

CURRENCY RECOGNITION AND CONVERTER SYSTEM

NURLAILA BINTI HAMAN

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS[♦]

JUDUL: **CURRENCY RECOGNITION AND CONVERTER SYSTEM**

SESI PENGAJIAN: 2008/2009

Saya NURLAILA BINTI HAMAN (860110 – 13 – 5052)

(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. ****Sila tandakan (√)**

SULIT

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

TERHAD

(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

TIDAK TERHAD

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**NO 49, TAMAN ANGSANA,
JALAN TUN HUSSEIN ONN,
97000, BINTULU,
SARAWAK.**

REZA EZUAN BIN SAMIN
(Nama Penyelia)

Tarikh: **28 OKTOBER 2008**

Tarikh: : **28 OKTOBER 2008**

- CATATAN:
- * Potong yang tidak berkenaan.
 - ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
 - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“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 : 28 OCTOBER 2008

CURRENCY RECOGNITION AND CONVERTER SYSTEM

NURLAILA BINTI HAMAN

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 : NURLAILA BINTI HAMAN

Date : 28 OCTOBER 2008

Specially Dedicated

To

My Beloved Mum and Dad & All My Family

My friends

Thanks for Everything

ACKNOWLEDGEMENT

Assalamualaikum w.b.t.. In the name of Allah, the Most Gracious, the Ever Merciful. Praise is to Allah, ord of the Universe and Peace and Prayers be upon His Final Prophet and Messenger Muhammad s.a.w.

I am greatly indebted to my supervisor, Mr. Reza Ezuan Bin Samin for his advice, ideas, guidance throughout my project. Thanks a lot to him.

I also would like to express my gratitude to my parents and all my family for the moral support and motivation from the beginning until now. Without them I would never come this far. Thank you very much.

Special thanks to FKEE staffs for helping me to complete my project. Suggestion and criticisms from my friends have always been helpful in finding solutions to my problems. Thank you all.

Finally, I would like to express my thanks to those who involves directly or indirectly in completion of my project.

ABSTRACT

The ease with we recognize a face, understand spoken words, read handwritten characters, identify our car keys in our pocket by feel, and decide whether an apple is ripe by its smell belies the astoundingly complex processes that underlie these acts of pattern recognition. As for currency recognition and converter system is an image processing technology that is used to identify currency amount and convert it into the other currencies as the users need. The purpose of currency recognition and converters is accurately to recognize the currencies and convert the currency immediately into the other currency. It can help human in order to live a better life. The system based on the computer communicates with web cam, catches video frames which include a visible image of currency amount and processes them. Various methodologies are used on the surface of the image. The selected area of the image is processed and analyzed with their parameters. Once the image of the currency amount was detected, its digit is recognized it will display on the user interface. This program will be developed using MATLAB. MATLAB is a high – performance language for technical computing.

ABSTRAK

Dalam menjalani kehidupan seharian, kita sangat mudah untuk mengenal wajah manusia, memahami segala perbualan sesama kita, mengenali tulisan yang tertera, mengenal kunci kereta, sehinggalah boleh mengetahui sama ada buah itu masak atau tidak. Ini semua adalah salah satu proses mengenali corak atau rentak suasana yang di sekeliling kita. Begitu juga dengan sistem mengenali matawang dan menukarkannya ke matawang asing yang merupakan salah satu aplikasi pemprosesan imej yang dibina untuk mengenal nilai matawang serta menukarkannya kepada matawang asing. Kegunaan program ini adalah untuk memudahkan kerja – kerja manusia dalam proses pengecaman nilai matawang serta menukarkannya kepada matawang lain dengan lebih efektif. Ini boleh membantu manusia untuk menjalankan kerja – kerja seharian dengan lebih cepat dan lancar. Sistem ini memproses imej duit yang diterima daripada kamera dan mengekstrak data pada imej tersebut. Seterusnya, imej ini akan terus diproses dan dibezakan antara satu sama lain melalui parameter yang telah didapati bagi setiap imej. Apabila imej itu telah dikenalpasti, nilai matawang tersebut akan dipaparkan pada skrin pengantara dengan pengguna (GUI) dan seterusnya pengguna boleh menukarkan nilai wang tersebut kepada matawang asing. Pengguna boleh memilih matawang asing mengikut senarai matawang yang tertera pada GUI tersebut. Program ini dihasilkan menggunakan perisian MATLAB.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF FIGURES	x
	LIST OF TABLES	xii
1	INTRODUCTION	1
	1.1 Background	1
	1.2 Image Processing	1
	1.3 Artificial Neural Network System	2
	1.4 Problem Statement	2
	1.5 Project Objectives	3
	1.6 Project Scopes	3
	1.7 Structure of Proposed System	4
	1.8 Thesis Organizations	4

2	LITERATURE REVIEW	6
	2.1 Introduction	6
	2.2 Neural Network System	6
	2.3 Images Representation	8
3	METHODOLOGY	11
	3.1 Introduction	11
	3.2 Image Processing Phase	11
	3.3 Neural Network Phase	13
	3.4 Development of Converter System	14
	3.5 Display The Amount of The Currency	14
	3.6 Block Diagram of The Currency Recognition And Converter System	15
	3.7 Flowchart of The Currency Recognition And Converter System	16
	3.8 Software and Hardware	17
4	RESULT AND DISCUSSION	18
	4.1 Introduction	18
	4.2 Result of The Captured Image	18
	4.3 Results From Image Processing Phase	21
	4.4 Parameters	28
	4.5 Discussion	34

5	CONCLUSION AND RECOMMENDATION	36
	5.1 Introduction	36
	5.2 Conclusion	36
	5.3 Costing and Commercialization	37
	5.4 Recommendation For Further Works	37
	REFERENCES	39 – 40
	APPENDICES A - B	41 – 49

LIST OF FIGURES

FIGURE	TITLE	PAGE
3.1	Image Acquisition Setup	12
3.2	Currency Recognition and Converter system Model	12
3.3	Design of Graphical User Interfaced	13
4.1	<i>“imaqhwinfo”</i> function is used to get info for webcam	20
4.2	Function that is used to captured image using webcam	20
4.3	Function to convert RGB image into grayscale image	21
4.4	RGB image captured using webcam	22
4.5	Image obtained after <i>“rgb2gray”</i> function	22
4.6	Image tool function to find coordinate of interest region	23
4.7	Function to crop the image after the coordinate have been specified	23
4.8	Image obtained after the cropping function execute	24
4.9	Function for Opening Method and erosion of the image	25
4.10	Image obtained after the Opening Method function executed	25
4.11	Image obtained after the erosion function executed	26
4.12	Function for <i>“thresholding”</i>	26
4.13	Image obtained after the <i>“thresholding”</i> function execute	27
4.14	Function for imcomplement method	27

4.15	Image obtained after the imcomplement function execute	28
4.16	GUI view with the amount display and converted amount	33

LIST OF TABLES

TABLE	TITLE	PAGE
3.1	Software Items	17
3.2	Hardware Items	17
4.1	Image captured using webcam	19
4.2	Results of Boundary Parameters for RM1 and RM5	29
4.3	Results of Boundary Parameters for RM10 and RM50	30 – 31
4.4	Results of Area Parameters for RM1, RM5, RM10 and RM50	32
5.1	Overall cost involve in the project	37

CHAPTER 1

INTRODUCTION

1.1 Background

This chapter explains the background of implementation between hardware, image processing, and neural network, for each part of the currency recognition system.

This thesis presents currency recognition as an application of computer vision. Computer vision is a process of using a computer to extract high level information from a digital image.

1.2 Image Processing

Image processing is any form of signal processing for which the input is an image, such as photographs or frames of video. The output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

Image processing usually refers to digital image processing, but optical and analog image processing are also possible.

1.3 Artificial Neural Network System

An artificial neural network (ANN), also called a simulated neural network (SNN) or commonly just neural network (NN) is an interconnected group of artificial neurons that uses a mathematical or computational model for information processing based on a connectionist 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.

1.4 Problem Statement

Currency Recognition and converter system is implemented to reduce human power to automatically recognize the amount of currency and convert it into the other currency without human supervision. Previously, human is needed to recognize the amount of the currency and convert it manually. So this project is developed to replace human power to recognize the amount of the currency.

1.5 Project Objectives

The objectives to develop this system are:

- Develop the program that can determine the amount of paper currency using neural network
- The system able to convert the currency from Ringgit Malaysia into other currencies after the amount has been determined

1.6 Project Scopes

The scope of this project is to develop a currency recognition and converter system by using image processing and neural network. In order to implement this system we have to use MATLAB Toolbox to achieve the objectives of the project. Thus, the focuses of this project are:

- i. To implement the system in order to recognize the currency amount.
- ii. To integrate hardware and software.
- iii. Extract the data from the currency image by using digital image processing toolbox.
- iv. Recognize the image of the currency amount by using neural network technique, using a feed-forward network with 3 layers

1.7 Structure of Proposed System

To achieve all the project objectives, the system must have a few stages. The stages are:

- Image Acquisition

The money tested in this project was Ringgit Malaysia 1, 5, 10 and 50. This amount of currency is used because it is easy to get a sample. The image was captured using a PC camera with 1.3 megapixels (Logitech QuickCam E3500 Plus).

- Data Extraction

Currency extraction is the key in a currency recognition system, which influences the accuracy of the system significantly. This is a very critical stage.

- Currency Segmentation

Currency segmentation takes the region of interest and attempts to divide it into individual characters.

- Currency recognition

Currency recognition is the last phase in currency recognition system. This stage will involve neural network system to recognize the isolated characters.

1.8 Thesis Organizations

Chapter 1: Explains the background of the image processing, neural network, objectives of the project and structures of the project.

Chapter 2: Focus on the previous projects contributions to this project. A brief literature review of the techniques used in Currency Recognition System

Chapter 3: Explains and discuss details about image processing process and neural network system. It discusses the brief review of how the morphology works in this project.

Chapter 4: Discuss on experimental analysis and results obtained at different stages under different conditions are discussed.

Chapter 5: Discuss on the conclusion for this project. This chapter also discuss on the problem encounter and recommendation for this project for future development or modification.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter will discuss about the previous project that have been done by others. Beside that, it gives basic knowledge about image processing and neural network.

2.2 Neural Network System

Up until now, we have developed banking machine for various kinds of paper currency using neural networks (NNs). In this paper, we report an enhanced neuro – recognition system to increase the more number of recognition patterns using axis – symmetrical mask and two image sensors. One sensor purpose is discrimination for a known image and another one is exclusion for an unknown image. Concretely, we implement the proposed method to an experimental system, which has two sensors. In addition, they are arranged on the up side and down side of the aisle, respectively.

Finally, we apply this proposed method to Euro currency, which will be delivered in 2002 year, using its dummy. The effectiveness of the proposed method is shown, numerically [1].

Nowadays, neural networks (NNs) are widely used in many fields of engineering and the most famous application is pattern recognition. In our previous researches, a banknote recognition system using a NN has been developed for various applications in worldwide banking systems such as banknote readers and sorters. In this paper, a new kind of banknotes, Thai banknotes, are being proposed as the object of recognition. First, the slab values, which are digitized characteristics of banknote by the mask set, are extracted from each banknote image. These slab values are summation of non – masked pixel values of each banknote. Second, slab values are inputted to the NN to execute its learning and recognition process. Third, for commercial usability, the NN algorithm is implemented on the DSP unit in order to execute the continuous learning and recognition. We show the recognition ability of the proposed system and its possibility for self – refreshed function on the DSP unit using Thai banknotes [2].

Standard back – propagation is a gradient descent algorithm. It was created using the Window – Hoof learning rule. The rule is the network weights and biases are updated or generated in the direction of the negative gradient of the performance function. The performance function is measured by MSE (mean square error)---the average squared error between the network outputs and the target outputs. This network can be used to approximate a general function. It can approximate equally well function with a finite number of discontinuous [3].

2.3 Images Representation

A digital image is a representation of a two-dimensional image using ones and zeros (binary). Depending on whether or not the image resolution is fixed, it may be of vector or raster type. Without qualifications, the term "digital image" usually refers to raster images[4].

Raster images can be created by a variety of input devices and techniques, such as digital cameras, scanners, coordinate-measuring machines, seismographic profiling, airborne radar, and more. They can also be synthesized from arbitrary non-image data, such as mathematical functions or three-dimensional geometric models; the latter being a major sub-area of computer graphics. The field of digital image processing is the study of algorithms for their transformation[4].

An image is stored as a matrix using standard MATLAB matrix conversions. There are five types of images supported by MATLAB:

- Intensity images
- RGB images
- Binary images
- Indexed images

2.3.1 Intensity Images

Intensity images are the simplest format for grayscale images. An intensity image is a data matrix, whose values represent intensities within some range[5].

2.3.2 RGB Images

The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue[6].

RGB is a device-dependent color space: different devices detect or reproduce a given RGB value differently, since the color elements (such as phosphors or dyes) and their response to the individual R, G, and B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same *color* across devices without some kind of color management[6].

The main purpose of the RGB color model is for the sensing, representation, and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors[6].

2.3.3 Binary Images

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color[7].

Binary images are also called bi-level or two-level. (The names *black-and-white*, *B&W*, monochrome or monochromatic are often used for this concept, but may also designate any images that have only one sample per pixel, such as grayscale images.) In Photoshop parlance, a binary image is the same as an image in "Bitmap" mode[7].

Binary images often arise in digital image processing as masks or as the result of certain operations such as segmentation, thresholding, and dithering. Some input/output devices, such as laser printers, fax machines, and bi-level computer displays, can only handle bi-level images[7].

2.3.4 Indexed Images

In computing, indexed color is a technique to manage digital images' colors in a limited fashion, in order to save computer's memory and file storage, while speeding up display refresh and telecom transfers. When an image is encoded this way, the color information is not directly carried by the image pixel data, but it is stored into a separate piece of data called a palette: an array of color elements, in which every element, a color, is indexed by its position within the array. This way, each pixel does not contain the full information to represent its color, but only its index into the palette. This technique is sometimes referred as pseudocolor or indirect color, as colors are addressed indirectly[8].

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the process of recognition phase will be discussed. It will cover about software process, analysis, design and development of the project. It also will explain about the software and hardware that will be used in this project. There are two phases in this project which are image processing phase and neural network phase.

3.2 Image Processing Phase

For the image processing program the system required various methodologies to process the desired image. The methodology includes convert the image, crop the region of interest (ROI), dilation method, erosion method, threshold, complement the image and applied some parameters to calculate the value of area boundaries to distinguish each character of number in the image.

3.2.1 Image Acquisition

The money tested in this project was Ringgit Malaysia 1, 5, 10 and 50. This amount of currency is used because it is easy to get a sample. The image was captured using a webcam with 1.3 megapixels (Logitech QuickCam E3500 Plus). The images were taken under natural daylight or with lighting from pendaflour lamp.

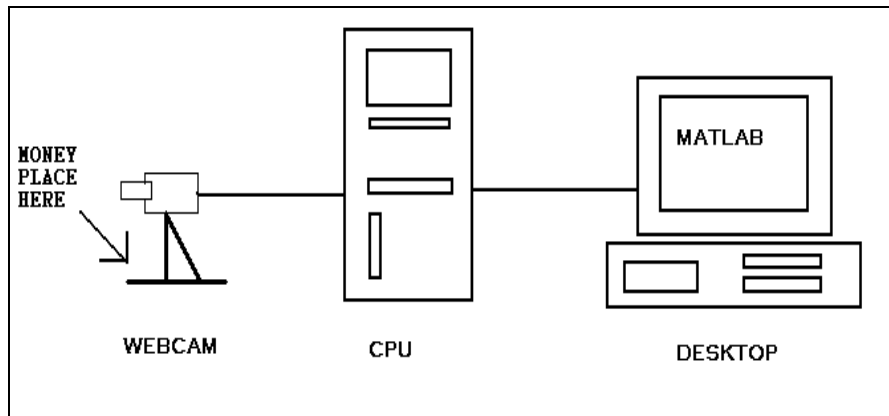


Figure 3.1: Image Acquisition Setup



Figure 3.2: Currency Recognition and Converter system Model

3.2.2 Captured Images

As the first step in the flow chart, the image of the money was captured by using webcam.

3.2.3 Developed Data Extraction Program

The extraction program is build by using MATLAB. The function and algorithms must be working sequence to get correct flow and results for the data extraction using image processing toolbox in MATLAB. This stage includes many steps and it is a key step in currency recognition system, which influences the accuracy of the system significantly. This is very critical stages and need a long time to go through.

3.3 Neural Network Phase

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.

3.4 Development of Converter System

The value of each currency is different from each other according to their parameters of boundaries and areas. This is easy to distinguish them from each other. The Ringgit Malaysia will convert into US Dollar, Baht and Rupiah.

The conversion rates are always changing time by time. The weakness for this project is the rates of the converter have to be updated daily by manually. For further improvement, this project has to implement with web base so that the rates of the conversion system are up to date.

3.5 Display The Amount Of The Currency

The recognition amount of the currency and the converted amount are displayed on the Graphical User Interface (GUI). GUI is type of user interface which allows people to interact with the software or device. This will helps users to easily used this program. It also provided user friendly environment.

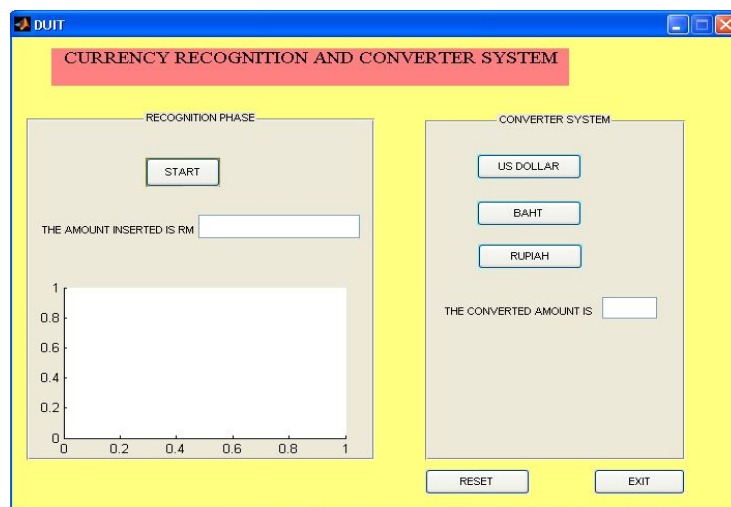
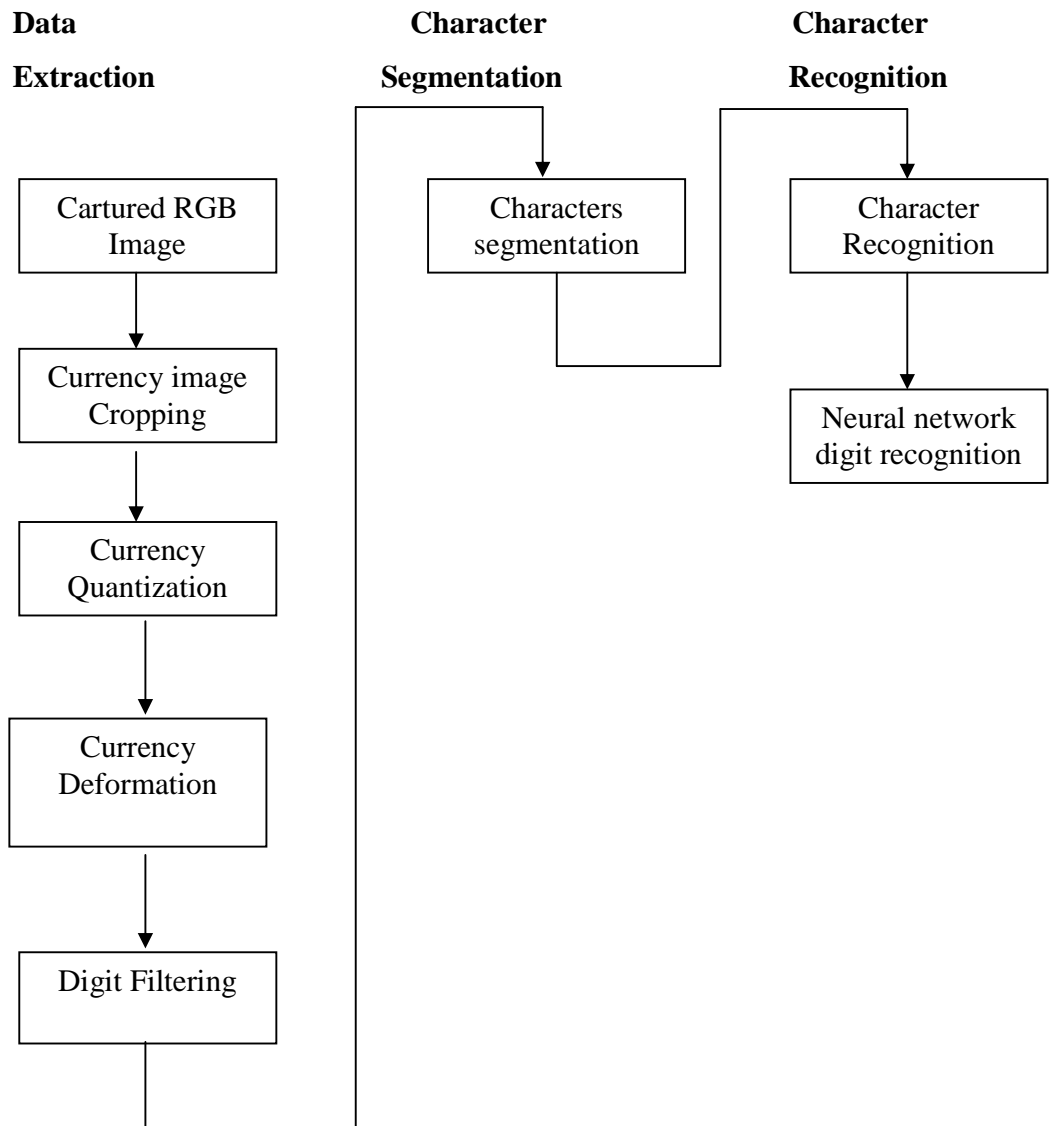
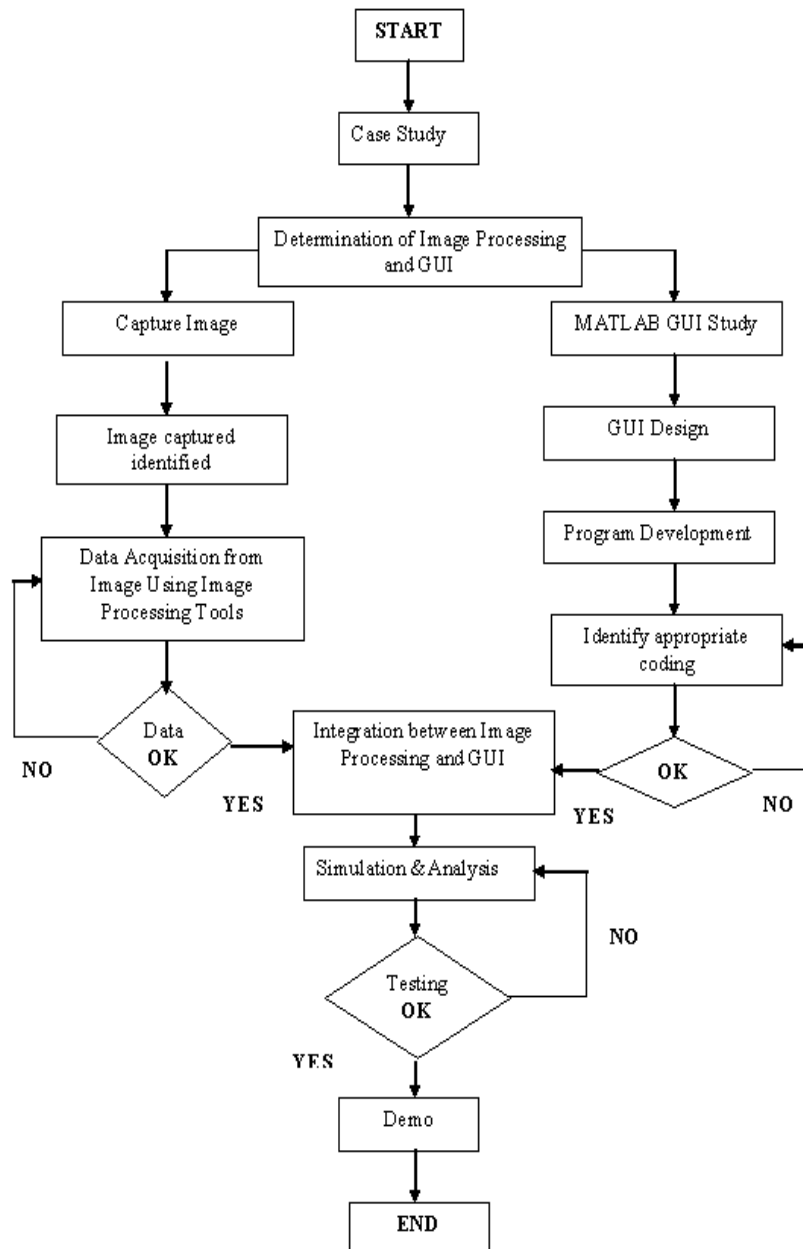


Figure 3.3: Design of Graphical User Interfaced

3.6 Block Diagram of the Currency Recognition and Converter System



3.7 Flowchart Of The Currency Recognition and Converter System



3.8 Software and Hardware

The software and hardware are the important part in this project. This is because the system is using the online forming to recognize and convert the value of the currency. The implementation of software and hardware are important for establishing and maintaining the resources necessary to perform the project successfully.

3.8.1 Software Items

Item Type	Description	Version
Operating system	Microsoft Windows XP	Professional Version 2002, Service Pack 2
Desktop	Microsoft Office 2003 MATLAB 7.1	Office Professional Version 7.1 product release 14

Table 3.1: Software Items

3.8.2 Hardware Items

Item Type	Minimum Requirement
Desktop	Intel®, Pentium® M 725, 1.6 GHz, 1.2 GB of RAM
Webcam	Frame rate: 640 x 480 pixels,

Table 3.2: Hardware Items

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter will discuss about the results obtained from the methodologies that have been proposed for this project.

4.2 Result of Captured Image

This subchapter will discuss about the image obtained for this project. The input for this project is from webcam. The images that have been captured will be processed using the methodologies that have been proposed earlier.

The *imaghwinfo* for the device is “*winvideo*”. The “*winvideo*” function is to connect the device through USB port with the program that has been developed. So, this will give an easier way to integrate the webcam with the MATLAB program.

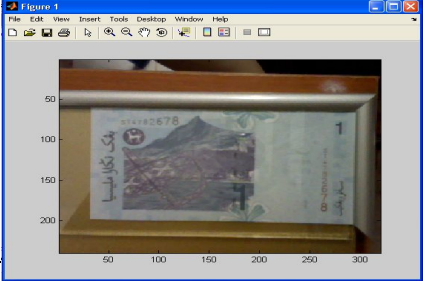
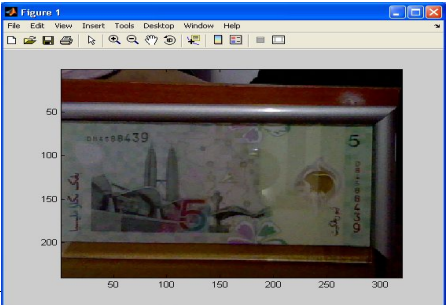
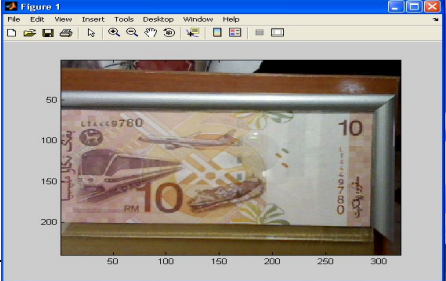
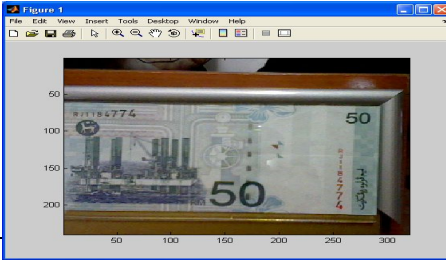
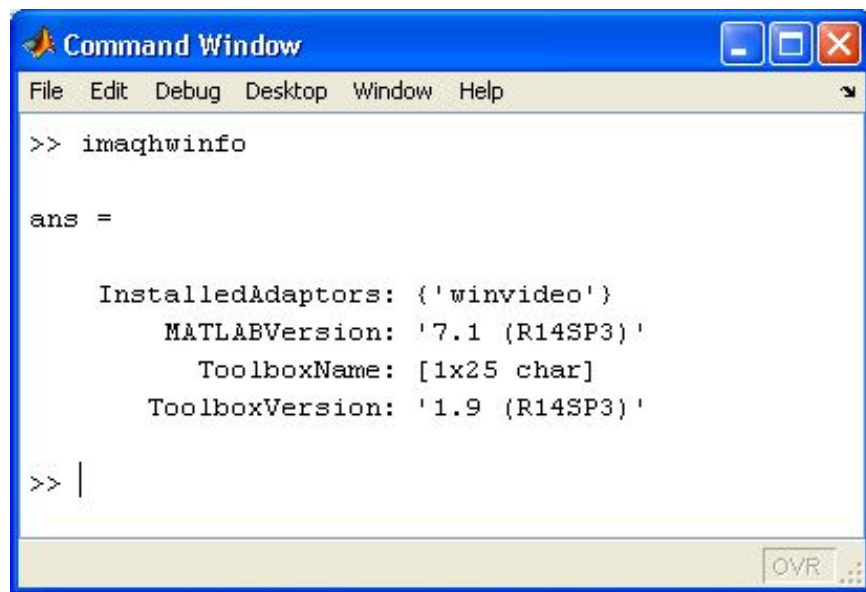
Currency Value	Image Captured
Ringgit Malaysia 1 (RM1)	
Ringgit Malaysia 5 (RM5)	
Ringgit Malaysia 10 (RM10)	
Ringgit Malaysia 50 (RM50)	

Table 4.1: Image captured using webcam



```
Command Window
File Edit Debug Desktop Window Help
>> imaqhwinfo

ans =

    InstalledAdaptors: ('winvideo')
      MATLABVersion: '7.1 (R14SP3)'
        ToolboxName: [1x25 char]
      ToolboxVersion: '1.9 (R14SP3)'

>> |
```

Figure 4.1: “*imaqhwinfo*” function is used to get info for webcam

```
clc;
%captured image
gmb=videoinput('winvideo');
preview(gmb);
pic=getsnapshot(gmb);
figure,imshow(pic);
```

Figure 4.2: Function that is used to captured image using webcam

4.3 Results From Image Processing Phase

This part will discuss the results obtained on image processing stages and all method that is used on this stage.

4.3.1 rgb2gray Function

This function is used to convert RGB image into a grayscale image. In this method, the RGB image will be converted into grayscale image by eliminating the hue and saturation information while retaining the luminance. “*rgb2gray*” function will help to reduce the noise of the image because in RGB image, the noise is difficult to determine. This is because RGB image is combined with various colors.

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

Figure 4.3: Function to convert RGB image into grayscale image

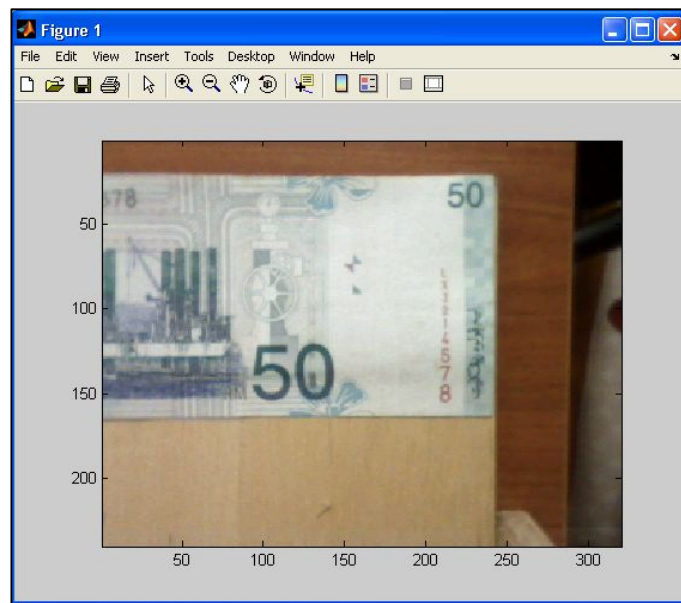


Figure 4.4: RGB image captured using webcam

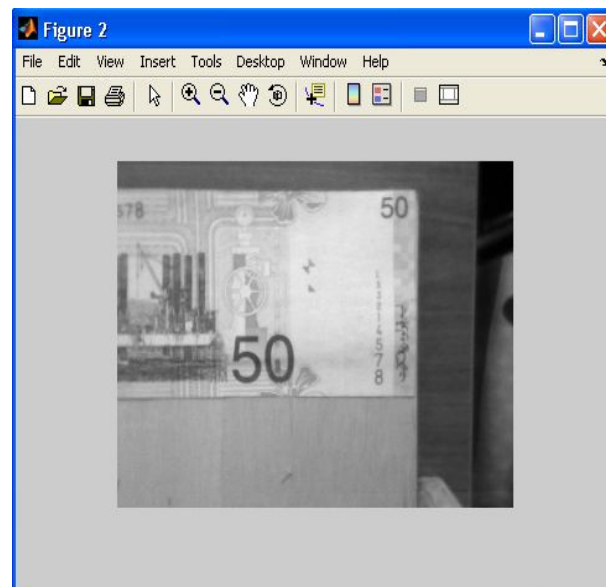


Figure 4.5: Image obtained after “*rgb2gray*” function

4.3.2 Cropping Function

For the crop function, there are two methods to crop the image. First methods are by manually crop and the second method by specifying the coordinate of the region of interest. For this project, the second method has been used to crop the region of the interest.

To find the coordinate that is going to specify, “*imtool*” function had been used. This function is one type of image tool function that used to find the specific coordinate.

```
%finding specific coordinate  
imtool (I);
```

Figure 4.6: Image tool function to find coordinate of interest region

```
%cropping region of interest  
I3=imcrop(I,[272 19 41 34]);  
figure, imshow(I3);
```

Figure 4.7: Function to crop the image after the coordinate have been specified



Figure 4.8: Image obtained after the cropping function execute

4.3.3 Opening Function

The Opening function will perform an erosion method and dilation method using a predefined neighborhood or structuring element. 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. This method is used to filter the noise of the image.

With erosion method, the image will be eroded and the image will become smaller. Then, the dilation method will be executed. The image will be dilated and become bigger.

```
%opening method
se=strel('disk',1); %dilation
o=imopen(R,se);
figure,imshow(o);
%erosion method
te=strel('disk',5);
p=imerode(o,te);
figure,imshow(p);
```

Figure 4.9: Function for Opening Method and erosion of the image

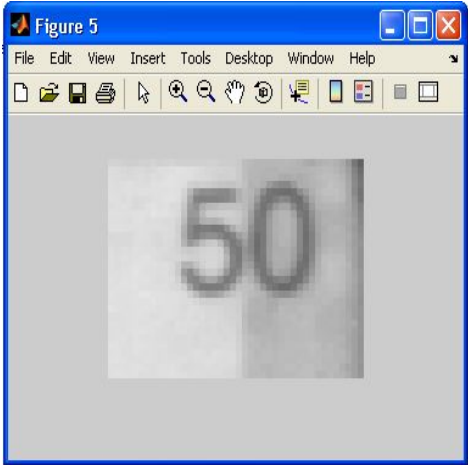


Figure 4.10: Image obtained after the Opening Method function executed

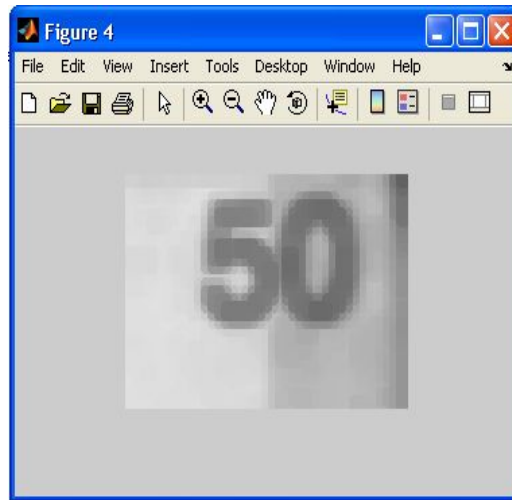


Figure 4.11: Image obtained after the erosion function executed

4.3.4 Thresholding Function

The “*thresholding*” function will automatically compute an appropriate threshold that is used to convert the grayscale image to binary image. This function uses Otsu’s Method, which chooses the threshold to minimize the intraclass variance of the black and white pixels.

Otsu’s method or Otsu’s Algorithm is usually used in computer vision and medical image processing to perform threshold. The purpose of this method is to separate foreground from background of the image.

```
%thresholding function  
level=graythresh(p);  
BW=im2bw(p, level);  
figure, imshow(BW);
```

Figure 4.12: Function for “*thresholding*”

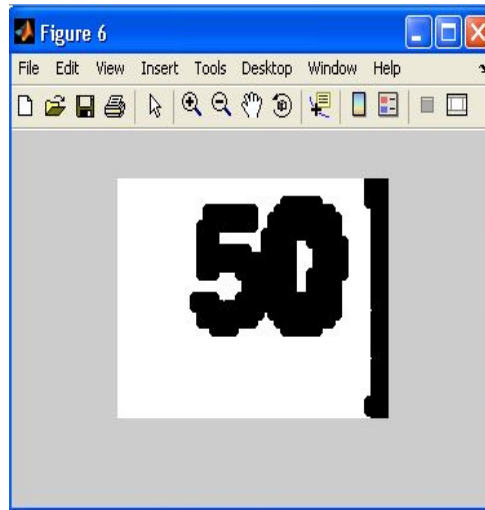


Figure 4.13: Image obtained after the “thresholding” function execute

4.3.5 Imcomplement function

This function will compliment the binary image where zeros become ones and ones become zeros while black and white are reversed. In the complement of an intensity or RGB image, each pixel value is subtracted from the maximum pixel value supported by the class (or 1.0 for double – precision images) and the difference is used as the pixel value in the output image. In the output image, dark areas becomes lighter and light areas become darker.

```
%compliment the image  
C=imcomplement(BW);  
figure,imshow(C);
```

Figure 4.14: Function for imcomplement method

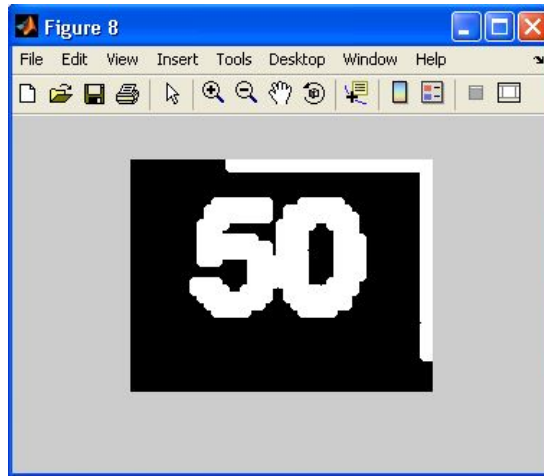


Figure 4.15: Image obtained after the imcomplement function execute

4.4 Parameters

This subchapter will discuss the final stages in image processing phase where it differentiates the image from each other. The “*boundaries*” and “*bwarea*” is used for every single amount of money to distinguish between them.

4.4.1 Boundaries Parameter

“*Boundaries*” method will calculate the boundary of the external side of the image. If the value is approximately to 1, the image is nearly equal to rounded image. The results are shown in Table 4.2 and Table 4.3.

	RM 1	RM 5
1.	0.40	0.22
2.	0.39	0.22
3.	0.40	0.21
4.	0.39	0.21
5.	0.39	0.20
6.	0.38	0.22
7.	0.40	0.22
8.	0.40	0.22
9.	0.39	0.21
10.	0.41	0.20
11.	0.40	0.22
12.	0.39	0.22
13.	0.37	0.22
14.	0.40	0.21
15.	0.39	0.23
16.	0.40	0.21
17.	0.41	0.22
18.	0.41	0.22
19.	0.39	0.22
20.	0.38	0.20
21.	0.40	0.23
22.	0.39	0.22
23.	0.38	0.21
24.	0.42	0.22
25.	0.40	0.21
26.	0.40	0.21
27.	0.39	0.22
28.	0.39	0.22
29.	0.38	0.21
30.	0.38	0.21
31.	0.41	0.22
32.	0.40	0.22
33.	0.39	0.22
34.	0.39	0.22
35.	0.40	0.21

Table 4.2: Results of Boundary Parameter for RM1 and RM5

NO.	RM 10		RM 50	
	1	0	5	0
1.	0.38	0.60	0.21	0.66
2.	0.37	0.65	0.22	0.64
3.	0.39	0.64	0.23	0.65
4.	0.37	0.61	0.22	0.66
5.	0.40	0.63	0.22	0.64
6.	0.39	0.65	0.22	0.65
7.	0.40	0.62	0.23	0.62
8.	0.41	0.60	0.21	0.66
9.	0.39	0.64	0.23	0.66
10.	0.37	0.64	0.23	0.65
11.	0.38	0.65	0.23	0.66
12.	0.40	0.66	0.22	0.64
13.	0.39	0.61	0.23	0.64
14.	0.38	0.62	0.21	0.66
15.	0.37	0.64	0.22	0.66
16.	0.39	0.63	0.23	0.66
17.	0.39	0.60	0.23	0.65
18.	0.40	0.61	0.22	0.66
19.	0.39	0.63	0.21	0.64
20.	0.38	0.64	0.22	0.64
21.	0.40	0.66	0.23	0.66
22.	0.41	0.65	0.23	0.66
23.	0.42	0.64	0.21	0.65
24.	0.37	0.63	0.22	0.66
25.	0.38	0.66	0.22	0.64
26.	0.40	0.65	0.23	0.66
27.	0.37	0.61	0.22	0.66
28.	0.40	0.60	0.23	0.65

29.	0.38	0.62	0.21	0.64
30.	0.39	0.65	0.23	0.66
31.	0.40	0.64	0.22	0.63
32.	0.42	0.63	0.22	0.66
33.	0.38	0.65	0.23	0.64
34.	0.39	0.66	0.22	0.63
35.	0.41	0.61	0.22	0.66

Table 4.3: Results of Boundary Parameter for RM10 and RM50

4.4.2 Bwarea Parameter

“*Bwarea*” method will compute the area of the objects in a binary image. This method will calculate the area for binary image only. That is why we have to complement the image after the “*thresholding*” function. The results of the area is shown in Table 4.4.

NO.	RM 1	RM 5	RM 10	RM 50
1.	3.9675×10^3	7.2143×10^3	1.7347×10^4	1.5765×10^4
2.	3.9260×10^3	7.2212×10^3	1.8120×10^4	1.5644×10^4
3.	4.1599×10^3	7.1234×10^3	1.7445×10^4	1.5543×10^4
4.	3.9543×10^3	6.9955×10^3	1.7897×10^4	1.5443×10^4
5.	4.1239×10^3	7.3686×10^3	1.7544×10^4	1.5234×10^4
6.	3.9945×10^3	7.2243×10^3	1.7989×10^4	1.5198×10^4
7.	3.9875×10^3	7.1523×10^3	1.7880×10^4	1.5297×10^4
8.	4.1955×10^3	7.0112×10^3	1.7549×10^4	1.5148×10^4
9.	3.9880×10^3	7.2234×10^3	1.7669×10^4	1.5346×10^4
10.	3.9610×10^3	7.2431×10^3	1.7789×10^4	1.5129×10^4
11.	4.1125×10^3	7.0231×10^3	1.7634×10^4	1.6102×10^4
12.	4.1340×10^3	7.1123×10^3	1.7569×10^4	1.6899×10^4
13.	3.9980×10^3	7.3223×10^3	1.7393×10^4	1.5243×10^4
14.	4.1220×10^3	7.4665×10^3	1.7612×10^4	1.6454×10^4
15.	4.0890×10^3	7.0342×10^3	1.7871×10^4	1.5678×10^4
16.	3.8680×10^3	7.4451×10^3	1.7654×10^4	1.6435×10^4
17.	3.8775×10^3	7.4066×10^3	1.7321×10^4	1.7110×10^4
18.	4.1110×10^3	7.5510×10^3	1.7456×10^4	1.5443×10^4
19.	4.1332×10^3	7.4432×10^3	1.7342×10^4	1.6721×10^4
20.	3.9787×10^3	7.0213×10^3	1.7231×10^4	1.5432×10^4
21.	3.8990×10^3	7.3380×10^3	1.7332×10^4	1.5679×10^4
22.	4.2112×10^3	7.3221×10^3	1.7651×10^4	1.5994×10^4
23.	4.1231×10^3	7.5510×10^3	1.7234×10^4	1.6478×10^4
24.	3.9771×10^3	7.4223×10^3	1.7451×10^4	1.6468×10^4
25.	3.8671×10^3	7.4845×10^3	1.7539×10^4	1.5498×10^4

Table 4.4: Results of Area Parameters for RM1, RM5, RM10 and RM50

4.4.3 Results On Graphical User Interfaced (GUI)

This parts will shown the results that are displayed on GUI. After the amount of the currency is shown, user can choose to convert the currency into the other currency as their want.

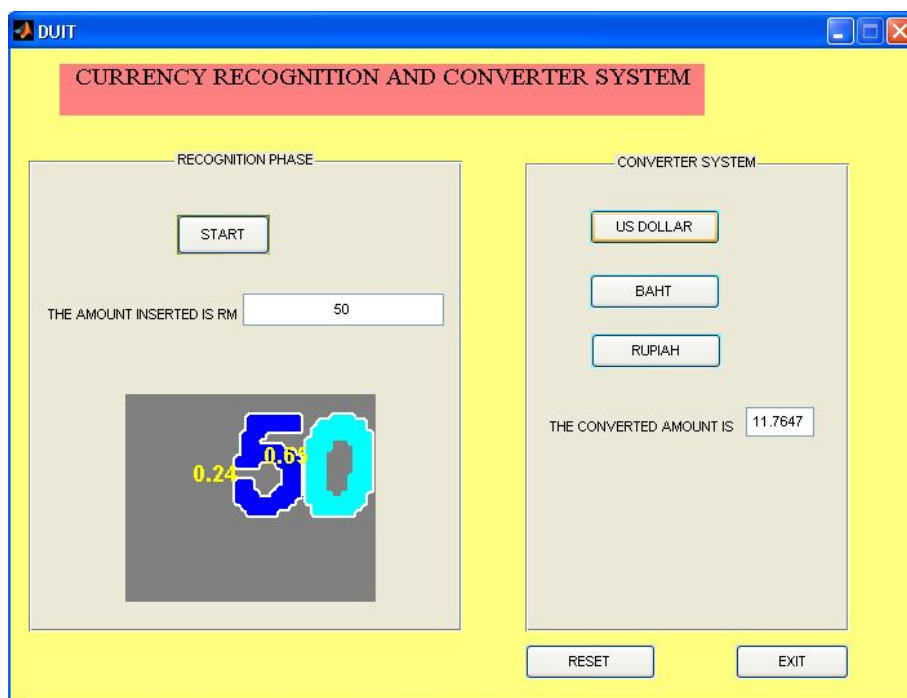


Figure 4.16: GUI view with the amount display and converted amount

4.5 Discussion

These subchapters will discuss about the results for this projects and the problems faces during the project development.

4.5.1 Image Processing Phase

The morphologies method that include erosion and dilation are very useful to make sure noise on the image are reduce before obtained the results which is the image of the amount on the surface of the currencies.

Thresholding function is required to convert the grayscale image into binary image because the last step required the binary image. The calculation for the area and the boundaries of the images is using binary image.

According to the results for boundaries and area, the images can be distinguish from each other by using if and else function.

4.5.2 Neural Network Phase

This stage had been eliminated because it's hard to find the input for neural network stages. Besides that, the scopes of this project are very big and need longer time to be completed. So, in order to complete this project in time, the neural network stages had been eliminated.

4.5.3 Problems Occur and Solutions

This project is totally implement using MATLAB software. So, it will take time to explore about MATLAB and Image Processing. In order to fully understand about the Image Processing, many tutorials have been done and studied the examples.

The scope of this project is very big where it includes image processing and neural network system. So, there are many problems occur such as time constrain and complexity of the neural networks system. To overcome this problem, neural network stages have been eliminates and all the data extraction and recognition process are using image processing.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Introduction

This chapter will discuss about the conclusion for the objectives of this projects and recommendation for further works of this projects.

5.2 Conclusion

From overall and based on the objectives of this project, it can be conclude that the objectives of this project are not well achieved because the main objectives are recognition using neural networks system. But the amounts of the currencies are well recognized using image processing and the converter system also convert the amount successfully.

There are many lessons learn such as time management, smoothness, of scheduling and well planned. It also experiences in exploring image processing toolbox.

Proper project management is the key to successful development of the system. Without proper planning and scheduling, the processes of software development become difficult to manage. The main constraints in this project are quality and time. The motivations also take part in personal matter.

5.3 Costing and Commercialization

This subchapter will discuss all the cost that involves in this project and commercialization value of this project.

No.	Items	No. of Items	Cost/RM
1.	Logitech QuickCam E3500 Plus	1	110.00
2.	Model (include wood, glass, plastic stand)	-	25.00
		Total:	135.00

Table 5.1: Overall cost involve in the project

For the commercialization value of the system, this system can be improved and developed for commercial used. It can reduce on human power especially in banking system where people will deal with many currencies.

5.4 Recommendation for Further Works

The converter system is not updated with daily value of conversion system. So, it is recommended to improve this project by link the system with the database or online converter system. This will help this system become more efficient and accurately converted.

Another weakness of this project is this system cannot distinguish between genuine and non – genuine currencies. So, it is highly recommended that this system is updated with device or program that can detect genuine and non – genuine currencies.

For further improvement, it is recommended to improve the methodology, technique and the algorithm to get more accurate results. With the more affective

method, this system will become more effective and the currencies can be distinguished between them more accurate.

As a conclusion, after completing this project, there are many experiences and gains knowledge in many areas besides it can make more understand about MATLAB and Image Processing although it cannot complete the task in neural network stages.

REFERENCES

- [1] Fumiaki Takeda, Toshihiro Nishikage, Multiple Kinds Of Paper Currency Recognition Using Neural Network and Application For Euro Currency, Department of Information Systems Engineering Kochi University of Technology 185, Miyanokuchi, Tosayamada-cho, Kami-gun, Kochi, 782-8502, Japan. Available at: <http://ieeexplore.ieee.org/iel5/6927/18622/00857888.pdf?arnumber=857888>
- [2] Fumiaki Takeda, Lalita Sakoobunthu, Hironobu Satou, Thai Banknote Recognition Using Neural Network and Continues Learning by DSP Unit, Department of Information System Engineering, Kochi University of Technology, 185 Miyanokuchi, Tosayamada-cho, Kami-gun, Kochi 782-8502 Japan. Available at: <http://spingerlink.com/content/dhdp16y88pnxh2w/>
- [3] Jinxing Han Gould (2004), FOREX Prediction Using an Artificial Intelligence System, Master Thesis, Oklahoma State University.
- [4] 10th February 2008, citing internet sources URL: http://en.wikipedia.org/wiki/Digital_image
- [5] Norliza Binti Wahab, Computer Vision Currency Calculator, Degree Thesis, Universiti Malaysia Pahang.
- [6] 10th February 2008, citing internet sources URL: <http://en.wikipedia.org/wiki/Rgb>

- [7] 10th February 2008, citing internet sources URL:
http://en.wikipedia.org/wiki/Binary_images
- [8] 10th February 2008, citing internet sources URL:
http://en.wikipedia.org/wiki/Indexed_color
- [9] Mohammed Jameel Ahmed, License Plate Recognition System, Master Thesis, King Fahd University Petroleum and Minerals.
- [10] Image Processing Toolbox 5 (User Guide), The Mathworks
- [11] Howard Demuth, Mark Beale, Martin Hagan, Neural Networks Toolbox 5 (User Guide), The Mathworks.
- [12] 18th August 2008 citing internet sources at URL
<http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do?objectType=category&objectId=138&objectName=Morphology%20and%20Segmentation>
- [13] 18th August 2008 citing internet sources at URL
<http://www.blinkdagger.com>
- [14] 18th August 2008 citing internet sources at URL
<http://www.mathworks.com/matlabcentral/fileexchange/loadCategory.do?objectType=category&objectId=26>
- [15] 19th September 2008 citing internet sources at URL
<http://updates.zdnet.com/tags/matlab.html>

APPENDIX A

PROGRAMMING OF CURRENCY RECOGNITION AND CONVERTER SYSTEM

```
clc;
%captured image
gmb=videoinput('winvideo');
preview(gmb);
pic=getsnapshot(gmb);
figure,imshow(pic);

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

%cropping region of interest
I3=imcrop(I,[272 19 41 34]);
figure,imshow(I3);

%resize the area that have been crop
R=imresize(I3,5);
figure,imshow(R);

%opening method
se=strel('disk',1);
o=imopen(R,se);
figure,imshow(o);

%erosion method
te=strel('disk',5);
p=imerode(o,te);
figure,imshow(p);

%thresholding function
level=graythresh(p);
BW=im2bw(p,level);
figure,imshow(BW);

%compliment the image
C=imcomplement(BW);
figure,imshow(C);
```



```

%calculate the area of the image
bwarea(C);

%calculate the boundaries of the image
[B,L]=bwboundaries(C,'noholes');
imshow (label2rgb(L,@jet, [.5 .5 .5]))
hold on

for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1),'w', 'LineWidth', 2)
end

stats = regionprops (L, 'Area', 'Centroid');
threshold = 0.94;

%loop over the boundaries
for k = 1:length(B)

    %obtain (X,Y) boundary coordinates corresponding to label 'k'
    boundary = B{k};

    %compute a simple estimate of the image's perimeter
    delta_sq = diff(boundary).^2;
    perimeter=sum(sqrt(sum(delta_sq,2)));

    %obtain the area calculation corresponding to label 'k'
    area = stats (k).Area;

    %compute the roundness metric
    metric = 4*pi*area/perimeter^2;

    %display the results
    metric_string=sprintf('%2.2f',metric);

    %mark the object above threshold with a black circle
    if metric > threshold
        centroid=stats(k).Centroid;
        plot(centroid(1),centroid(2),'ko');
    end

    text(boundary(1,2)-
35, boundary(1,1)+13, metric_string, 'Color', 'y', ...
        'fontsize', 14, 'Fontweight', 'bold');
end

satu = bwarea(C);
dua=metric;

if (satu <= 1.9000e+004)&&(satu>=1.3000e+004);
    if(dua <= 0.42)&&(dua >= 0.35);
        if (dua <= 0.67)&&(dua >= 0.59);
            end
        end
    end
end

```

```
        disp('THE MONEY INSERTED IS RM10')
    end
    if (satu <= 5.0000e+003)&&(satu>=3.0000e+003);
        if (dua <= 0.50)&&(dua >= 0.30);
            end
            disp('THE MONEY INSERTED IS RM5')
        end
    end

    if (satu <= 8.2000e+003)&&(satu >= 6.0000e+003);
        if (dua <= 0.24)&&(dua >= 0.20);
            end
            disp('THE MONEY INSERTED IS RM50')
        end
    end

    if (satu <= 1.8000e+004)&&(satu >= 1.0000e+004);
        if (dua <= 0.24)&&(dua >= 0.20);
            if (dua <= 0.67)&&(dua >= 0.60);
                end
            end
            disp('THE MONEY INSERTED IS RM1')
        end
    end

    else
        disp('NOT VALID.PLEASE TRY AGAIN')
    end
end
```

APPENDIX B

PROGRAMMING FOR CURRENCY RECOGNITION AND CONVERTER SYSTEM USING GUI

```
function varargout = DUIT(varargin)
% DUIT M-file for DUIT.fig
%     DUIT, by itself, creates a new DUIT or raises the existing
%     singleton*.
%
%     H = DUIT returns the handle to a new DUIT or the handle to
%     the existing singleton*.
%
%     DUIT('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in DUIT.M with the given input
arguments.
%
%     DUIT('Property','Value',...) creates a new DUIT or raises the
%     existing singleton*. Starting from the left, property value
pairs are
%     applied to the GUI before DUIT_OpeningFunction gets called. An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to DUIT_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 DUIT

% Last Modified by GUIDE v2.5 23-Sep-2008 12:03:57

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @DUIT_OpeningFcn, ...
                  'gui_OutputFcn',  @DUIT_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
```

```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

% --- Executes just before DUIT is made visible.
function DUIT_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 DUIT (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DUIT wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = DUIT_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)

```

```

clc;
gmb=videoinput('winvideo');
preview(gmb);
pic=getsnapshot(gmb);
%figure,imshow(pic);
I=rgb2gray(pic);
%figure,imshow(I);
I3=imcrop(I,[272 19 41 34]);
%figure,imshow(I3);
R=imresize(I3,5);
%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);
C=imcomplement(BW);
%figure,imshow(C);
bwarea(C);
[B,L]=bwboundaries(C,'noholes');
imshow(label2rgb(L,@jet,[.5 .5 .5]))
hold on
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1),'w', 'LineWidth', 2)
end
stats = regionprops(L,'Area','Centroid');
threshold = 0.94;
for k = 1:length(B)
    boundary = B{k};
    delta_sq = diff(boundary).^2;
    perimeter=sum(sqrt(sum(delta_sq,2)));
    area = stats(k).Area;
    metric = 4*pi*area/perimeter^2;
    metric_string=sprintf('%2.2f',metric);
    if metric > threshold
        centroid=stats(k).Centroid;
        plot(centroid(1),centroid(2),'ko');
    end
    text(boundary(1,2)-
35, boundary(1,1)+13, metric_string, 'Color', 'y', ...
        'fontsize', 14, 'Fontweight', 'bold');
end
satu = bwarea(C);
dua=metric;
if (satu <= 1.9000e+004)&&(satu>=1.3000e+004);
    if(dua <= 0.42)&&(dua >= 0.35);
        if (dua <= 0.67)&&(dua >= 0.59);
            end
        end
        set(handles.edit1, 'string', '10');
end
if (satu <= 5.0000e+003)&&(satu>=3.0000e+003);

```

```

        if (dua <= 0.50)&&(dua >= 0.30);
        end
        set(handles.edit1,'string','1');
    end
    if (satu <= 8.2000e+003)&&(satu >= 6.0000e+003);
        if (dua <= 0.24)&&(dua >= 0.20);
        end
        set(handles.edit1,'string','5');
    end
    if (satu <= 1.8000e+004)&&(satu >= 1.0000e+004);
        if (dua <= 0.24)&&(dua >= 0.20);
            if (dua <= 0.67)&&(dua >= 0.60);
            end
        end
        set(handles.edit1,'string','50');
    end

else
    set(handles.edit1,'string','NOT VALID.PLEASE TRY AGAIN');
end

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

% --- 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)

```

```

A=get(handles.edit1,'string');
rupiah=str2num(A)*3.25;
set(handles.edit2,'string',rupiah);
%guidata(hObject, handles);

% --- 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)
B=get(handles.edit1,'string');
dollar=str2num(B)/4.25;
set(handles.edit2,'string',dollar);
%guidata(hObject, handles);

% --- 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)
C=get(handles.edit1,'string');
baht=str2num(C)*5.25;
set(handles.edit2,'string',baht);
%guidata(hObject, handles);

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 on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.edit1,'string','');
set(handles.edit2,'string','');
%guidata(hObject, handles);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (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
```