIN SAMIN

Date : 20 APRIL 2008

To my belove mother and father,

Mr.Mustafa Bin Sulaiman
Mrs. Selamah Binti Yaakob

Thank you for supporting me all the time

AKNOWLEDGEMENT

First of all I am grateful to ALLAH S.W.T for blessing and giving me enough courage in completing my final year project (PSM).

Secondly I would like to thank my family for giving morale support and encouragement in completing my project and also throughout my study in UMP as they are my inspiration to success. Unforgettable, I want to give a special gratitude to my supervisor En. Reza Ezuan Bin Samin for guiding and supervising my final year project throughout these two semesters. He has been very helpful to me in finishing my project and I appreciate every advice that he gave me in correcting my mistakes. I apologize to my supervisor for any mistakes and things that I done wrong while finishing my project. The credit also goes to all lecturers and everyone who willing to share their knowledge, cooperation and guide that is related to my project.

Last but not lest I want to thank all my friends that have gave me advice and encouragement in completing my project. Thank you very much to all and May ALLAH bless you.

ABSTRACT

The growth of technologies requested higher performance tools in order to fulfill human needs and market. This system is implemented to make human work easier besides can reduce the uses of human power and because of its potential application. The development of automatic car license plate recognition system will resulted greater efficiency for vehicle monitoring system. Car plate recognition systems are used commercially, both in overseas and locally. In Malaysia, however the usage of car plate recognition system is restricted to the ordinary car plates. This means that the system is unable to detect special types of car plates. Therefore, this system is aimed for implementation of a recognition system for special Malaysian car plates. This system is implementing by using MATLAB7.1 Image Processing Toolbox, which uses optical character recognition on images to read the license plates on vehicles. The system is an online system where the image will automatically extracted once after the image is captured by webcam using image processing technique. First, the image is converted into a binary image and then the chosen area will be cropped so that only the plate number is left .Next, the image is compliment so that the black plate background becomes white while the white plate number becomes black because the system can only detect binary image where the background should be white while the plate number should be black. One of the important step is the integration between image processing and Graphical User Interface (GUI) where, the output of this project will displayed using GUI.

ABSTRAK

Perkembangan teknologi yang pesat mendorong kepada keperluan peralatan berpotensi tinggi bagi memenuhi permintaan manusia dan pasaran. Sistem ini dibangunkan untuk kemudahan manusia disamping dapat mengurangkan kepada penggunaan tenaga manusia memandangkan ia berpotensi untuk dipelbagaikan aplikasinya. Dengan adanya sistem pengecaman plat kereta ini dapat meningkatkan kecekapan system pengawasan kereta. Sistem ini digunakan secara komersial sama ada di dalam mahupun diluar negara. Walaubagaimanpun, di Malaysia penggunaan sistem ini terhad kepada pengecaman plat kereta biasa. Ini bermakna, sistem tersebut tidak dapat menegcam plat kereta yang menggunakan perkataan khas. Pada dasarnya, sistem ini dibangunkan khas untuk mengenal pasti nombor plat kereta Malaysia. Untuk mengenalpasti nombor plat kereta, program pemprosesan imej dibuat dengan menggunakan perisian MATLAB. Sistem ini merupakan sistem online dimana imej plat kereta secara automatik akan diekstrak atau diproses sebaik sahaja gambar kereta diambil. Pertama sekali, imej yang diambil tadi akan ditukar kepada imej binari kemudian ia akan dipotong supaya hanya nombor plat kereta sahaja yang tinggal dan akan melalui proses yang seterusnya. Kemudian warna imej tersebut akan disongsangkan dimana warna hitam latar belakang plat kereta tersebut bertukar menjadi warna putih manakala nombor plat kereta tersebut akan bertukar menjadi warna hitam. Kaedah ini penting kerana proses mengenal pasti nombor plat kereta akan menjadi lebih mudah dengan menggunakan imej binari. Nombor kereta yang telah dikenalpasti itu akan dipaparkan di skrin khas yang dikenal sebagai GUI.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

This chapter explains what is Image Processing and Neural network that is used to develop the 'Automatic Car Plate Recognition System Using Neural Network'. Both elements can be found in MATLAB Toolbox. All this elements are essential parts as a guide to develop the car plate recognition system. This chapter also explains the problem statements of the system, objective of project, project scope and thesis outline.

1.1.1 What is Digital Image Processing

An image can be defined as two-dimensional function ***f***(x,y), where x and y are *spatial* (plane) 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. When x,y and the amplitude values of ***f*** are all finite, discrete quantities, we call the *image digital image*. An image is stored as a matrix using standard Matlab matrix conventions. There are five basic types of images supported by Matlab; *Indexed images, Intensity images, Binary images, RGB images* and *8-bit images* The field of digital image processing refers to processing digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as picture elements, image elements and pixels. Pixels is the term most widely used to denote the elements of a digital image.[1]

Digital image processing allows the use of much more complex algorithms for image processing, and hence can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means. In particular, digital image processing is the only practical technology for: [2]

- Classification
- Feature extraction
- Pattern recognition
- Projection
- Multi-scale signal analysis

Some techniques which are used in digital image processing include:

- Principal components analysis
- Independent component analysis
- Self-organizing maps
- Hidden Markov models
- Neural networks

1.1.2 What is Neural Network

Term **neural network** had been used to refer to a network or circuit of biological neurons. In the other side, neural network refers to artificial neural networks, which are composed of artificial neurons or nodes. Artificial neural networks are made up of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Artificial neural networks may either be used to gain an understanding of biological neural networks, or for solving artificial intelligence problems without necessarily creating a model of a real biological system. An *artificial neural network* (ANN), is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connection approach to computation.

In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network. In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data. An artificial neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. In a neural network model simple nodes, which can be called variously "neurons", are connected together to form a network of nodes — hence the term "neural network". Its practical use comes with algorithms designed to alter the strength (weights) of the connections in the network to produce a desired signal flow.[3]

1.2 Problem Statement

Automatic car license plate recognition (CLPR) system is implemented to help the human to automatically detect plate number without human supervision. Previously, human is needed to observe and list the user car plate number manually. So this project is developing to replace human to monitor the car and automatically capture the image. Besides that, the system can automatically display the status of the car which it will compare between the car plate numbers recognized with the database from JPJ. So we can know either the car is in JPJ observation or not.

1.3 Project Objectives

1. Develop a car license plate recognition system using Image Processing Toolbox and Neural Network Toolbox
2. Integrate between Image Processing and Neural Network

1.4 Project Scopes

This project is to develop a car plate recognition system by using neural network (CLPR).For implementing CLPR system we have use MATLAB Toolbox to achieve the objectives of the project. Thus, the focuses of this project are as below

1. To implement the system in order to recognize the car license plate.
2. To integrate hardware and software.
3. Extract the data from the car plate image by using digital image processing toolbox.
4. Recognize the image of the car license plate by using neural network technique, using a feed-forward network with 3 layers.

1.5 Thesis Outline

Chapter 1 Explain the background of image processing and neural Network, problem statement, objectives of the projects and project scopes all about.

Chapter 2 focuses on the project and literature review about the project that is used as references that helps me in order to finishing my final year project.

Chapters 3 explain and discuss details about digital image processing process and neural network process. In addition, this chapter discusses detail about the method used for this project and some mathematical algorithm applied in the project.

Chapter 4 this chapter will discuss about all result obtained from the system and the limitation of the project. All the discussions are concentrating on the results and overall performance of Car Plate Recognition (CLPR) system.

Chapters 5 discuss the conclusion of development of the whole CLPR system. This chapter also discusses the problem and the recommendation for this project and the overall CLPR system for the future development or modification.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

License plate identification/recognition (LPI/R) is one form of ITS technology that not only recognizes and counts vehicles, but distinguishes each as unique. For some applications, such as electronic toll collection and red-light violation enforcement, LPI/R records a license plates alphanumeric so the vehicle owner can be assessed the appropriate toll or fine. In others, like commercial vehicle operations or secure-access control, a vehicle's license plate is checked against a database of acceptable ones to determine whether a truck can bypass a weigh station or a car can enter a gated community or parking lot. [4]

License plate recognition (LPR) is a new tool for automatic vehicle and traffic monitoring by using digital image processing. For implementing LPR system we have used digital image processing technique and artificial neural network.

The LPR system can be used to traffic control management for recognize vehicles that commit traffic violation, such as entering restricted area without permission ; occupying lanes reserved for public transport, crossing red light, breaking speed limits ; etc.

The purpose for which this system is implemented real time applications, this system is using advance and new techniques of digital image processing such as pattern recognition for recognize characters of license plate and artificial neural network to extract the data.[5]

2.2 MATLAB

MATLAB is a numerical computing environment and programming language. Created by The MathWorks, MATLAB allows easy matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages. Although it is numeric only, an optional toolbox interfaces with the Maple symbolic engine, allowing access to computer algebra capabilities. [6]

MATLAB is built around the MATLAB language, sometimes called *M-code* or simply *M*. The simplest way to execute M-code is to type it in at the prompt, >> , in the Command Window, one of the elements of the MATLAB Desktop. In this way, MATLAB can be used as an interactive mathematical shell. Sequences of commands can be saved in a text file, typically using the MATLAB Editor, as a script or encapsulated into a function, extending the commands available.

This project could be successfully implementing an initial program to recognize car plate using MATLAB. Image Processing Toolbox and Neural Network Toolbox are used to implement the system. [7]

2.3 Digital Image Processing

The paper represents the automatic plate localization component of a Car License Plate Recognition system. The approach concerns stages of preprocessing, edge detection, filtering, detection of the plate's position, slope evaluation, and character segmentation and recognition. Single frame gray-level images are used as the only source of information. [8]

There are four primary algorithms that the software requires for identifying a license plate:

1. Plate localisation – responsible for finding and isolating the plate on the picture
2. Plate orientation and sizing – compensates for the skew of the plate and adjusts the dimensions to the required size
3. Normalisation – adjusts the brightness and contrast of the image
4. Character segmentation – finds the individual characters on the plates

The complexity of each of these subsections of the program determines the accuracy of the system. During the third phase (normalisation) some systems use edge detection techniques to increase the picture difference between the letters and the plate backing. A median filter may also be used to reduce the visual "noise" on the image. [9]

2.4 Neural Network Technology

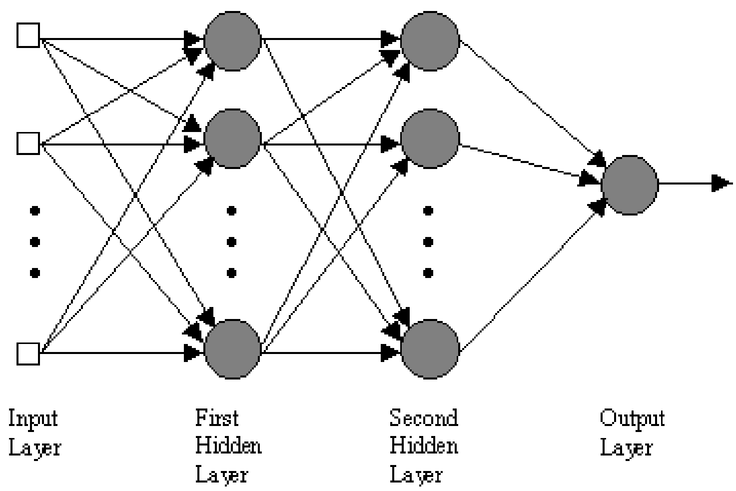

Figure 2.1: Three layer feed –forward Neural Network

Neural networks are data analysis methods and algorithms, indirectly based on the nervous systems of humans and animals.

A typical feed forward network has neurons arranged in a distinct layered topology. The input layer is not really neural at all: these units simply serve to introduce the values of the input variables. The hidden and output layer neurons are each connected to all of the units in the preceding layer. Again, it is possible to define networks that are partially-connected to only some units in the preceding layer; however, for most applications fully-connected networks are better.

When the network is executed (used), the input variable values are placed in the input units, and then the hidden and output layer units are progressively executed. Each of them calculates its activation value by taking the weighted sum of the outputs of the units in the preceding layer, and subtracting the threshold. The activation value is passed through the activation function to produce the output of the neuron. When the entire network has been executed, the outputs of the output layer act as the output of the entire network.[10]

2.4.1 Gathering Data for Neural Networks

Once we have decided on a problem to solve using neural networks, we will need to gather data for training purposes. The training data set includes a number of cases, each containing values for a range of input and output variables. The first decisions you will need to make are: which variables to use, and how many (and which) cases to gather.[11]

Neural networks process numeric data in a fairly limited range. This presents a problem if data is in an unusual range, if there is missing data, or if data is non-numeric. Fortunately, there are methods to deal with each of these problems. Numeric data is scaled into an appropriate range for the network, and missing values can be substituted for using the mean value (or other statistic) of that variable across the other available training cases.

2.4.2 Multilayer Perceptrons

The number of input and output units is defined by the problem. There may be some uncertainty about precisely which inputs to use, a point to which we will return later. However, for the moment we will assume that the input variables are intuitively selected and are all meaningful. The number of hidden units to use is far from clear. As good a starting point as any is to use one hidden layer, with the number of units equal to half the sum of the number of input and output units.

2.4.3 Training Multilayer Perceptrons

Once the number of layers, and number of units in each layer, has been selected, the network's weights and thresholds must be set so as to minimize the prediction error made by the network. This is the role of the training algorithms. The historical cases that have been gathered are used to automatically adjust the weights and thresholds in order to minimize this error. This process is equivalent to fitting the model represented by the network to the training data available. The error of a particular configuration of the network can be determined by running all the training cases through the network, comparing the actual output generated with the desired or target outputs. The differences are combined together by an error function to give the network error. [12]

Training multilayer process is done by using model setup as shown below.

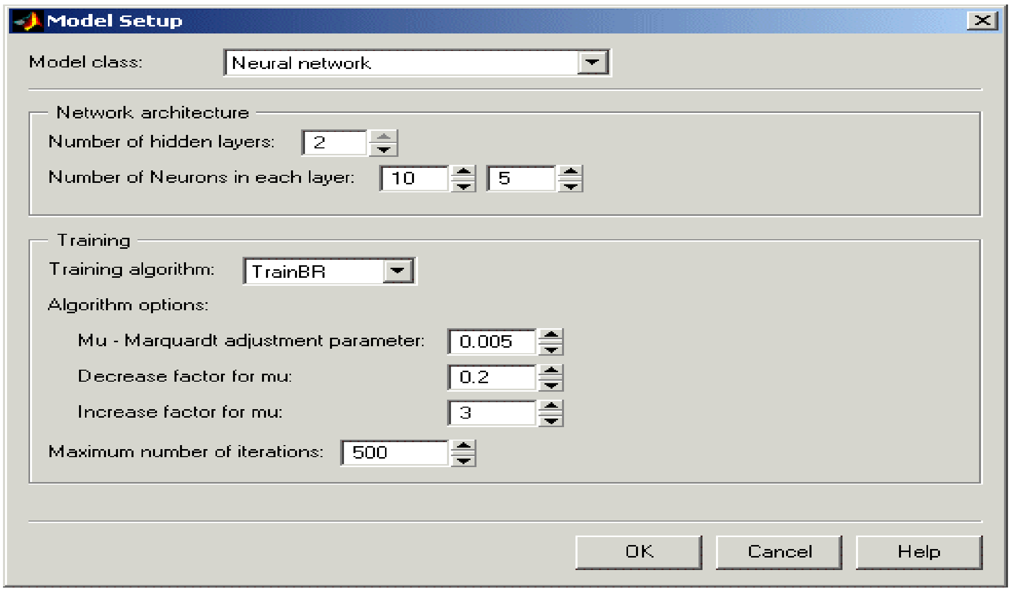

Figure 2.2: Neural Network Training Model Setup

The most common error functions are the sum squared error (used for regression problems), where the individual errors of output units on each case are squared and summed together, and the cross entropy functions.

Neural networks contain no preconceptions of what the model shape will be, so they are ideal for cases with low system knowledge. They are useful for functional prediction and system modeling where the physical processes are not understood or are highly complex. The disadvantage of neural nets is that they require a lot of data to give good confidence in the results, so they are not suitable for small data sets. Also, with higher numbers of inputs, the number of connections and hence the complexity increase rapidly.

MBC provides an interface to some of the neural network capability of the Neural Network Toolbox. Therefore these functions are only available if the Neural Network Toolbox is installed. Urig the training process,the Square Mean Error (SME) is calculated to determine the accuracy of the system. The result of the training process is as shown below. Now call train: [13]

[net,tr] = train(net,Pseq,Tseq);

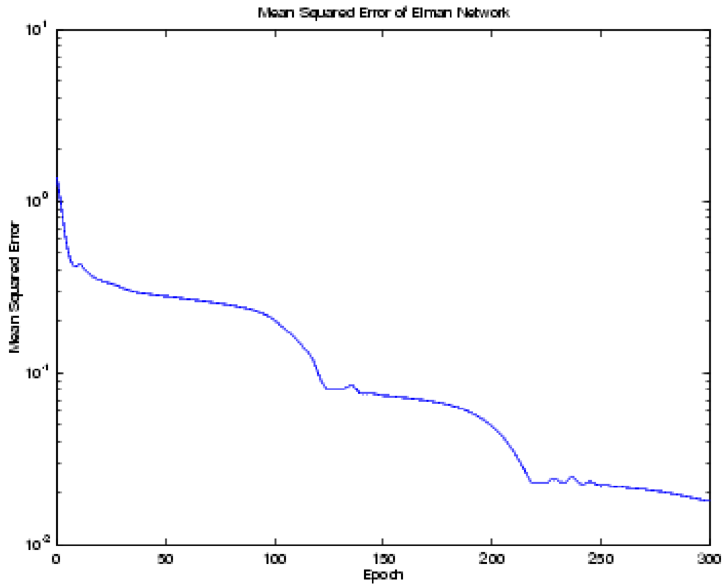

Figure 2.3 : Square Mean Error of Training Process

2.4.4 Neural Network Architecture

This network is sometimes called a MADALINE for Many ADALINES. Note that the figure on the right defines an S-length output vector a. The Widrow-Hoff rule can only train single-layer linear networks. This is not much of a disadvantage, however, as single-layer linear networks are just as capable as multilayer linear networks. For every multilayer linear network, there is an equivalent single-layer linear network.[14]

The ADALINE network shown below has one layer of S neurons connected to R inputs through a matrix of weights W

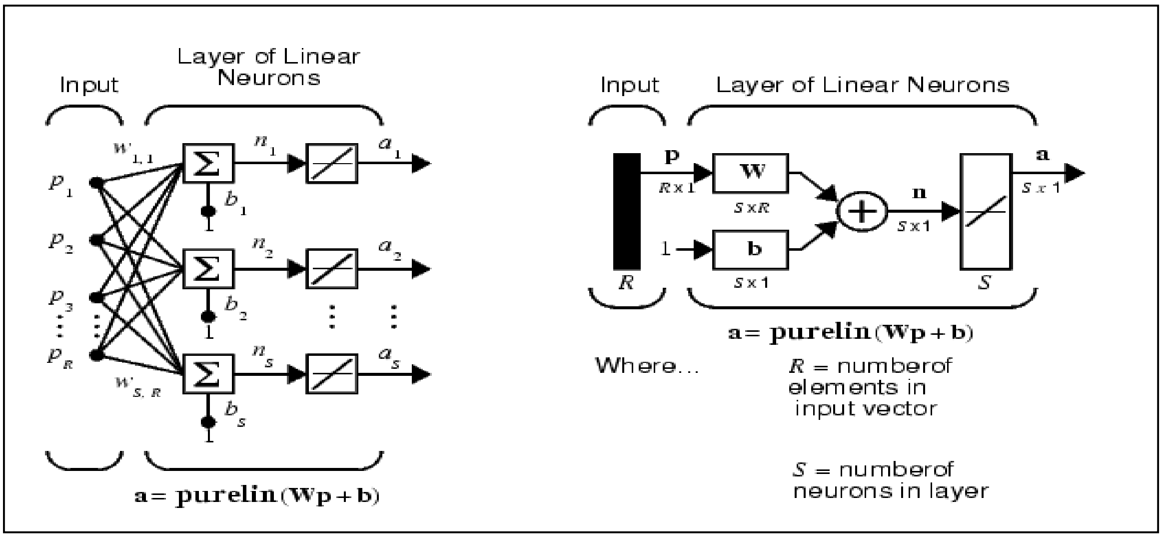

Figure 2.4: ADALINE Network

2.5 Graphical User Interface (GUI)

Graphical user interface (GUI) is a type of user interface which allows people to interact with electronic devices like computers, hand-held devices (MP3 Players, Portable Media Players, Gaming A devices), household appliances and office equipment. A GUI offers graphical icons, and visual indicators as opposed to text-based interfaces, typed command labels or text navigation to fully represent the information and actions available to a user. The actions are usually performed through direct manipulation of the graphical elements. [15]

2.6 Waterfall Methodology

Waterfall methodology is used to implement the software of the system. The flow of the methodology is as shown below.

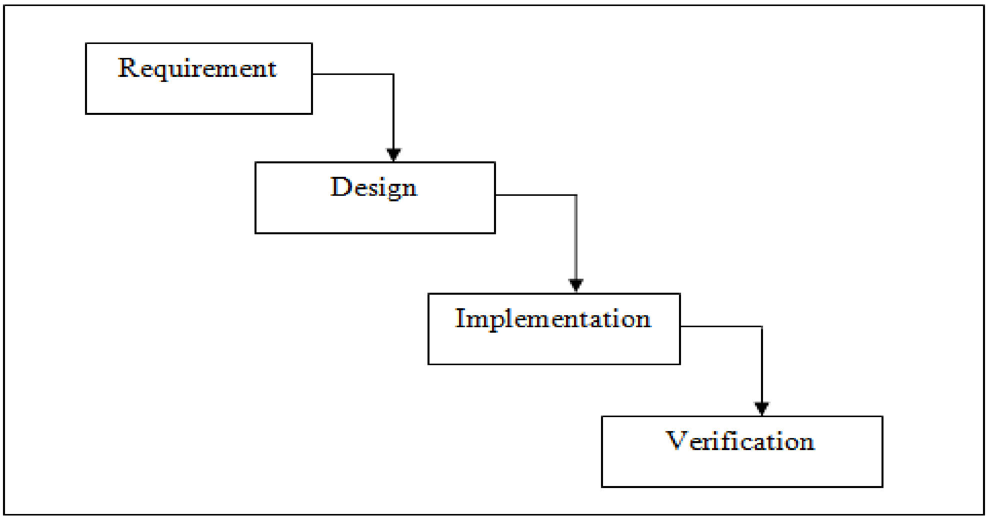

Figure 2.5: Waterfall Methodology

2.6.1 Requirement

CLPR system is implemented by using hardware and software that has their own specification suitable for the system. Logitech Quickcam 1.3 megapixel is used to capture the car image and Image Processing Toolbox in MATLAB is used to design the software of the system. Both of them are important requirement to make the project succeed. The webcam is integrated with the system by using specific command so that they can be interface to complete the system.

2.6.2 Design

The first important thing before implement the project is designing and planning. This part is very important so that we can show the overall flow of the system. The planning is needed so that we can design the system suitable for the application. For CLPR system, the design should be focuses on two important parts that are software and hardware part.

2.6.3 Implementation

The implementation is done after finishing the design of hardware and software. The implementation is important to make sure it is working properly. A special command should be applied to connect the hardware and software.

2.6.4 Verification

Verification is the last step after implementation the software and hardware. Verification is important to make sure that the system is well function and obtain the desired output.

2.6.5 Optical Character Recognition (OCR)

Optical Character Recognition (OCR) is the process of converting scanned images of machine-printed or handwritten text including numerals, letters, and symbols into a computer-processable format; also known as optical character recognition (OCR). A typical OCR system contains three logical components: an image scanner, OCR software and hardware, and an output interface. The image scanner optically captures text images to be recognized. Text images are processed with OCR software and hardware. The process involves three operations: document analysis which extracting individual character images, recognizing these images based on shape, and contextual processing either to correct misclassifications made by the recognition algorithm or to limit recognition choices. The output interface is responsible for communication of OCR system results to the outside world.[16]

Commercial OCR systems can be grouped into two categories:

1. Task-specific readers;

A task-specific reader handles only specific document types. Some of the most common task-specific readers read bank checks, letter mail, or credit-card slips. These readers usually utilize custom-made image-lift hardware that captures only a few predefined document regions. For example, a bank-check reader may scan just the courtesy-amount field where the amount of the check is written numerically and a postal OCR system may scan just the address block on a mail piece. Such systems emphasize high throughput rates and low error rates.

2. General-purpose page readers;

General-purpose page readers are designed to handle a broader range of documents such as business letters, technical writings, and newspapers. These systems capture an image of a document page and separate the page into text regions and nontext regions. Nontext regions such as graphics and line drawings are often saved separately from the text and associated recognition results. Text regions are segmented into lines, words, and characters, and the characters are passed to the recognizer. Recognition results are output in a format that can be postprocessed by application software. Most of these page readers can read machine-written text, but only a few can read hand-printed alphanumerics.[17]

Table below shows some example of the commercial OCR software that is widely used today

Name	Operating System	Notes
Zonal OCR	Windows	Zonal OCR is the process by which Optical Character Recognition (OCR) applications "read" specifically zoned text from a scanned image. Many batch document imaging applications allow the end user to identify and draw a "zone" on a sample image to be recognized. Once the zone has been established on the sample image, this zone will be applied to each image processed so that the data can be extracted from the image file and converted to a ASCII format.
Microsoft Office Document Imaging	Windows, Mac OS X	
Microsoft Office OneNote 2007	Windows	
SILVERCODERS OCR Server	Linux	Server side system, multi language, very good recognition quality, can save text formatting and recognizes complicated tables of any structure
Computhink's ViewWise	Windows	Document Management system

Table 2.1: OCR Software

CHAPTER 3

METHODOLOGY

3.1 Introduction

In developing a project, methodologies is one of the most important element to be consider to make sure that the development of the project is smooth and get the expected result. A good methodology can described the structure or the flow of the project whereby it can be the guideline in managing it. For the image processing program the system required various methodologies to process he desired image. The methodology include convert the image, crop the region of interest (ROI), dilation method, erosion method, threshold, compliment the image and applied some parameters to calculate the value of area boundaries to distinguish each character of number in the image.

The second part is by using the fed forward Neural Network that consists of three layers that are input layers, hidden layers and output layers. The inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed then processed by a nonlinear function (usually the hyperbolic tangent). As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output

3.2 System Design

System design is the first step in software development that needs careful and intricate planning. It helps us to prepare detailed technical design of application based system. It provides the specification and design for showing flow of work, program and user function. The system is begin by design the flow of the system from starting until ending of the program. The flow of the system should be clear to make sure that the system implementation can be done smoothly and can functioning as desired. Then the block diagram is create to describe the part content in the system. Every part has their own function. The flow of block diagram is also important because and it is related to each other.

Block Diagram of CLPR System

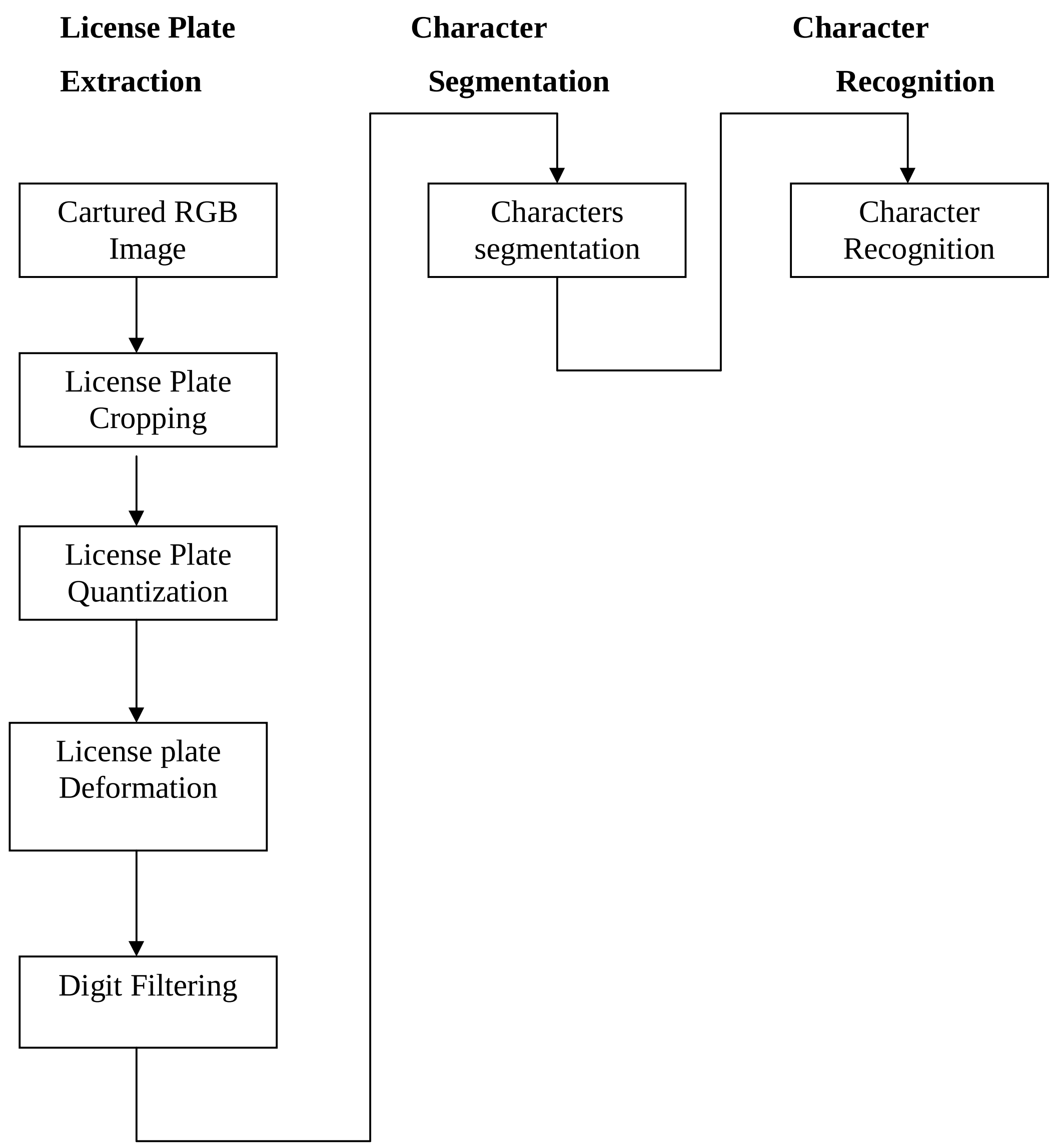

Figure 3.1: Block Diagram of CLPR System

Since the different steps are strongly inter-related, each step of analysis, if the results are incoherent, it is possible to go back to the previous steps in order to modify parameters.

Figure 3.1 below shows the block diagram of methodology of this project which includes:

PHASE 1 : IMAGE ACQUISITION SYSTEM

1. Image Cropping
2. Image Quantization
3. Plate Deformation
4. Digit Filtering
5. Character Segmentation
6. Character Recognition

PHASE 2: System design

PHASE 3: Graphical User Interface (GUI) design

PHASE 4: Integrate hardware and software

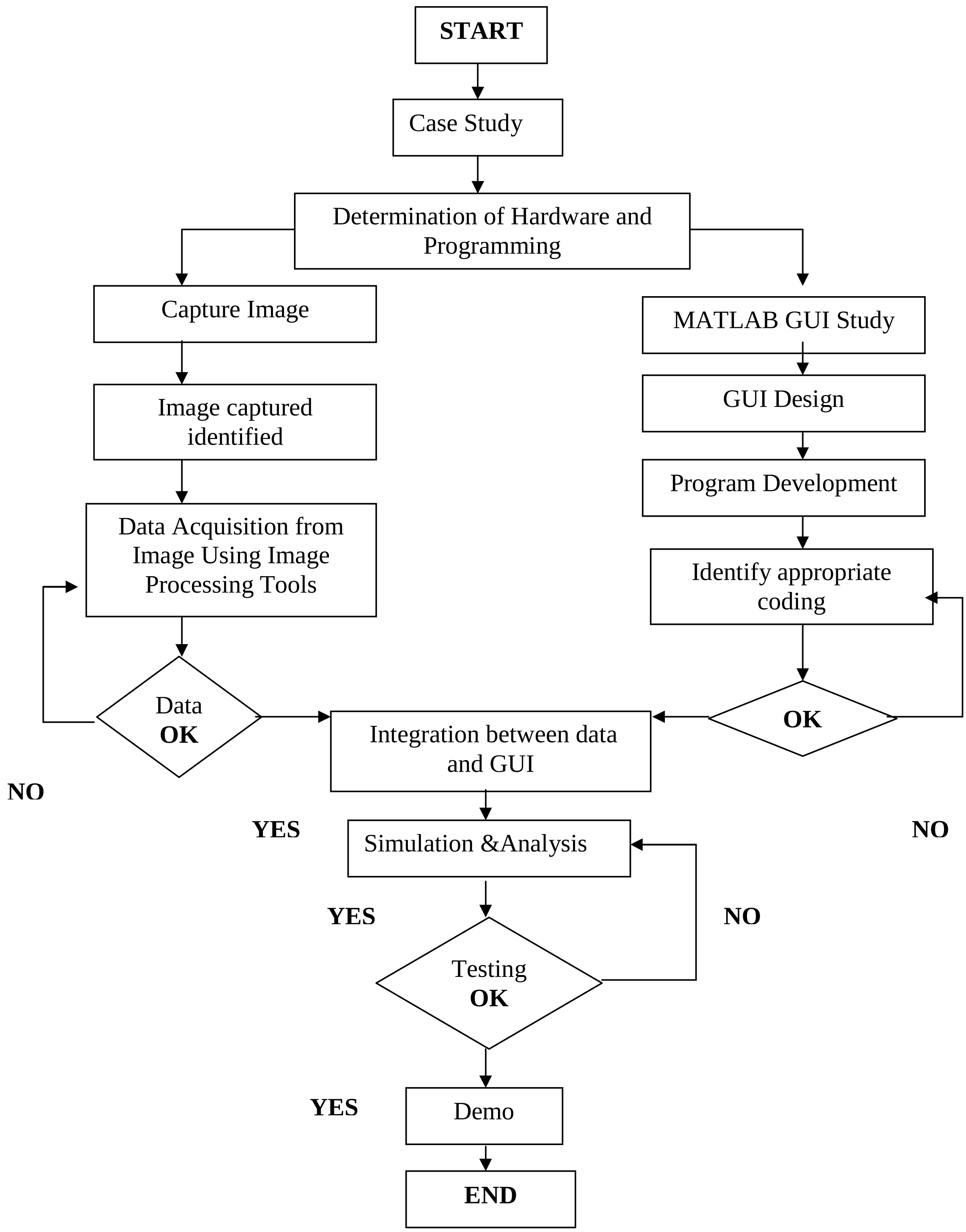

Figure 3.2: Flowchart of CLPR System

3.3 Image Acquisition System

A car plate number is tested in this project that is captured by using a webcam with 640 x 480 pixels (Logitech Quickcam).The image were taken under natural daylight lightning condition or with lightning from the pendaflour lamp. For the experimental and result, both condition of lightning were considered.

Figure 3.3: Image under lighting

3.3.1 License Plate Cropping

The first stage in the recognition process is the extraction of the smallest rectangle that contains the license plate edges, and thus the license plate itself. The importance of this stage is in speeding up the further processing of the image, which has no interest in the plate’s surroundings.

3.3.2 License Plate Quantization

Image quantization is the formation of a binary image from the given RGB colored image. This stage is very critical for the separation of the digit images from the plate. Morphological operations are applied in order to convert RGB image to binary image.

3.3.3 Plate Deformation

A key issue involved in the recognition of objects in a picture shot by a camera, is the fact that the camera may be positioned anywhere, and the objects may be viewed in different angles. In the case of a license plate, this can result in different digit sizes, and thus creating problem dealing with not only different digits, but with different digit sizes. To overcome this difficulty, an integral part of the algorithm is the plate deformation. In basic, this stage turns an image with 4 coordinates to a standard size rectangle. The deformation is done using simple algorithm, which maps for each pixel in the deformed image.

3.3.4 Digit Filtering

Basically, digit filtering stage is done to filter out everything which is not a digit, and leave only digit in the picture. The process that is involved during the digit filtering is removing small connected components that are unnecessary. Each component is first labeled and the digits components will be leaved while the non-digits components, will then removed.

3.3.5 Character Segmentation

Another process that is involved is Identifying and separating cluttered digits or known as segmentation. After that, the normalized digits image will be extracting. This is the final stage before the most important stage in the algorithm.

3.3.6 Character Recognition

Character recognition is actually used optical character recognition technique that is aimed to classify optical patterns corresponding to alphanumeric or other character. It is translation of car plate images captured by a webcam into machine-editable text.OCR requiring minimal training where the input should be in perfect shape. Before OCR can be used, the source material must be scanned using an optical scanner to read in the page as a bitmap (a pattern of dots). The system then differentiates between images and text and determines what letters are represented in the light and dark areas of car plate image.

3.4 Image Processing Techniques

In the CLPR system, image processing is the main program used to implement the system. They are several techniques involve in image processing and the important one is Morphology method. It is a broad set of image processing operations that process images based on shapes. The most basic morphological operations are erosion and dilation. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors by choosing the size and shape of the neighborhood since it is sensitive to specific shapes in the input image. Dilation and erosion are two fundamental morphological operations. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image.

3.4.1 Converting Images

Converting image is the process of converting a captured RGB image into a grayscale image. This method is choosing to make the image easily processed because RGB color quite difficult to determine the noise since it is a combination of various color. It is an important step in many image analysis system including document image processing.

3.4.2 Crop the Region of Interest (ROI)

The crop function will be used to extract a rectangular portion of car plate image. There are two conditions to perform cropping step that are crop image manually and determine the specify coordinates of a rectangle that defines the crop area. If we call imcrop function without specifying the crop rectangle, we can specify the crop rectangle interactively and imcrop function draws a rectangle around the area we are selecting and a new image from the selected region will be created. While the image will automatically cropped if we already determine the width and height elements of the images.

3.4.3 Morphology

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. The most basic morphological operations are dilation and erosion. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, we can construct a morphological operation that is sensitive to specific shapes in the input image.

3.4.4 Erosion Method

The value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0. Erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. The effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels. *Grayscale* erosion will generally darken the image. Bright regions surrounded by dark regions shrink in size, and dark regions surrounded by bright regions grow in size. Small bright spots in images will disappear as they are eroded away down to the surrounding intensity value, and small dark spots will become larger.

3.4.5 Dilation Method

Dilation used an OR operation to combine the images. Dilation is one of the two basic operators in the area of mathematical morphology, the other being erosion. It is typically applied to binary images, but there are versions that work on grayscale images. The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (*i.e.* white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller. In dilation operation, the value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

3.4.6 Threshold

From a grayscale image, thresholding can be used to create binary images. For this project the plate's background is dark, and the characters are bright. This property turned out to be the best way for separating the digits: turning the image to gray scale. For the formation of the binary image, the darker and the brighter areas must be seperated. If we assume there is some kind of constant threshold, which could tell whether a pixel belongs either to the darker area, or the brighter one, then we could use this number to reach the desired binary image.

3.5 Software and Hardware Integration

Software and hardware integration is the most important part in implement the system since the system is using the online forming to recognize the car plate number. An appropriate coding is needed to interface between webcam used and the hardware. The '*imaqhinfo*' function will give the correct info for the devices that installed. For this project, the device installed is '*winvideo*' type that is for USB port.The webcam will automatically capture the car plate image by using '*getsnapshot*' function.

3.5.1 Software Items

To implement the Car License Plate Recognition (CLPR) system, a standard software are used as shown in the table below.

Item Type	Description	Version
Operating System	Microsoft Window XP	Professional, Version 2002, Service Pack 2
Desktop	Microsoft Office 2007 MATLAB 7.1	Office Professional Version 7.1 release 14 pack 3

Table 3.1: Software Items

3.5.2 Hardware Item

This system can be operated by using specific hardware with specific requirement. Below is the list of hardware used to implement the system with its minimum requirement.

Item Type	Minimum Requirement
Laptop	Intel (R) Pentium (R) 1.86 GHz 252 MB of RAM
Webcam (Logitech Quickcam E 3500 Plus)	VGA resolution : 640 x 480 pixels Live Capture :Up to 800 x 720 pixels Photos :Up to 1.3 megapixels Video Capture :Up to 30 frames per

	second Interface : USB port

Table 3.2: Hardware Items

3.6 Graphical User Interface (GUI) Design

Graphical User Interface (GUI) is used to display the output of the system on the screen. First the GUI will integrate with the image processing by using special command. The main reason GUIs are used is because it makes things simple for the end-users of the program. If GUIs were not used, we would have to work from the command line interface, which can be extremely difficult. For the CLPR system, the car plate number is displayed together with the status of the car. The first step to create a new GUI by using MATLAB is as shown in the figure below.

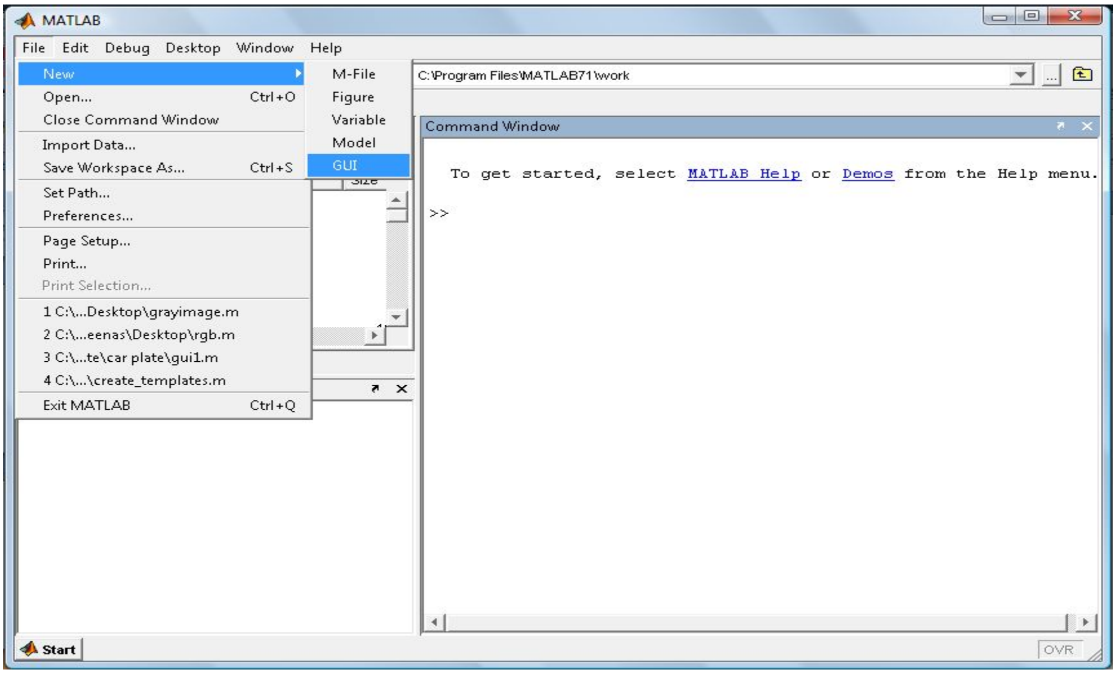

Figure 3.4: MATLAB Command Window

GUI can also can be created by type 'guide' in the MATLAB command window and it will activate a blank GUI frame to allow the user to visually design the GUI as he desires.

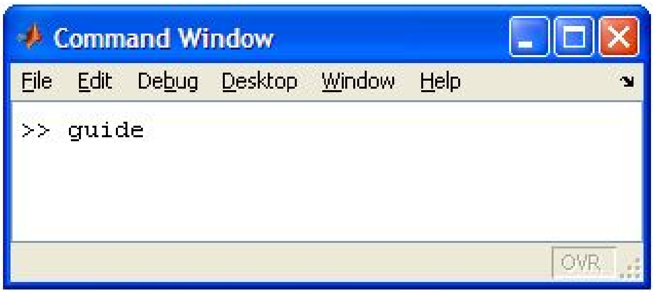

Figure 3.5: Guide command window

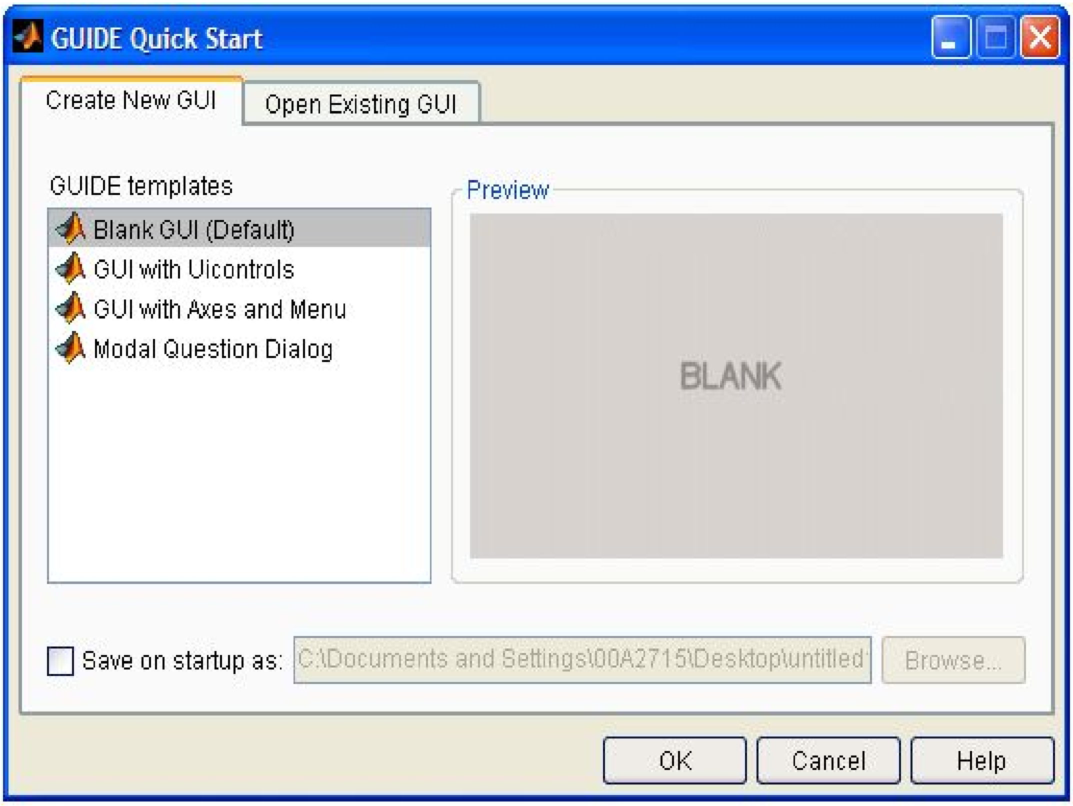

Figure 3.6: GUI GUIDE Quick Start

Figure above show the GUIDE Quick Start screen and choose Blank GUI to create a new GUI or we can click button 'Open Existing GUI' if we already create it before. Then click at 'Save on start up as' to save the GUI file at the desired location.

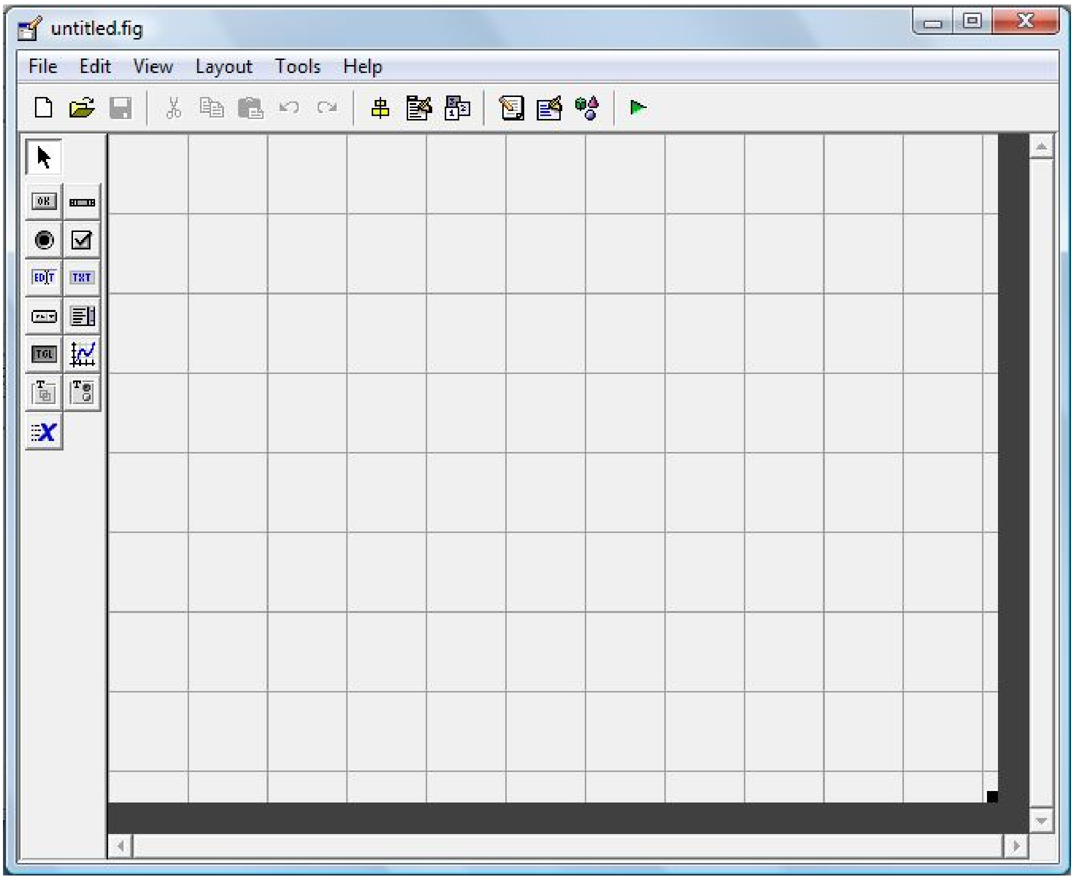

Figure 3.7: GUI Design Screen

The user can now add as many code activating devices (i.e. buttons, edit boxes, popup menus, etc.) as he wishes. These items are located on the left panel on the GUI frame. Then MATLAB automatically generates an .m file to go along with the figure or activating devices that we just put together. The .m file is where we attach the appropriate code to create the desired GUI.[18]

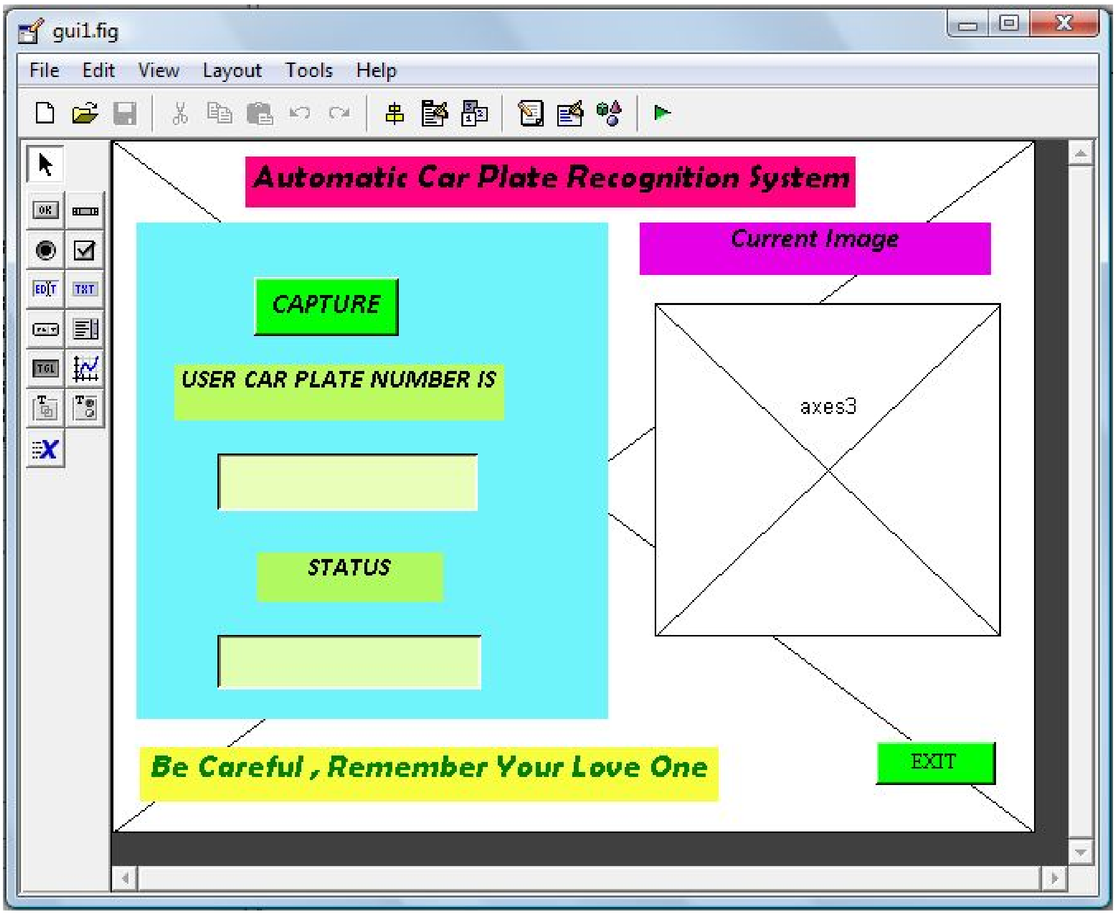

Figure 3.8: GUI with code activating devices

CHAPTER 4

RESULT AND ANALYSIS

4.1 Introduction

The Car License Plate Recognition (CLPR) system has been designed special for Malaysian car plate characters. For this project, three car plates are prepared and the system is trained to recognize them. A car parking entry prototype is designed and a sample car is used.

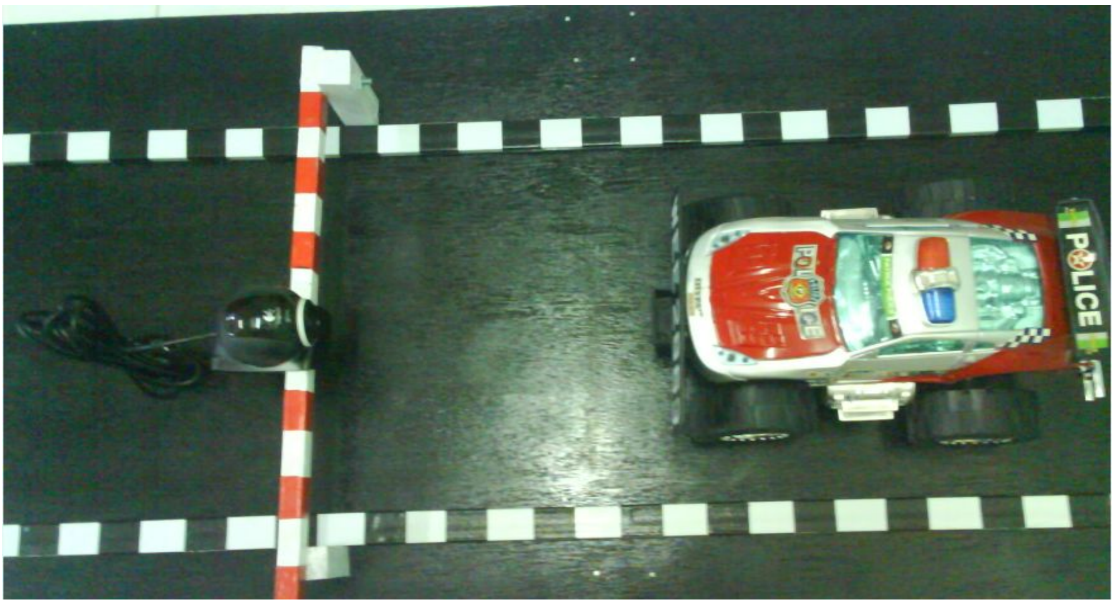

Figure 4.1: Parking Entry Prototype

SAMPLES OF CAR PLATE NUMBER


Figure 4.2: Car Plate Samples

For this project, there are three car plate samples used. The size of the car plate is actually not an actual size since it is just a prototype but its characteristic is same as the real one.

4.2 Image Processing Techniques

CLPR system is totally implemented using software. This project is actually combination of Optical Character Recognition and Image Processing Toolbox. Both method are important and need well planning. This subchapter will discuss the result obtained in image processing part.

4.2.1 Result of Captured Image

Logitech Quickcam 1.3megapixels is used to capture the car plate image. There are two type of image sources used in plate detection that are video image and static image. For this project, static image is choosing because it is more relevant for parking entry application and it is involve simple step to process it compare to the video images. Figure 4.3 shows the info for the webcam used to capture the image.

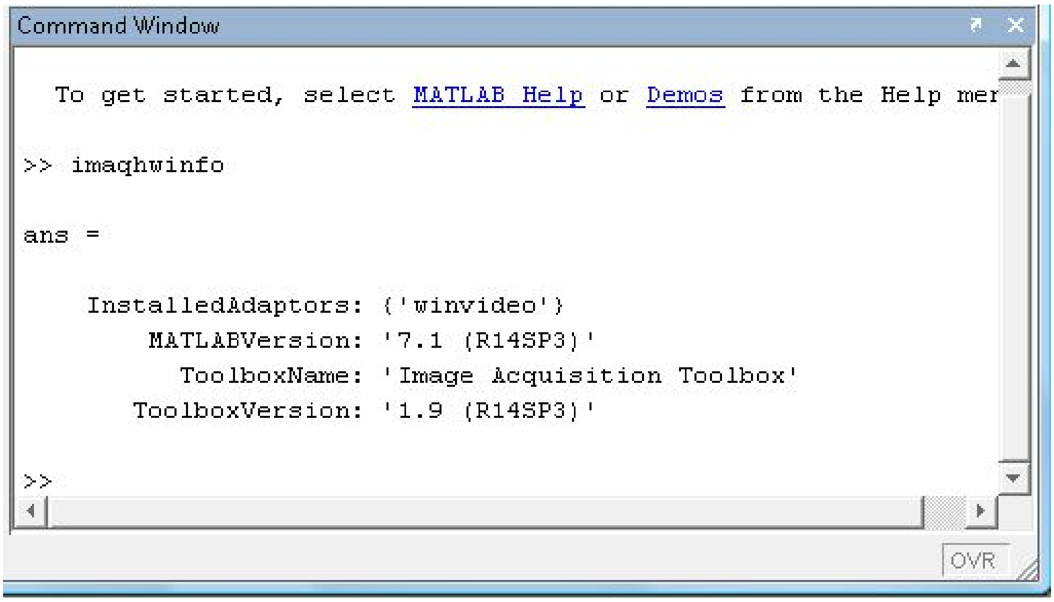

Figure 4.3: Info for Webcam

The ‘*imaqhwinfo*’ is a structure that contains information about the image acquisition adaptors for ‘*winvideo*’ type that is available on the system. An adaptor is the interface between MATLAB and the image acquisition devices connected to the system. The ‘*winvideo*’ is for USB port devices such as the webcam used with the specification shown in the Table 3.2.

To integrate between webcam and laptop, a special command is used as shown in figure 4.4 below.

```
clc, close all, clear all      %clear previous command
gmb=videoinput('winvideo');    %get image from webcam
pic=getsnapshot(gmb);          %snap the image
image(pic);                    %show the image captured
```

Figure 4.4 : MATLAB function to capture the image

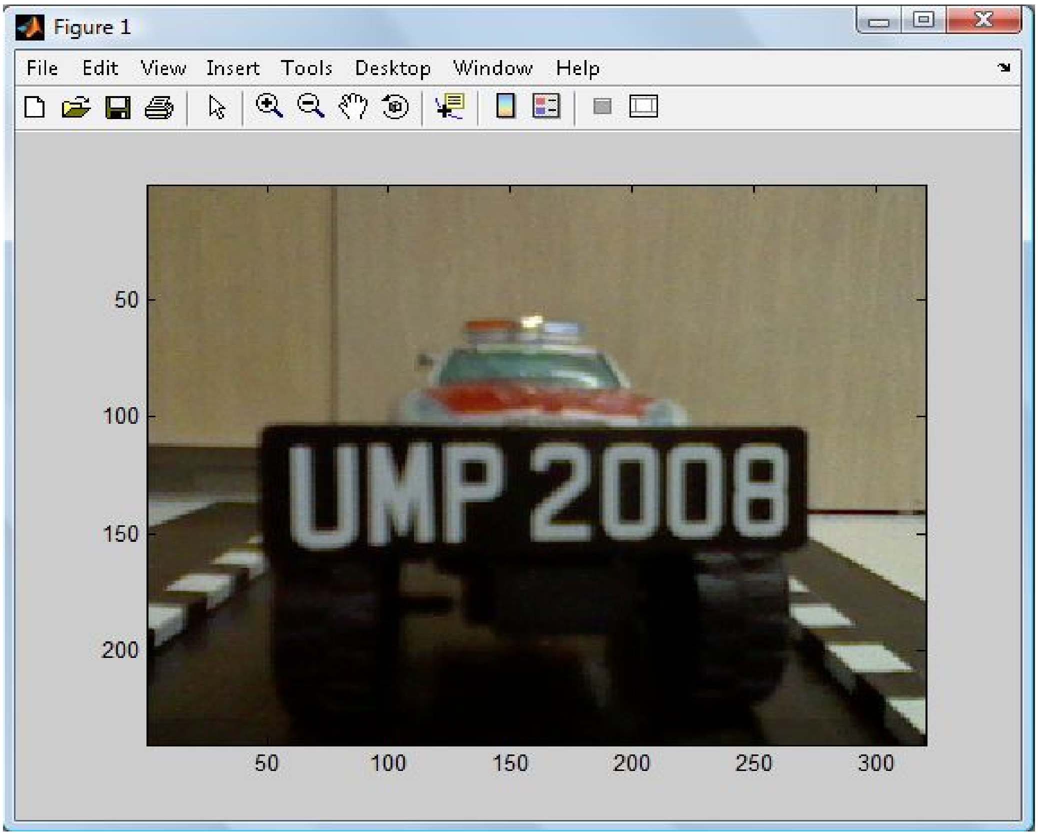

Figure 4.5: Image captured using webcam

4.2.2 Convert RGB Image to grayscale image

The original image captured is in RGB format and it need to be converted into grayscale so that the next process is become easier. With RGB colour, it is quite difficult to determine the noise because it is combined with various colours.'*rgb2gray*' converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

```
I=rgb2gray(pic);             %convert the RGB image to grayscale
figure,imshow(I);            %show the image
```

Figure 4.6: MATLAB function to convert RGB image to Grayscale image

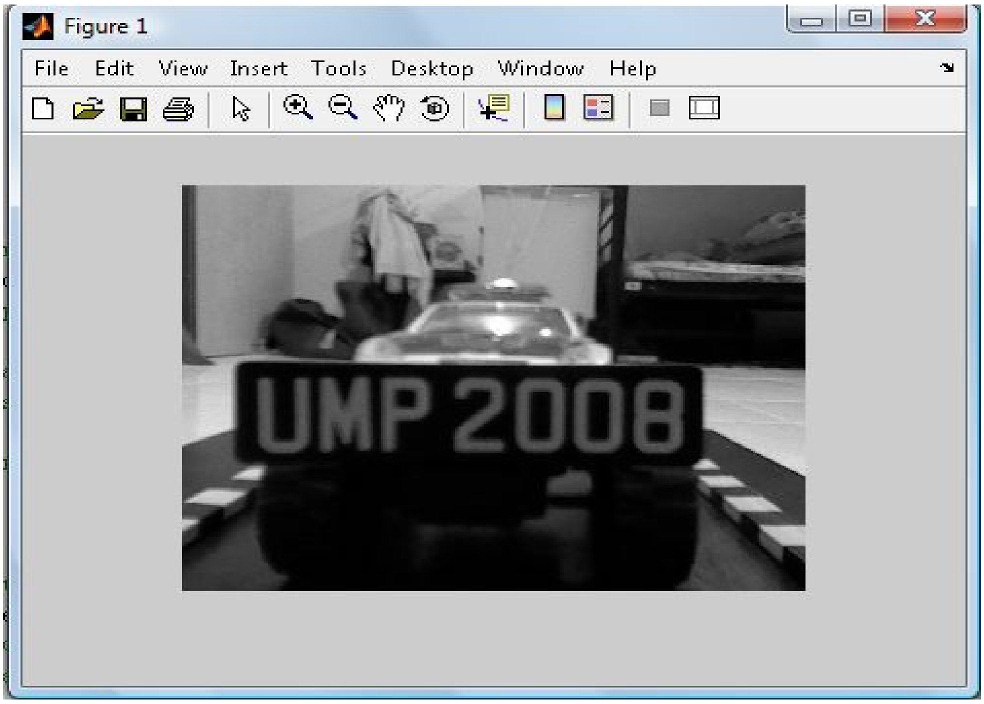

Figure 4.7: The grayscale image

4.2.3 Image Crop

Image crop is a process of extraction of the smallest rectangle that contains only the license plate edges. The importance of this stage is to make further processes easier without counting the surrounding area of the car plate. There are two conditions to perform cropping step that are crop image manually and determine the specify coordinates of a rectangle that defines the crop area.

```
I1=imcrop(I,[35 111 227 51]);%crop the    image with the
                               specific    coordinate

imshow(I1);                   %show the image crop
```

Figure 4.8: MATLAB function for crop image

Figure 4.9: Crop image

4.2.4 Opening Function

Morphology Open is just an erosion operation followed by a dilation operation. Morphological opening is used to remove small objects from an image while preserving the shape and size of larger objects in the image. The erosion function is used to filter all noise from the original car image, creating an output image that contains only the rectangular shapes of the car plate character. The erosion function will make the image become smaller. Then the process is followed by dilation which the image will dilate to make the erode image become bigger.

```
se=strel('disk',1);        %define the structuring element
o=imopen(R,se);            %opening method
```

Figure 4.10: MATLAB function for Opening function

Figure 4.11: Opening Image

4.2.5 Threshold Function

Threshold function is used to converts the grayscale image to a binary image. The output image `BW` replaces all pixels in the input image with luminance greater than level with the value 1 (white) for car plate characters and replaces all other pixels with the value 0 (black) that is the background of the car plate. The `level` is set in the range [0,1], regardless of the class of the input image. The function '*graythresh*' can also be used to compute the `level` argument automatically if we do not specify their level.

```
level=graythresh(p);   %threshold the image
BW=im2bw(p,level);     %convert from grayscale to binary
```

Figure 4.12: MATLAB Threshold function

Figure 4.12: Thresholding Image

4.2.6: Imcomplement Function

Imcomplement function is performed to get better image of plate characters which zeros (black) become ones (white) and ones become zeroes. In the output image, dark areas become lighter and light areas become darker. Imcomplement function is used because to get the better shape of character since Optical Character Recognition is performed by comparison concept. The car plate characters will compared to the scanned source image that is saved as bitmap image.

```
imagen=imcomplement(BW);   %complement the image
```

Figure 4.14: Imcomplement Function

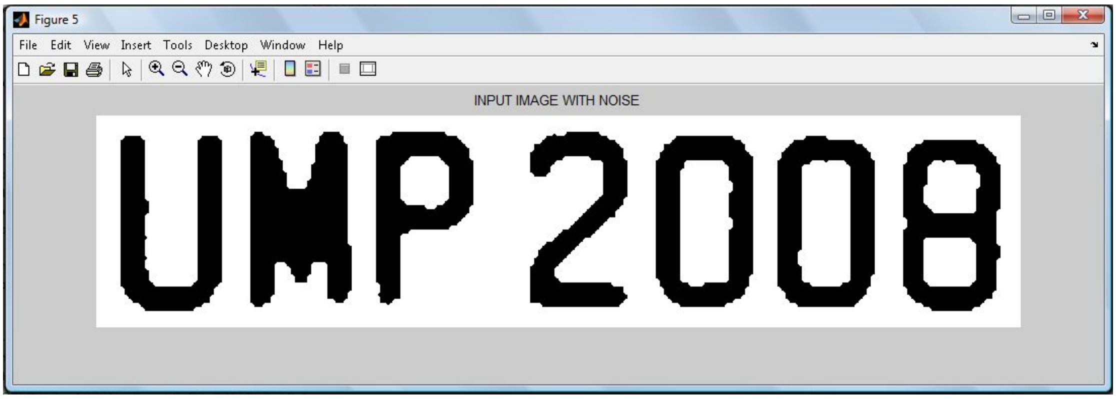

Figure 4.15: Imcomplement Image

4.3 Result of Optical Character Recognition (OCR)

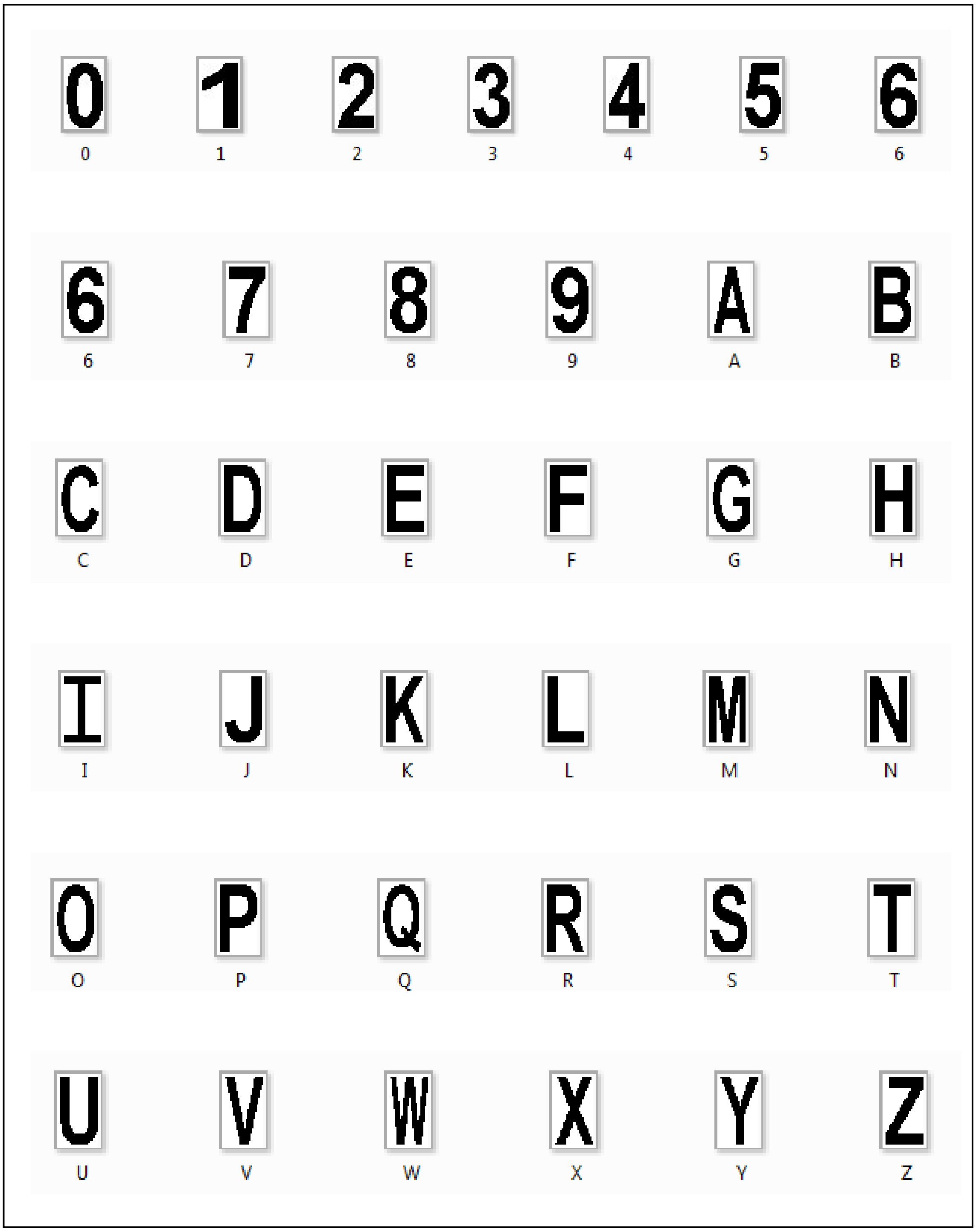

Figure 4.16: Scanned Bitmap image

OCR is the recognition of printed or written text characters by a computer. This involves photoscanning of the text character-by-character, analysis of the scanned-in image, and then translation of the character image into character codes such as ASCII.

Before OCR can be used, the source material must be scanned using an optical scanner or a specialized circuit board in the PC to read in the page as a bitmap like a pattern of dots. The concept of OCR is like pattern recognition and requiring a minimal training to make sure that the system can recognize the characters accurately. The input image should be in perfect shape.

4.4 Graphical User Interface (GUI) Representation

GUI is used to display the output image and the car plate characters recognized. Figure 4.17 shows the GUI created for this project.

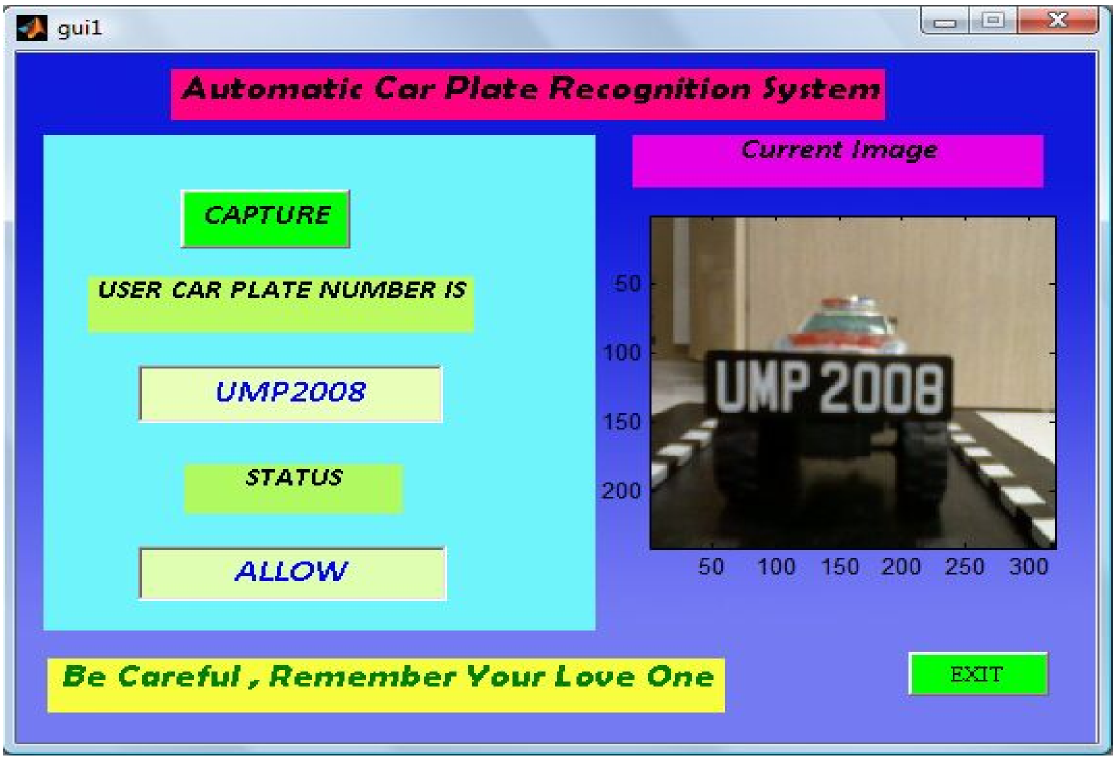

Figure 4.17: Graphical User Interface (GUI)

Figure 4.18: Exit Window option

‘CAPTURE’ button is used to capture the image of the car that enter the private parking. Then the image will show at the ‘current image’ screen and it will always update every time the capture button is clicked and always show current image. Then the car plate number that captured by the webcam will be analyzed and compared to the scanned alphabetic letter and numeric digits. Then it will display at the set space. Exit button is used to exit the program without close the MATLAB window. Exit window option is used as confirmation if the users want to exit the program.

4.5 Training Result

A simple training is done to measure the accuracy of the system. The system is run for thirty times for three samples of the car plate and the percentage is calculated by using simple mathematical operation. The training process can help the system to recognize the car plate characters accurately. The image should be in perfect shape to make sure that the car plate character recognized the same as the sample of scanned images. They are many disturbances that should not take for granted to implement the system to make sure that the system can perform as desired. Figure 4.19 show the training result and percentage of accuracy for every car plate sample.

	WEE 2863	UMP 2008	NAS 5479
1	WEEZ883	UMP 2008	NAS 5478
2	WEE2863	UMP 2008	NAS 5479
3	WEE2863	UMP 2008	NAS 5479
4	WEE2883	UMP 2008	NAS 5478
5	WEE2883	UMP 2008	NAS 5478
6	WEE2863	UMP 2008	NAS S479
7	WEEZ863	UMP 2008	NAS S478
8	WEE2883	UMP 2008	NAS 5479
9	WEE2883	UMP 2008	NAS 5478
10	WEE2863	UMP 2008	NAS 5479
11	WEE2883	UMP 2008	NAS 5478
12	WEE2883	UMP 2008	NAS 5479
13	WEE2863	UMP 2008	NAS 5479
14	WEE2863	UMP 2008	NAS 5478
15	WEE2883	UMP 2008	NAS 5479
16	WEE2883	UMP 2008	NAS 5478
17	WEE2863	UMP 2008	NAS 5478
18	WEE2863	UMP 2008	NAS 5479
19	WEE2883	UMP 2008	NAS 5478
20	WEE2883	UMP 2008	NAS 5479
21	WEE2883	UMP 2008	NAS 5479
22	WEEZ863	UMP 2008	NAS 5479
23	WEE2883	UMP 2008	NAS 5478
24	WEE2863	UMP 2008	NAS 5479
25	WEE2883	UMP 2008	NAS 5478
26	WEE2863	UMP 2008	NAS 5479
27	WEE2883	UMP 2008	NAS 5478
28	WEE2883	UMP 2008	NAS 5479
29	WEE2863	UMP 2008	NAS 5478
30	WEE2863	UMP 2008	NAS5478
PERCENTAGE ACCURACY	**43.33%**	**100%**	**50.0%**

Table 4.1: Training Result

4.6 Discussion

From the result obtained, we can see that the system is not too stable. It cannot recognize certain alphabetic letter and numeric digit correctly especially the characters that have quite similar shape. This problem occurs because of the image captured is blur. This situation can be approving at the last step during the imcomplement step. We can see that each character is not in perfect shape. As explained before, to get the better result, the input image should be in perfect shape. To overcome this problem, the high resolution webcam should be used to get the perfect input image and obtained the desired output.

4.7 Costing and Commercialization

Items	**Unit**	**Price/unit**	**Total Price**
Logitech Quickcam	1	RM100.00	RM100.00
Car Plate (sample)	4	RM4.00	RM12.00
Car (sample)	1	RM13.90	RM13.90
TOTAL			**RM125.90**

Table 4.2: Items Cost

Table 4.2 shows the items used to design the prototype of the system. In this project, Logitech Quickcam is the main item used to capture the car plate image. This system is testing by using three samples of car plate which the size of each car plate is same but it is not the actual size of Malaysian car plate. A sample of car is used together with the car plates.

Car plate recognition (CLPR) system is already used several years ago. It is implemented by using many techniques depend on the applications. Different country has different system because of the car plate characters. So this project actually has high potential to be commercialized but need to further development so it has maximum accuracy.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

The purpose of this project is to develop a system that can recognize the car plate and display at GUI. From the result obtained, the system can recognize the character of car plate number but sometime there is confusion between the characters that has quite same shape. This is because of some noise and other disturbances occur during perform the certain process when implement the system. The weakness of this system is sometime it is not too stable causes the output display is not the same as the image of car plate number captured. Some solutions has been identified to overcome the weaknesses of the system and will be explain in the next subchapter.

5.2 Future Recommendations

Automatic car license plate recognition (CLPR) system is an advance system used commercially in many countries including Malaysia. So this system needs further improvement to make sure that it becomes perfect system. Some recommendation is identified to overcome the problems.

To make the system more accurate, the system actually can be implemented by using image processing and feed forward neural network. This technique is choosing because neural network is capable to perform complex signal processing and classification tasks with real-world data. The car-plate recognition is a relatively complex task, for the variety of environment and targets, and the ordinary presence of disturbing elements.

The car plate recognition system should be implemented in a real time set up. This would make the system suitable for commercialization. Besides that, the system can also be improvised to incorporate the various car plates in order to increase the range for the functionality of the system.

Finally, most important of all, work can be done to expand the application of car plate recognition system to car plates with much more complicated such as Japanese car plates and Chinese car plates.

5.3 Conclusion

The experiment results show that the system is able to recognize the car plate number. So, we can conclude that the system is can also implemented using optical character recognition method. This method is easier and no need too much data compare to the neural network. However this system also has weaknesses and should be improved to get better result and more accurate. There are a number of techniques which can be used for car license plate characters recognition. This system include BAM (Bi-directional Associative Memories), Neural Networks, image processing optical character recognition, pattern matching method and etc. So this system can be improved by using any techniques that can give the best result.

REFERENCES

[1] Digital Image Processing, Second Edition, Rafael C. Gonzalez, Richard E. Woods

[2] G.W.Awcock & R. Thomas (1996) Applied Image Processing, McGraw Hill, Inc

[3] Michael Negnevitsky ,(2002).A Guide To Intelligent System, Second Edition Pearson Education Limited.

[4] 20th May 2008, Internet Sources , Electro-Optical Technologies, Inc.
URL: http://users.erols.com/lnelson/lpir.html

[5] 20th May 2008, Internet Sources,
URL: http://en.wikipedia.org/wiki/Automatic_number_plate_recognition

[6] 25th May 2008, Internet Sources
URL: http://en.wikipedia.org/wiki/MATLAB

[7] 2nd June 2008, Internet Sources
URL: http://www.math.utah.edu/lab/ms/matlab/matlab.html

[8] Digital Image Processing using MATLAB, Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins

[9] 15th June 2008, Internet Sources
URL:http://www.computerworld.com/softwaretopics/software/apps/story/0,10801,73023,00.html

[10] 1st July 2008, Internet Source , Christos Stergiou and Dimitrios Siganos
URL: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

[11] N.Vazquez / M.Nakano / H.Pearez Meana, Journal of Applied Research & Technology ,Universidad Nacional Autonoma de Mexico, volume 1 (April 2003),

[12] 10th July 2008, Internet Sources, © 1994-2008 The MathWorks, Inc.
URL: http://www.mathworks.com/products/neuralnet/

[13] 10th August 2008, Internet Sources, Lee J. Nelson , leading consultants in license-plate recognition URL: http://www.photocop.com/recognition.htm

[14] HSien-ChuWu,Chwei-Shyong TSAI and Ching-Hao LAI,
A LICENSE PLATE RECOGNITION SYSTEM IN E-GOVERNMENT

[15] 20th September ,Internet Sources, Quan Quach
URL: http://www.blinkdagger.com/matlab/matlab-gui-graphical-user-interface-tutorial-for-beginners

[16] 22nd September 2008, Internet Sources, Eric Spellmann and XoXus LAST UPDATED: 01 May 2003
URL:http://searchciomidmarket.techtarget.com/sDefinition/0,,sid183_gci214132,00.html

[17] 5th October 2008, Internet Sources
URL: http://www.answers.com/topic/optical-character-recognition

[18] 5th October 2008, Internet Sources
URL: http://en.wikipedia.org/wiki/Optical_character_recognition

APPENDIX A

GANT CHART PSM 1

ACTIONS	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18
Find the supervisor						M											S	F
Choose title						I											T	I
Title confirmation						D											U	N
Title Registration																	D	A
Draft proposal and																	Y	L
slide presentation preparation						S												
Submit proposal & slide						E											W	E
Seminar 1						M											E	X
Find hardware specification																	E	A
Do research and find appropriate source																	K	M
Submit final report and log book																		
Start Develop the system																		

APPENDIX B
GANT CHART PSM 11

ACTIONS	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18
						M											S	F
First Thesis draft						I											T	I
Second Thesis draft						D											U	N
Develop the system																	D	A
Slide preparation																	Y	L
Seminar 11						S												
Final Report Draft & log book						E												
Final Thesis						M												

APPENDIX C
CAR PLATE RECOGNITION (CLPR) SYSTEM CODING

```
warning off
clc, close all, clear all
gmb=videoinput('winvideo');
pic=getsnapshot(gmb);
image(pic);
I=rgb2gray(pic);
figure,imshow(I);
imtool(I)
I1=imcrop(I,[31 97 275 63]);
imshow(I1);
R=imresize(I1,4);
figure,imshow(R);
se=strel('disk',1);
o=imopen(R,se);
figure,imshow(o);
te=strel('disk',5);
p=imerode(o,te);
figure,imshow(p);
level=graythresh(p);
BW=im2bw(p,level);
figure,imshow(BW);
imagen=imcomplement(BW);
imshow(imagen);title('INPUT IMAGE WITH NOISE')
%*-*-*Filter Image Noise*-*-*-*
if length(size(imagen))==3 %RGB image
    imagen=rgb2gray(imagen);
```

```
end
imagen = medfilt2(imagen);
[f c]=size(imagen);
imagen (1,1)=255;
imagen (f,1)=255;
imagen (1,c)=255;
imagen (f,c)=255;
%*-*-*END Filter Image Noise*-*-*-*
word=[];%Storage matrix word from image
re=imagen;
fid = fopen('text.txt', 'wt');%Opens text.txt as file for write
while 1
    [fl re]=lines(re);%Fcn 'lines' separate lines in text
    imgn=~fl;
    %*-*Uncomment line below to see lines one by one*-*-*-*
    %    imshow(fl);pause(1)
    %*-*--*-*-*-*-*-*-
    %*-*-*-*-*-Calculating connected components*-*-*-*-*-

%     L = bwlabel(imgn);
    mx=max(max(L));
    BW = edge(double(imgn),'sobel');
    [imx,imy]=size(BW);
    for n=1:mx
        [r,c] = find(L==n);
        rc = [r c];
        [sx sy]=size(rc);
        n1=zeros(imx,imy);
        for i=1:sx
            x1=rc(i,1);
            y1=rc(i,2);
            n1(x1,y1)=255;
        end
        %*-*-*-*-*-END Calculating connected components*-*-*-*-*
        n1=~n1;
        n1=~clip(n1);
        img_r=same_dim(n1);%Transf. to size 42 X 24
        %*-*Uncomment line below to see letters one by one*-*-*-*
        %        imshow(img_r);pause(1)
        %*-*-*-*-*-*-*-*
        letter=read_letter(img_r);%img to text
        word=[word letter];
    end
    %fprintf(fid,'%s\n',lower(word));%Write 'word' in text file (lower)
    fprintf(fid,'%s\n',word);%Write 'word' in text file (upper)
    word=[];%Clear 'word' variable
    %*-*-*When the sentences finish, breaks the loop*-*-*-*
    if isempty(re)  %See variable 're' in Fcn 'lines'
        break
    end
    %*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
end
fclose(fid);
winopen('text.txt')%Open 'text.txt' file
```

APPENDIX D
CAR PLATE RECOGNITION (CLPR) SYSTEM WITH GUI CODING

```
function varargout = gui1(varargin)
% GUI1 M-file for gui1.fig
%      GUI1, by itself, creates a new GUI1 or raises the existing
%      singleton*.
%
%      H = GUI1 returns the handle to a new GUI1 or the handle to
%      the existing singleton*.
%
%      GUI1('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI1.M with the given input
arguments.
%
%      GUI1('Property','Value',...) creates a new GUI1 or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before gui1_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to gui1_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```
% Edit the above text to modify the response to help gui1

% Last Modified by GUIDE v2.5 12-Oct-2008 13:27:16

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @gui1_OpeningFcn, ...
                   'gui_OutputFcn',  @gui1_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
  gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before gui1 is made visible.
function gui1_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui1 (see VARARGIN)

% Choose default command line output for gui1
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui1 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
%load the background image into Matlab
%if image is not in the same directory as the GUI files, you must use
the
%full path name of the iamge file
backgroundImage = importdata('background.jpg');
%select the axes
axes(handles.axes1);
%place image onto the axes
image(backgroundImage);
%remove the axis tick marks
axis off
axes(handles.axes3);
imshow('');
axis off
```

```
% --- Outputs from this function are returned to the command line.
function varargout = gui1_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%OCR (Optical Character Recognition).

%**********************************************************
warning off
%clc, close all, clear all   %clear previous command
gmb=videoinput('winvideo'); %get image from webcam
pic=getsnapshot(gmb);       %snap the image
image(pic);                 %show the image captured
I=rgb2gray(pic);            %convert the RGB image to grayscale
figure,imshow(I);           %show the image
imtool(I)
I1=imcrop(I,[35 111 227 51]);  %crop the image with the specific
coordinate
imshow(I1);                    %show the image crop
R=imresize(I1,4);
figure,imshow(R);
se=strel('disk',1);       %define the structuring element
o=imopen(R,se);           %opening method
figure,imshow(o);
te=strel('disk',5);
p=imerode(o,te);
figure,imshow(p);
level=graythresh(p);   %threshold the image
BW=im2bw(p,level);     %convert from grayscale to binary
figure,imshow(BW);
imagen=imcomplement(BW);   %complement the image
imshow(imagen);title('INPUT IMAGE WITH NOISE')
%*-*-*Filter Image Noise*-*-*-*
if length(size(imagen))==3 %RGB image
    imagen=rgb2gray(imagen);
end
imagen = medfilt2(imagen);
[f c]=size(imagen);
imagen (1,1)=255;
imagen (f,1)=255;
imagen (1,c)=255;
imagen (f,c)=255;
%*-*-*END Filter Image Noise*-*-*-*
word=[];%Storage matrix word from image
```

```
re=imagen;
fid = fopen('text.txt', 'wt');%Opens text.txt as file for write
while 1

    [fl re]=lines(re);%Fcn 'lines' separate lines in text
    imgn=~fl;
    %*-*Uncomment line below to see lines one by one*-*-*-*
    %    imshow(fl);pause(1)
    %*-*--*-*-*-*-*-*-
    %*-*-*-*-*-Calculating connected components*-*-*-*-*-

    L = bwlabel(imgn);
    mx=max(max(L));
    BW = edge(double(imgn),'sobel');
    [imx,imy]=size(BW);
    for n=1:mx
        [r,c] = find(L==n);
        rc = [r c];
        [sx sy]=size(rc);
        n1=zeros(imx,imy);
        for i=1:sx
            x1=rc(i,1);
            y1=rc(i,2);
            n1(x1,y1)=255;
        end
        %*-*-*-*-*-END Calculating connected components*-*-*-*-*
        n1=~n1;
        n1=~clip(n1);
        img_r=same_dim(n1);%Transf. to size 42 X 24
        %*-*Uncomment line below to see letters one by one*-*-*-*
        %        imshow(img_r);pause(1)
        %*-*-*-*-*-*-*-*
        letter=read_letter(img_r);%img to text
        word=[word letter];
    end
    %fprintf(fid,'%s\n',lower(word));%Write 'word' in text file (lower)
    fprintf(fid,'%s\n',word);%Write 'word' in text file (upper)
    set(handles.edit1,'string',word);
    set(handles.edit2,'string',('CLEAR'));
    guidata(hobjet,handles);
    fprintf(fid,'%s\n',word)
    %word=[];%Clear 'word' variable

 if (word==WEE2883)
    h=msgbox('BLACKLIST');
elseif(word==UMP2008)
        h=msgbox('CLEAR');
elseif(word==NAS5479)
       h=msgbox('CLEAR');
    else
        h=msgox('UNRECOGNIZED NUMBER');
    end;

    %*-*-*When the sentences finish, breaks the loop*-*-*-*
    if isempty(re)  %See variable 're' in Fcn 'lines'
        break
```

```
    end
    %*-*-*-*-*--*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
end
fclose(fid);
winopen('text.txt')%Open 'text.txt' file


%set(handles.edit1,'string',text);
%set(handles.edit2,'string',('CLEAR'));
%guidata(hobjet,handles);




function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as
a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as
a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes during object creation, after setting all properties.
function axes2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes2



% --- Executes on mouse press over axes background.
%function axes2_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% --- Executes during object creation, after setting all properties.
function axes3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes3



% --- Executes during object creation, after setting all properties.
function axes4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes4
```

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)




% --- Executes during object deletion, before destroying properties.
function pushbutton2_DeleteFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)




% --- Executes during object creation, after setting all properties.
function axes5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: place code in OpeningFcn to populate axes5




% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%exitbtn=questdlg('Exit Now?','Exit Program','Yes','No','No');
%switch exitbtn
%    case 'Yes'
        %delete(handles.figure1)
    %case 'No'
        %return
%end


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
exitbtn=questdlg('Exit Now?','Exit Program','Yes','No','No');
```

```
switch exitbtn
    case 'Yes'
        delete(handles.figure1)
    case 'No'
        return
end
```