

**INTERACTIVE HANDWRITING RECOGNITION FOR COURSEWORK  
DATA ENTRY**

By

LEE WENG SENG

16455

Final Report submitted in partial fulfilment of  
the requirements for the  
Degree Bachelor of Engineering (Hons)  
(Electrical & Electronics Engineering)

FYP II January 2016

Universiti Teknologi PETRONAS  
Bandar Seri Iskandar  
31750 Tronoh  
Perak Darul Ridzuan

MR PATRICK SEBASTIAN

# **CERTIFICATION OF APPROVAL**

## **INTERACTIVE HANDWRITING RECOGNITION FOR COURSEWORK DATA ENTRY**

by

LEE WENG SENG

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

Approved:

---

Mr. Patrick Sebastian  
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS  
TRONOH, PERAK

January 2016

## **CERTIFICATION OF ORIGINALITY**

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

---

Lee Weng Seng

## **ABSTRACT**

Current data entry method from exam scripts is a manual process which takes time and prone to incorrect data entry. In addition to entering the total score which is tedious and difficult enough, the individual question scores also have to be recorded. Interactive Handwriting Recognition for Coursework Data Entry project has its purpose to develop a machine that is robust enough to recognize human handwriting digits for multiple scores. In addition, the enrolment process of the handwriting recognition would have an interactive GUI process to ease user enrolment. User enrolment data will be record and store in Neural Network Testing part as comparing element. A web-cam is to be used to capture images of handwritten digit value on the examination cover page. The captured image will be export and preprocessed before feature extraction. For FYP 1, character vector module will be implemented and tested with the GUI interface to record the difference in accuracy compare without GUI user enrolment. After extract feature from selected image, it will then fed into a neural network and based on selected feature extraction module. Neural network will start setting structure and train with various test samples. Then the trained neural network will be tested with its own sample store at user enrolment stage. The accuracy of this project by using user enrolment data entry is expected to be higher and more consistent. The outcome of this project is to obtain a system that is able to recognize handwritten digit value with the lowest variation and create a stable platform. To increase the handwriting recognition rate from web camera and improve the flexibility of user data entry for handwriting recognition.

## **ACKNOWLEDGEMENTS**

I would like to express my special thanks of gratitude to Universiti Teknologi PETRONAS (UTP) for giving me such an opportunity to undertake Final Year Project and perform the knowledge learn in 4 years. This Final Year Project report, entitled 'Interactive Handwriting Recognition using Webcam for Coursework Data Entry', was the output of 2 semester 8 months of effort and support from all party.

Primarily, the author would like to place a sincere gratitude to Mr. Patrick Sebastian, the FYP project supervisor whom have place his trust on the author on his ability on completing this project and contribute toward this project completion. Without his support, advice and guidance, author would never able to complete the project and achieve improvement in this time

Lastly, I would also like to express my sincere thanks to all friends, lecturers, seniors and classmates for all the supportive advice and knowledge sharing on this project. Finally, I would also express my gratitude toward them, their sense of gratitude towards his parents for their encouragement and support. Special thanks to those who the author did not name but encourage and support the author during this 8 months.

## **Table of Contents**

ABSTRACT .....	3
ACKNOWLEDGEMENTS .....	4
CHAPTER 1 .....	8
INTRODUCTION .....	8
BACKGROUND .....	8
PROBLEM STATEMENT .....	9
OBJECTIVES.....	9
SCOPE OF STUDY .....	10
CHAPTER 2.....	11
LITERATURE REVIEW AND THEORY .....	11
BACK PROPAGATION NEURAL NETWORK.....	11
ONE VS ALL CLASSIFIER.....	13
IMAGE PRE-PROCESSING.....	14
Grayscale Selection .....	14
Segmentation Selection .....	15
Graphical User Interface (GUI).....	15
Feature extraction combination module .....	16
Character Vector Module .....	16
Edge Detection Method.....	17
Image Compression module.....	17
CHAPTER 3.....	18
METHODOLOGY .....	18
SOFTWARE & HARDWARE TOOLS .....	19
IMAGE PRE-PROCESSING.....	19
Slant Correction Method .....	20
Feature Extraction Modules .....	20
Combination Feature Extraction Module .....	20
NEURAL NETWORK.....	21
IMAGE ACQUISITION MODULE .....	22
PROTOTYPE.....	23
REGION OF INTEREST (ROI) .....	24

GRAPHICAL USER INTERFACE MODULE.....	24
GRAPHICAL USER INTERFACE COMMAND STEPS .....	25
GANTT CHART & KEY MILESTONES.....	26
CHAPTER 4.....	28
Result and Discussion.....	28
HANDWRITING RECOGNITION TESTS .....	28
TEST 1.....	28
TEST 2.....	30
TEST 3.....	32
CONCLUSION AND RECOMMENDATIONS .....	34
APPENDICES .....	35
REFERENCES .....	40

## Table of Figure

Figure 1: Grayscale selection .....	14
Figure 2: Region of Interest (ROI) .....	15
Figure 3: Captured digit vs Plot digit .....	16
Figure 4: Edge Detection Method .....	17
Figure 5: Process Flow .....	18
Figure 6: Image Processing Steps.....	19
Figure 7: Neural network viewer.....	21
Figure 8: Graphical User Interface .....	22
Figure 9: Prototype .....	23
Figure 10: Region of Interest shown in GUI .....	24
Figure 11: Complete GUI.....	24
Figure 12: Complete GUI process flow.....	25
Figure 13: Test 1 Handwritten Digits vs Average Accuracy .....	29
Figure 14: Example of digit "2" shown in GUI.....	29
Figure 15: Test 2 Handwritten Digits vs Average Accuracy .....	31

## Table of Table

Table 1: Feature Extraction process flow .....	20
Table 2 : FYP 1 Gantt Chart.....	26
Table 3: FYP 2 Gantt Chart.....	27
Table 4Test 1 Handwriting Recognition System Accuracy .....	28
Table 5: Test 2 Handwriting Recognition System Accuracy .....	30
Table 6: Test 3 of Light intensity at different condition .....	32
Table 7: Accuracy comparison between cold light and warm light .....	33



# CHAPTER 1

## INTRODUCTION

### BACKGROUND

Teachers and lecturers are busy with their academic duty especially at examination period for producing exam question and marking the papers. Current score entry method from exam scripts is a manual process which consume time and human error always occur. Besides entering the total scores of few hundred students which is tedious and difficult enough, the scores of each question also have to be recorded as required. It makes the task doubly hard and waste a lot of times which lecturers can be used for researches studies. In order to step down the burden of lecturers there is a need to develop a solution for it with the progress of artificial intelligence (A. I.) in modern technology. Handwriting recognition technology has advance and achieve a standard but is yet to prove a prime solution.

In modern education, students are compulsory to take examination to prove their understanding on the specific subject and the examiner are responsible to key in the information of student's result and scores from the examination paper to keep in record in data form. Human errors are unpredictable and cannot be omitted, with the scores always often key in wrongly or the handwriting of the examiner been misunderstood by another examiner. It cause the student the get a wrong result for the subject. By having a handwriting recognition system for interactive coursework data entry, it will ease the transfer of data from exam paper in a precise and accurate condition.

There are similarities between this project and a previous thesis by Wong entitled, "Handwriting Recognition for Data Entry" [1], which in this project , the goal is to achieve a higher accuracy and create an interactive software for self-enrollment.

## **PROBLEM STATEMENT**

Human handwriting is hard to identify by machine sensor which proves to be a difficult task for handwriting recognition system. Multiple studies have been done to improve the recognition of different person handwriting mostly by using the artificial neural network. By using real time image processing technique, the recognition of handwriting or data entry for multiple scores are capable to be implemented in universities. In addition, the enrolment process for personal data record of the handwriting recognition would have to create an interactive GUI process which allow easier enrollment for users.

## **OBJECTIVES**

The main purpose of this project is to utilize a web camera to capture images of handwritten scores on the examination score sheet and process the image with Artificial Neural Network with the ability to teach the machine to understand the handwriting of specific person. By using user enrollment system, user has to pre-define their handwriting and let the machine to capture their handwritten digits then store the data under the user name. Instead of comparing the new capture handwritten digit images with other handwriting, we will comparing the user digit with his/her own data which we store at the beginning. It will highly increase the accuracy and shorten the time taken for the program to react.

## **SCOPE OF STUDY**

The scope of study for this project is highly related to Mathworks software – MATLAB. In this project Graphical User Interface (GUI), Neural Network Toolbox and Image Processing Toolbox are all features found in MATLAB and will be a great help in simplified the Handwriting Recognition System. By having the basic understanding on Neural Network can help to create a more robust and interactive system. For experimenting unit is limited to Universiti Teknologi PETRONAS (UTP) examination score sheet handwriting recognitions. Handwriting experiment subject will be focus on UTP examination score sheet because it will be used as the format for the segmentation of handwritten recognition system. The handwritten scores from lecturers will be recorded and taken into consideration for data entry system. Furthermore, image processing technique in identify the red color marker pen that lecturer usually used is needed to differentiate student sketches and the results.

## **CHAPTER 2**

### **LITERATURE REVIEW AND THEORY**

#### **BACK PROPAGATION NEURAL NETWORK**

In each hemisphere of human brain consists of hundred million neurons which ease human on understand and recognize digits. It is difficult to write a software to recognize digits like humans when approaching visual pattern recognition. Nevertheless, neural network is a robust network to recognize handwriting digits and provide a high accuracy system. Artificial Neural Network (ANN) is like people but it approach the problem is a different way, it learns through examples which known as training set. Training set contains a large number of handwritten digits which can be used as data comparison to get a high accuracy of success rate [2].

For better recognition, the pre-processed image is interpreted as a form of signal that can propagated through the network of the three layers in forward direction. The neural network is referred as multilayer perceptron. Neural network has to ability to derive meaning from complex data and extract the patterns or trends through training set. It also has the ability include[3]:

1. Adaptive learning: Able to learn the ways of doing task based given in training set.
2. Self-organization: Able to create its own organization during learning time.
3. Real time operation: Able to carried out in parallel computations.

There are three layers in a neural network, which are input layer, hidden layer and output layer. Inside these layer carrier weight and bias which decide the type of output to perform based on the system requirement. In this case we will be using feed-forward network which allow signals to travel only from input to output. No feedback loops will be allowed. There are also feedback network which allow the neural network to reply or direct back to the user if certain points are trigger but it will not be used in this project.

Most of the studies show that the recognition system built from neural network have succeed on providing more than 95% accuracy [4]. Even with this high accuracy, there are still ways to make the learning process smooth and more consistent which is by adding error back propagation algorithm[2].

Throughout the year, neural network has been proven to be a key to unlock the door for future computer processing as it is very successful at solving numerous problem domain. Neural network has been able to counter problems which requires the computer processor to predict and classify a control problem due to neural network ability to learn the structure of building its own path when user train it that way. Neural network also able to carry out parallel computation which will shorten the calculation period.

## **ONE VS ALL CLASSIFIER**

This is a module that used to create a standard classification model that able to predict multiple classes. In this project, it will be used to predict 10 different classes (0 – 9) and it can be used for binary classification as well.

While some classification algorithms naturally permit the use of more than two classes, others are naturally binary (two-class) algorithms. However, binary classification algorithm can be turned into multi-class classification algorithms by a variety of strategies. This module implements the one vs. all method, in which a binary model is created for each of the multiple output classes. Each of the binary models for the individual classes is assessed against its complement (all other classes in the model) as though it were a binary classification issue. Prediction is then performed by running these binary classifiers, and choosing the prediction with the highest confidence score.

In essence, an ensemble of individual models is created and the results are then merged, to create a single model that predicts all classes. Thus, any binary classifier can be used as the basis for a one-vs-all model.

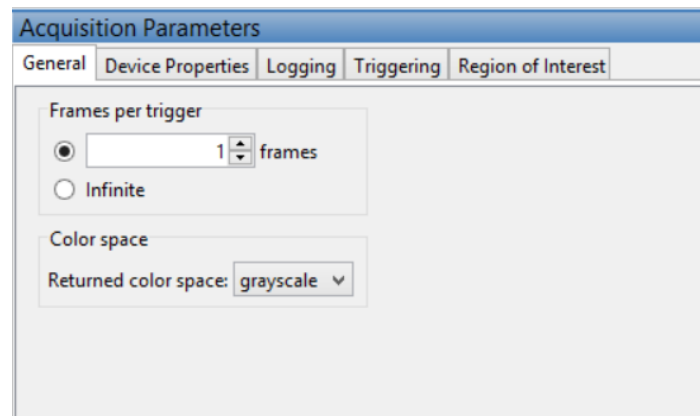
To use this method, at least an untrained two layer classification algorithm have to be feed into the function. Then train the model by using a Train Model with a fix training dataset. For now, feature extraction module has been merge into the ‘train\_predict’ function as it will be used to train the neural. Then neural will start to learn by comparing the feature with the random data from the dataset and isolate the example that are similar with the feature getting from feature extraction module and start feeding the training info into the neural. In this project, total of 10 classifier has been set and the information of each classifier will be convert to array for next process.

## IMAGE PRE-PROCESSING

Before the image being analyzed in the neural network, those sample of data must be pre-process before being train in the neural network. Whenever an image is capture by the web camera and send to the MATLAB, Image Processing Toolbox or by coding function can be used to perform the image processing like the data segmentation, noise reduction and scratch removal. Data segmentation will provide the system a more concentrate area to be analyze from the value inside the box. Noise reduction will reduce the variation of the handwriting digits. Scratch removal will remove the invalid handwritten data which have been scratched out.

### Grayscale Selection

Figure 1 shows the used of grayscale selection before capture the image, it can save number of bits and decrease the size of the image by a little.



*Figure 1: Grayscale selection*

## Segmentation Selection

Segmentation is used to make the system focus on identify and recognize the digit value instead of analyzing the empty spaces. Figure 2 shows the Image Processing Toolbox function found in MATLAB.

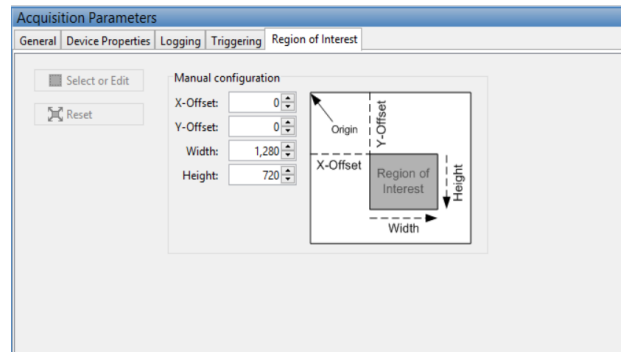


Figure 2: Region of Interest (ROI)

There is an alternative way for image preprocessing by using the function code in MATLAB. The code that being used in this project will be implement at edges detection, image dilation, image filling for round shape number (6,8,0)

## Graphical User Interface (GUI)

When user turn on the system, user data entry will appear for user to enter name, identity number and testing sample. Testing sample will be captured by web camera and send to the database as training set. The stored testing sample will be recognize under user account. GUI is important achieve one of our goal which is to add the flexibility of self-enrollment for users. A GUI program will be create for users to enter the handwriting specimen and user name to be stored and used for comparison during feature extraction. Not much parameters will be shown on the GUI to avoid confusing the users [5]. Tab editor and Model Question Dialog to allow users few options to choose.



## Feature extraction combination module

Combination of Alphabet Profile Module, Modified Character Module and Image Compression Module will be used for the second stage when character vector module succeed. The reason for choosing this combination module is because of the input harvest from this 3 method is the least beside character vector module. Heavy input collection will burden the system, thus defining the perfect ratio is important while at the same time of increasing the accuracy.

## Character Vector Module

There are many methods that can be used but character vector module is the most simple extraction module of image processing of characters. In this module, the digit will be separate into 5 X 7 equal areas. Each element will represent the marked bits during feature extraction testing. Character Vector Module will be the first of our experimenting method and will be compare to a previous thesis result by Wong [1] if we get a more promising result, we will proceed to another method.

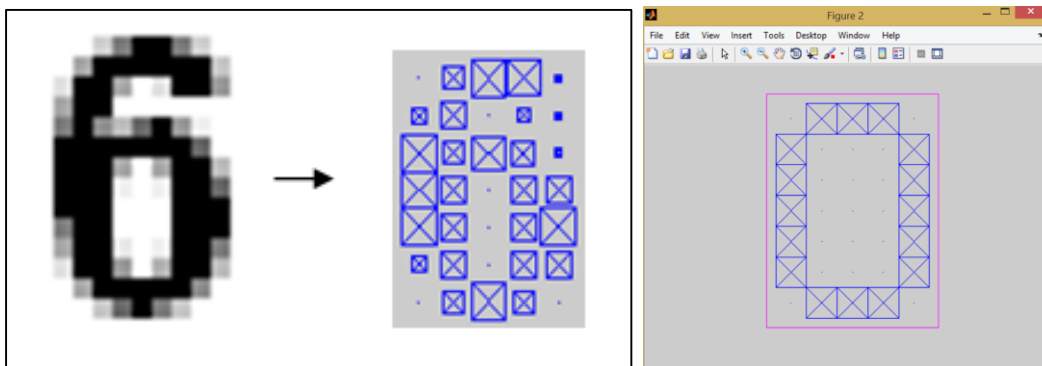
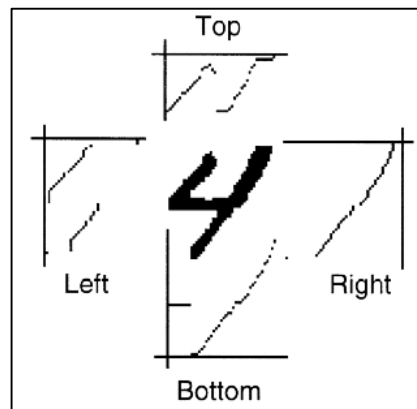


Figure 3: Captured digit vs Plot digit

As shown above, captured image will be separated to 35 equal areas, when there are value detected on the spot, it will show on the plotted graph. The 35 equal areas will be compare to the training set as the structure for defining weight and biases of recognition.

## Edge Detection Method

This method is form based on the sudden changes of discontinuities in an image are called as edges. It generally detect 3 main types of edges (Horizontal, Vertical and Diagonal). As shown in Figure below, we are using Edge Detection Method for analyze edges from 4 direction (top, bottom, left and right). This method will generate more pixels for neural network to analyze and at the mean time will sharpen the image.



*Figure 4: Edge Detection Method*

## Image Compression module

This method used are pretty similar to Matlab function “imgresize” however, image compression module will allow the user to manually select the size to compress and the main pixels will remain as it is and cancel the general or repetitive pixels.

# CHAPTER 3

## METHODOLOGY

Based on the previous study, creating handwriting recognition system by using artificial neural network is the optimal solution for this project through the MATLAB software.

Figure below shows the process flow for the handwriting recognition system with Feature Extraction module combination. In One-vs-all classifier included in the neural network design and the Bias for training set has been fix to 1.

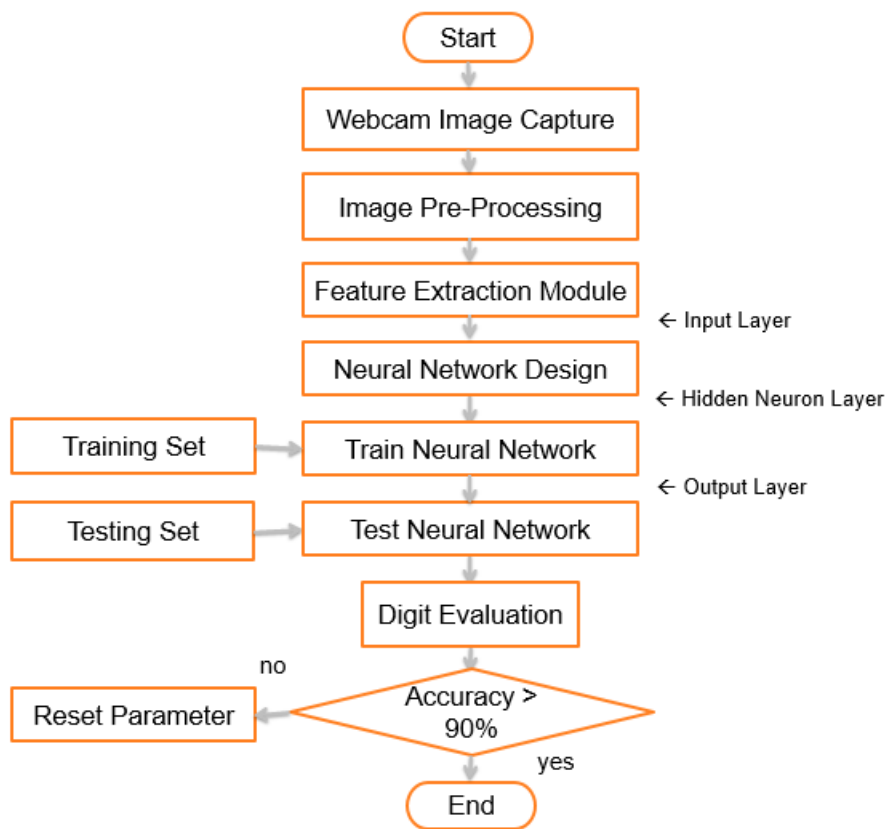


Figure 5: Process Flow

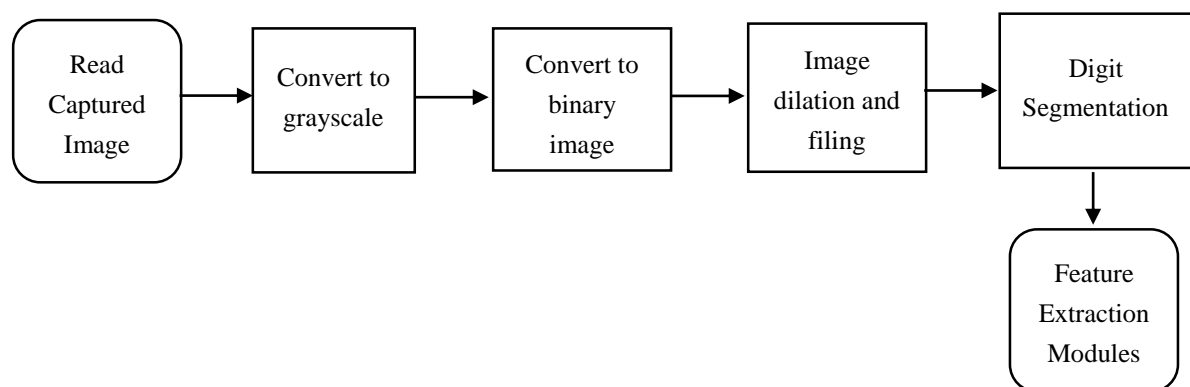
## SOFTWARE & HARDWARE TOOLS

For this project, a web camera will be used to capture handwritten data into MATLAB to be preprocess and train for recognition. Web camera is chosen for its real time image input and algorithms that can be implemented to capture the data in the region of interest (ROI). The captured image will be analyze and stored in a database for user data entry which will be stored under user account for higher accuracy identification.

## IMAGE PRE-PROCESSING

Captured image will be passing through a preprocess function to simplified the recognition process. MATLAB software had already contain an Image Processing Toolbox which has the capabilities to perform ideal image processing such as Greyscale Converter, Region of Interest/Segmentation, Parameterization and Brightness control.

Figure 6 shows the process flow of raw image using MATLAB's Image Processing toolbox. These are the steps for image preprocessing before send to feature extraction module.



*Figure 6: Image Processing Steps*

## Slant Correction Method

This method is to reduce the variation of handwritten digits which varies from the angle in degrees clockwise from vertical at which the characters were drawn. Slant correction allow the system to normalize the slant of the characters in a line or paragraph to the vertical. For this project, we will first crop the empty space detected from 4 direction (top, bottom, left and right) and start adjust the image using gradient direction.

## Feature Extraction Modules

Table below shows 2 tests conducted using only Character Vector Module and Combination Feature Modules.

Test 1				Test 2			
No	Feature Extraction Module	Size of Matrix	Average Accuracy (%)	No	Feature Extraction Module	Matrix Size	Average Accuracy (%)
1	Character Vector	5 x 7	66%	1	Character Vector	5 x 7	91%
				2	Edge Detection	4 x 8 x 8	
				3	Image Compression	8 x 8	
Total output		35		Total size		355	

*Table 1: Feature Extraction module vs Combination Modules*

## Combination Feature Extraction Module

This combination of feature extraction module consists of 3 main module include Character Vector Module, Edge Detection Module and Image Compression module. These 3 modules combine and output a size of 1 x 355 pixels that send into neural to analyze.

## NEURAL NETWORK

The neural network system is the brain behind this interactive handwriting recognition for coursework data entry project. Human identify number by using hemisphere of our brain, the primary visual cortex but machine does not have a brain that can solve this problem. Neural network is one of the way to solve this problem but neural network approach the problem in a different way. The basic idea of neural is to get a large dataset known as training set and feed into the neural during initialize and this will act as the teaching material which teach the neural network what it is facing and what it will classifier. The larger and more details the training set the more the neural network will learn. Neural network consist of input layer, hidden layers and output layers. The number of input neuron is based on the data/pixels collected from image processing. The number of output layer is set to be from 0 to 9, total of 10 output. Examples of designing the network is to encode the intensities of the image pixel into the input neurons. If the image is processed and is a 5 X 7 matrix, then we will have 35 input neurons with the intensities scaled between 0 and 1. It is known that using less amount of neurons will result a low accuracy system. However, if the number of neurons are too much, the system will be overload and affect the accuracy and stability [1].

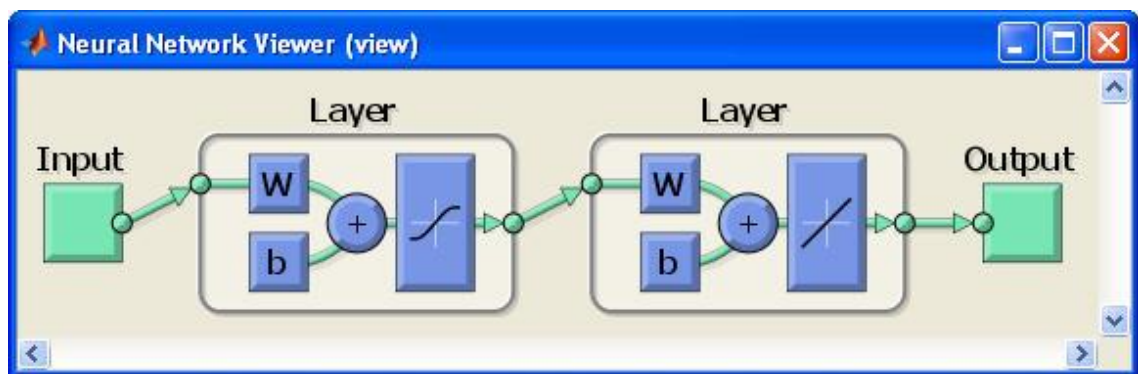


Figure 7: Neural network viewer

## IMAGE ACQUISITION MODULE

Image acquisition plays an important role as it is the step where live image will be captured through web camera before being pre-process by the system. There are 2 ways to capture live image which is through command function or by using image acquisition toolbox. However, after combine with Graphical User Interface (GUI) it would be easier to process by calling out function.

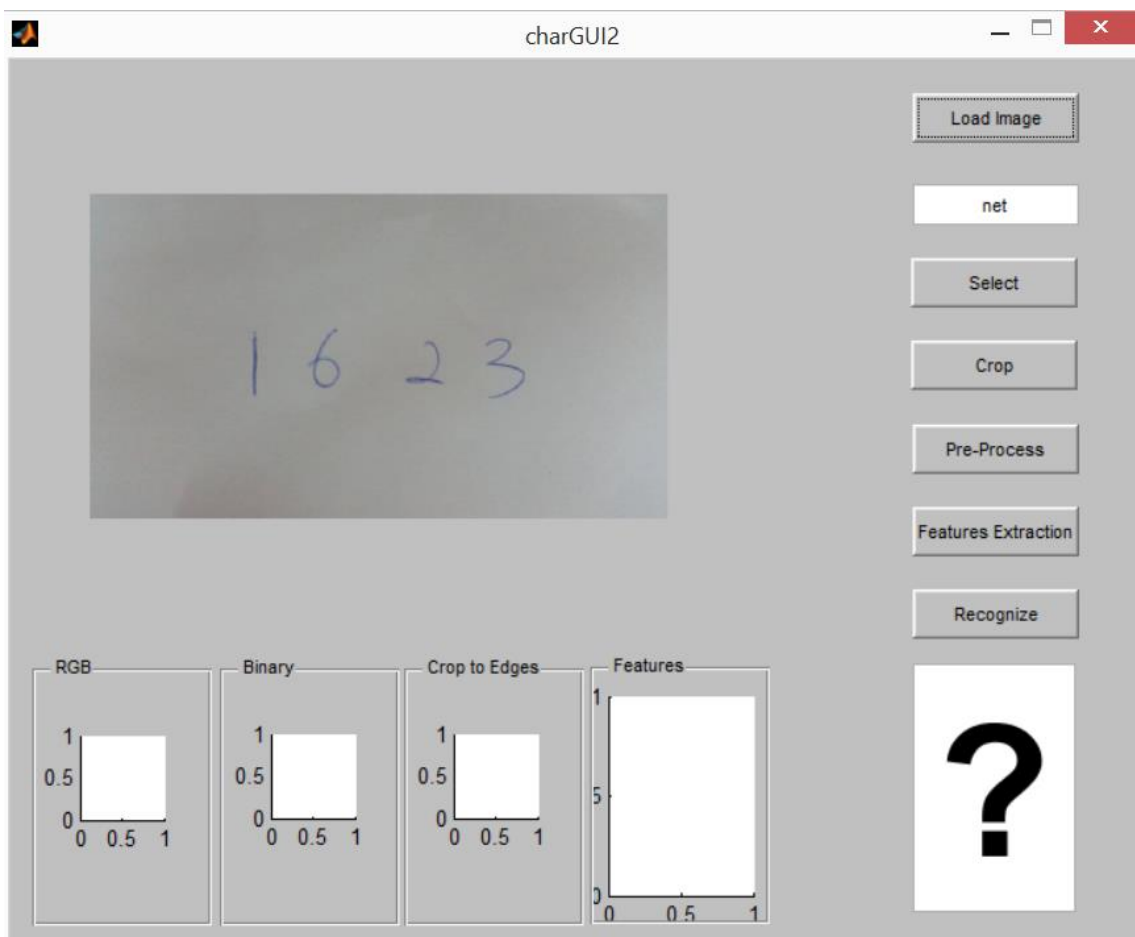


Figure 8: Graphical User Interface

## PROTOTYPE

A simple prototype was built to capture the sample image (score sheet). The camera used was Logitech HD webcam C270 with 1280 x 720 pixels at video capture and up to 3.0 megapixels at photo capture. However without the software update, in real life only 0.9 megapixels image was captured. A table lamp was used to increase the light intensity on image capturing. Okes Cold White light bulb was used with specification of 220-240v, 50-60 Hz and 15W.

Logitech HD webcam c270 specification:

- Video Capture Resolution : 1280 x 720 pixel
- Logitech Fluid Crystal Technology : Video Smoothing Technology
- Photos Capture : Up to 3.0 Megapixels
- Interface : USB 2.0
- Upgradable webcam software : Face tracking, Motion detection and zoom control



Figure 9: Prototype



## REGION OF INTEREST (ROI)

The figure shown below is an example of using region of interest to extract the information contain inside the box. In this project, user will have two option manual or auto. For manual, user can set the ROI by themselves but for only one digit recognition in one box. For auto, user does not need to fix the ROI and can start recognition immediately.

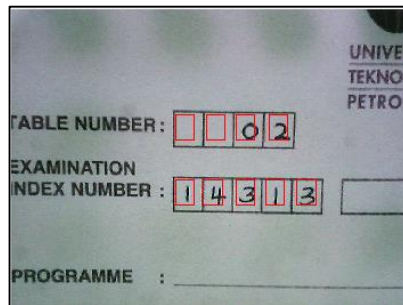


Figure 10: Region of Interest shown in GUI

## GRAPHICAL USER INTERFACE MODULE

Graphical User Interface (GUI) is a user interactive system which enhances the usability between user and the system. GUI is a platform that create to ease user at using the system function.

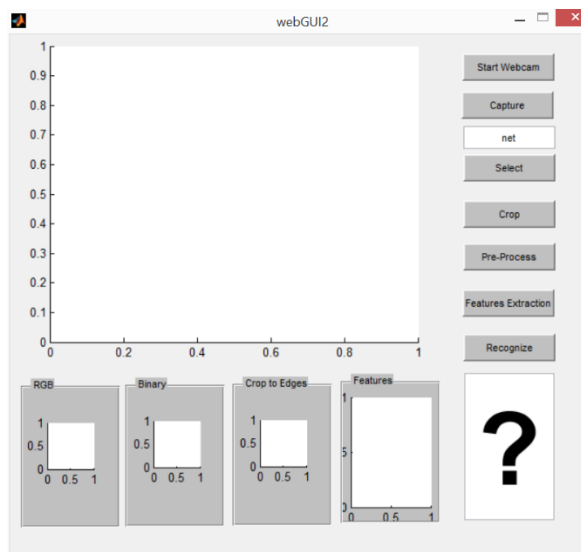


Figure 11: Complete GUI

# GRAPHICAL USER INTERFACE COMMAND STEPS

Figure below shows the command steps when performing digit recognition using GUI.

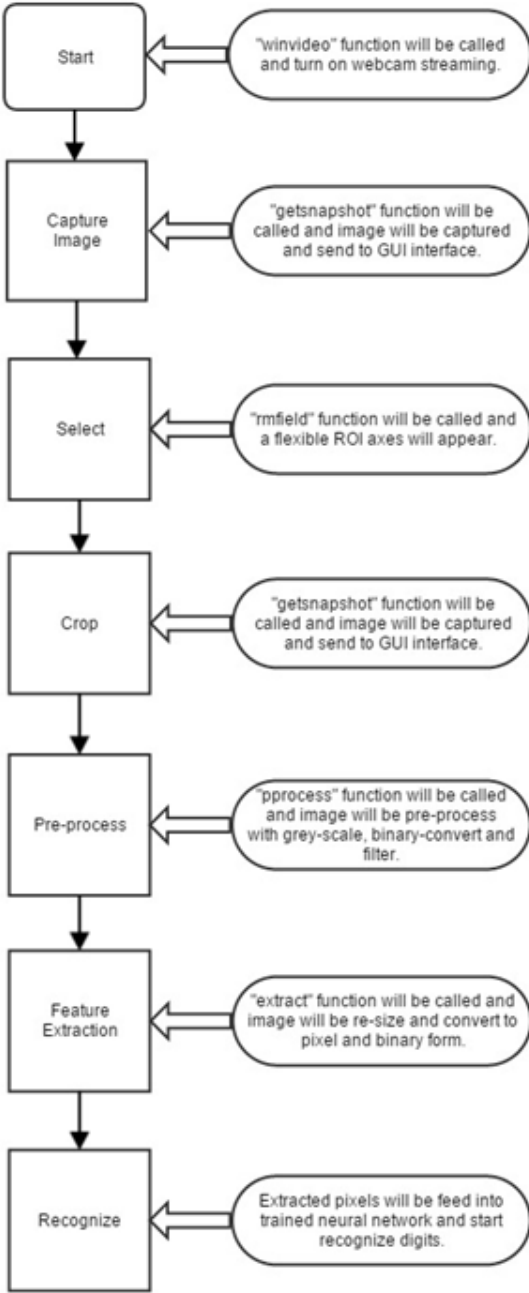




Figure 12: Complete GUI process flow

# GANTT CHART & KEY MILESTONES

The table below shows the timeline of work involved within 14 of FYP 1.

Task/Week	FYP 1													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Literature Review	█	█	█	█	█									
Extended Proposal						●								
Theory understanding						█	█							
Neural Modeling & Modification						█	█	█						
Proposal Defense								●						
Parameter Testing									█	█	█	█	█	█
Interim Report														●

Table 2 : FYP 1 Gantt Chart

-  Key milestone
-  Process

The following tables, Table 3 depict a detailed timeline of the work involved within the 14 weeks of FYP 2 respectively.

Details/Week	FYP 2														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Improve algorithm	█	█	█	█	█										
Sample gathering and testing						█	█								
Progress report								●							
Comparative analysis						█	█	█	█						
Pre SEDEX											●				
Final report (Draft)													●		
Technical Report														●	
Project Viva															●

Table 3: FYP 2 Gantt Chart

● Key milestone

█ Process

## CHAPTER 4

### Result and Discussion

#### HANDWRITING RECOGNITION TESTS

##### TEST 1

Test 1 Handwriting Recognition Test only include Character Vector Module as its feature extraction module. Table below shows the result of recognition system using only character vector module.

Handwritten Numbers	Occurrence	Total Mistakes	Percentage of Accuracy (%)
0	20	1	95.00
1	20	7	65.00
2	20	10	50.00
3	20	13	35.00
4	20	8	60.00
5	20	9	55.00
6	20	4	80.00
7	20	4	80.00
8	20	9	55.00
9	20	3	85.00
Average Accuracy (%)			66.00

*Table 4 Test 1 Handwriting Recognition System Accuracy*

Based on figure below, only digit 0 able to reach above 90%. This is mainly because using only character vector module will cause the system to insufficient pixel to analysis because character vector module only analyze 35 outputs only. Furthermore, digit 2, 3, 5 and 8 always cause problem and confuse the recognition system. This is because Character Vector module are recognizing based on the dot being detected on within the ROI and Character Vector module does not extract enough data from the image to differentiate it.

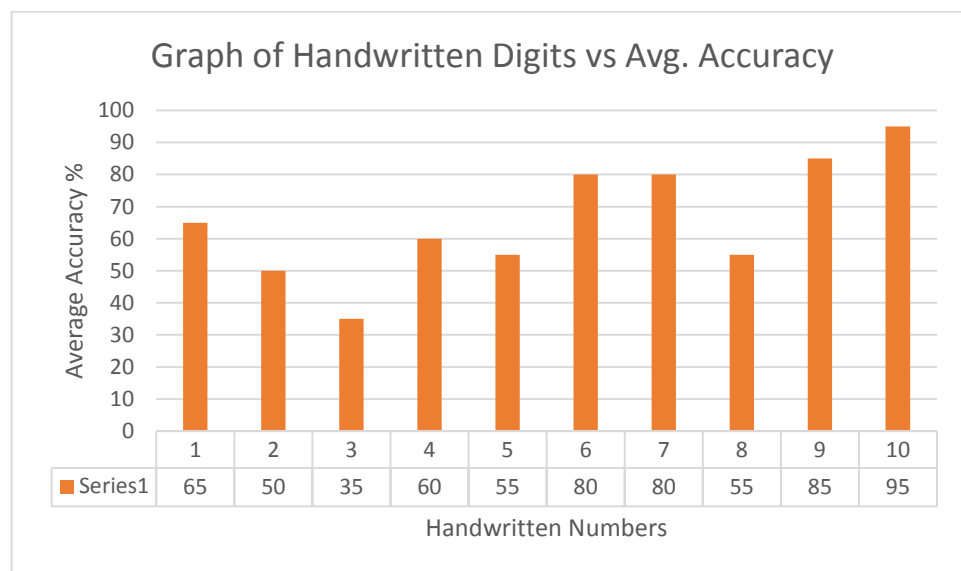


Figure 13: Test 1 Handwritten Digits vs Average Accuracy

Figure below shows an example of digit “2” being recognize as 1 due to slant correction and gradient adjustment.

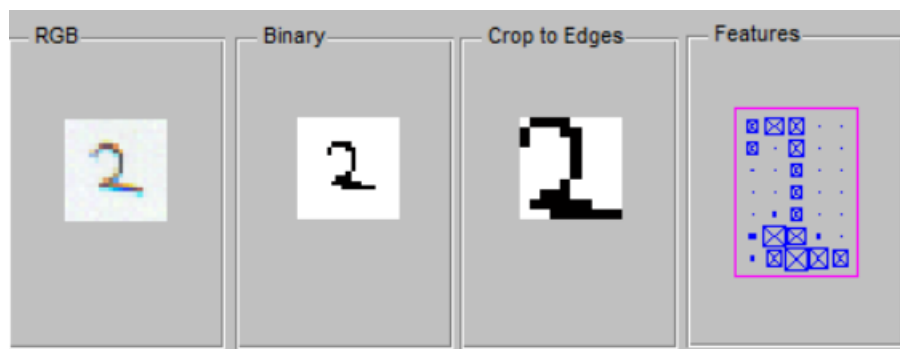


Figure 14: Example of digit "2" shown in GUI

## TEST 2

Test 2 Handwriting Recognition Test include the combination of character vector module, Edge detection method and Image Compression Module as its feature extraction module. Table below shows the result of recognition system using only character vector module. Three scale being used to test the accuracy and effect on the system like auto scale, manual scale and by loading reference data.

Handwritten Numbers	Occurrence (Each)	Total Mistakes	Best Results (%)			Avg. Accuracy (%)
			Auto Scale	Manual Scale	Reference Data	
0	20	2	90	100	100	96.67
1	20	4	90	95	95	93.33
2	20	7	80	90	95	88.33
3	20	6	85	95	90	90.00
4	20	2	95	95	100	96.67
5	20	9	75	90	90	85.00
6	20	4	85	95	100	93.33
7	20	12	65	85	90	80.00
8	20	2	90	100	100	96.67
9	20	3	90	95	100	95.00
System Accuracy						91.00

*Table 5: Test 2 Handwriting Recognition System Accuracy*

Figure below shows the number of handwritten digits vs average accuracy using combination feature extraction module.

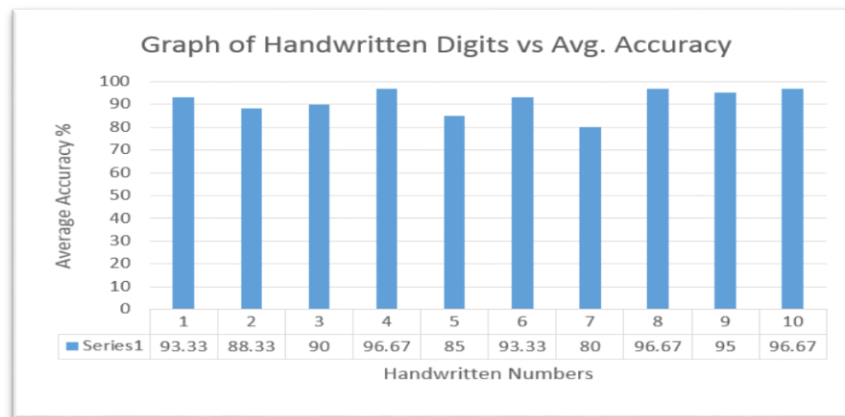


Figure 15: Test 2 Handwritten Digits vs Average Accuracy

For combination feature module, digit “7” was the lowest accuracy and from the GUI result, sometimes 7 will be recognize as digit “1” or “9” thus lower the accuracy of the system.



### TEST 3

Test 3 Handwriting Recognition Test include the combination of character vector module, Edge detection method and Image Compression Module as its feature extraction module but tested with different light source. A cold white light bulb with specification of 220 – 240V/50 – 60 Hz/ 15W and a warm white light bulb with specification of 160 -260V/50 – 60 Hz/ 11W were both used in testing the system.

Table below shows the result of light intensity using HD 9221 photo-radio meter. Photo radio meter is a portable instrument for intensity measurement with range of 400-760nm and radiometer with range of 400-900nm and automatic change of scale. It has precision of 0.15% of reading and the optimal working temperature of 15-75 degree. It also has different measuring unit and for light/ radiometric intensity was measured in watts per meter squared. Table below shows the measured result using HD 9221 photo-radio meter.

Condition	Intensity (W/m <sup>2</sup> )
9am morning (indoor)	0.71 ~ 1.23
Cold White Light	132 ~ 187
Warm White Light	57 ~ 79

*Table 6: Test 3 of Light intensity at different condition*

Table below shows the comparison of accuracy percentage between warm light and cold light.

Handwritten Numbers	Occurrence	Cold Light Accuracy (%)	Warm Light Accuracy (%)
0	20	95.00	90.00
1	20	92.25	95.00
2	20	85.00	75.00
3	20	90.00	80.00
4	20	95.00	90.00
5	20	82.25	70.00
6	20	90.00	75.00
7	20	75.00	75.00
8	20	95.00	85.00
9	20	92.25	90.00

*Table 7: Accuracy comparison between cold light and warm light*

Table above shows that cold light are a better light source compare to warm light because using warm light will generate more noise which cannot be seen by using human eyes but can only be detected using filtering.

## **CONCLUSION AND RECOMMENDATIONS**

Handwriting recognition algorithm has been study and research by many parties for a long time, however it have yet to reach a 100% accuracy that can be used in commercial or industrial area due to largely different samples. This project does not just increase the accuracy from the previous project, it also increase the flexibility of self-enrollment of users for handwriting recognition.

As discussed before, neural network can provide a high accuracy of recognition rate but the system are not stable as the testing data getting larger and more complicated, the successful rate will decrease dramatically as shown by Wong [1]. By limiting the testing set and compare the user's handwriting with his/her own data set we able to get a higher accuracy and more robust system.

If by limiting the testing set on Character Vector Module achieve a more stable but low accuracy result and the combination feature extraction module result a higher accuracy and stable recognition system. Moreover, different light source result to be affecting the system accuracy.

In conclusion, one-vs-all classifier prove to be a stable and simple pattern fitting neural network tools. It simplify the iteration and able to perform in high accuracy even the bias is set as 1. The combination of feature extraction module can be improve by having more variation to test the accuracy of each set.

## APPENDICES

```
nn_3.m
%% Initialization
clear ; close all; clc

%% Setup the parameters
input_layer_size = 400; % 20x20 pixel
hidden_layer_size = 25; % 25 hidden units
num_labels = 10; % 10 labels, from 1 to 10

%%
% Load Training Data
fprintf('Loading and Visualizing Data ...\n')

load('ex3data1.mat');
m = size(X, 1);

% Randomly select 100 data points to display
sel = randperm(size(X, 1));
sel = sel(1:100);

displayData(X(sel, :));

fprintf('Program paused. Press enter to continue.\n');
pause;

%%

fprintf('\nLoading Saved Neural Network Parameters ...\n')

% Load the weights into variables Theta1 and Theta2
load('ex3weights.mat');

%% Predict

pred = predict(Theta1, Theta2, X);

fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) *
100);

fprintf('Program paused. Press enter to continue.\n');
pause;

% Randomly permute examples
rp = randperm(m);

for i = 1:m
    % Display
    fprintf('\nDisplaying Example Image\n');
end
```

```

displayData(X(rp(i), :));

pred = predict(Theta1, Theta2, X(rp(i),:));
fprintf('\nNeural Network Prediction: %d (digit %d)\n', pred,
mod(pred, 10));

% Pause
fprintf('Program paused. Press enter to continue.\n');
pause;
end

```

---

onevsall.m

```

function [all_theta] = oneVsAll(X, y, num_labels, lambda)
%ONEVSALL

%variables
m = size(X, 1);
n = size(X, 2);

% Return variable
all_theta = zeros(num_labels, n + 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

for c = 1:num_labels
    init_theta = zeros(n + 1, 1);
    options = optimset('GradObj', 'on', 'MaxIter', 50);
    [theta] = fmincg (@(t) (lrCostFunction(t, X, (y == c), lambda)),
    init_theta, options);
    all_theta(c,:) = theta';
end;

%

end

```

---

predictonevsall.m

```

function p = predictOneVsAll(all_theta, X)
%PREDICT

m = size(X, 1);
num_labels = size(all_theta, 1);

% You need to return the following variables correctly
p = zeros(size(X, 1), 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

```

```

%
sig = sigmoid(X * all_theta');
[maxSig, maxSig_2] = max(sig');
p = maxSig_2';

```

```

%
```

```

end

```

---

sigmoid.m

```

function g = sigmoid(z)
%SIGMOID Compute sigmoid function
% J = SIGMOID(z) computes the sigmoid of z.

g = 1.0 ./ (1.0 + exp(-z));
end

```

---

preprocess.m

```

function img = edu_imgpreprocess(I)

Igray = rgb2gray(I);

Ibw = im2bw(Igray,graythresh(Igray));

Iedge = edge(uint8(Ibw));

se = strel('square',3);

Iedge2 = imdilate(Iedge, se);

Ifill= imfill(Iedge2,'holes');

[Ilabel num] = bwlabel(Ifill);

Iprops = regionprops(Ilabel);

Ibox = [Iprops.BoundingBox];
Ibox = reshape(Ibox,[4 50]);

Ic = [Iprops.Centroid];
Ic = reshape(Ic,[2 50]);
Ic = Ic';
Ic(:,3) = (mean(Ic.^2,2)).^(1/2);
Ic(:,4) = [1:50];

```

```

% Extra lines compare to example2 to extract all the components into
an
% cell array
Ic2 = sortrows(Ic,2);

for cnt = 1:5
    Ic2((cnt-1)*10+1:cnt*10,:) = sortrows(Ic2((cnt-
1)*10+1:cnt*10,:),4);
end

Ic3 = Ic2(:,1:2);
ind = Ic2(:,4);

for cnt = 1:50
    img{cnt} = imcrop(Ibw,Ibox(:,ind(cnt)));
end

```

---

#### Imresize.m

```

function lett = edu_imgresize(bw2)
% This function will take the cropped binary image and change it to 5
x 7
% character representation in single vector.

bw_7050=imresize(bw2,[70,50]);
for cnt=1:7
    for cnt2=1:5
        Atemp=sum(bw_7050((cnt*10-9:cnt*10),(cnt2*10-9:cnt2*10)));
        lett((cnt-1)*5+cnt2)=sum(Atemp);
    end
end

lett=((100-lett)/100);
lett=lett';

```

---

#### imcrop.m

```

function bw2 = edu_imgcrop(bw)

% Find the boundary of the image
[y2temp x2temp] = size(bw);
x1=1;
y1=1;
x2=x2temp;
y2=y2temp;

% Finding left side blank spaces
cntB=1;
while (sum(bw(:,cntB))==y2temp)
    x1=x1+1;
    cntB=cntB+1;
end

```

```

% Finding right side blank spaces
cntB=1;
while (sum(bw(cntB,:))==x2temp)
    y1=y1+1;
    cntB=cntB+1;
end

% Finding upper side blank spaces
cntB=x2temp;
while (sum(bw(:,cntB))==y2temp)
    x2=x2-1;
    cntB=cntB-1;
end

% Finding lower side blank spaces
cntB=y2temp;
while (sum(bw(cntB,:))==x2temp)
    y2=y2-1;
    cntB=cntB-1;
end

% Crop the image to the edge
bw2=imcrop(bw,[x1,y1,(x2-x1),(y2-y1)]);

```



## REFERENCES

- [1] W. Y. XIANG, "Handwriting Recognition Using Webcam For Data Entry," 2014.
- [2] W. L. Goh, D. P. Mital, and H. A. Babri, "<An artificial neural network approach to handwriting recognition.pdf>," 1997.
- [3] T.-S. Yan, Y.-Q. Tao, and D.-w. Cui, "<Research on handwritten numeral recognition method based on improved genetic algorithm and neural network.pdf>," 2007.
- [4] Zili-Chen and Zuxue-Wei, "<A Handwriting Numeral Character Recognition System.pdf>."
- [5] R. Arnold and P. Miklós, "<Character recognition using neural networks.pdf>," 2010.
- [6] W. Wellner, L. Jessica, and S. Caleb, "<Recognition of Handwritten Digits using a Neural Network.pdf>," 2002.
- [7] C. Zili and W. Zuxue, "A Handwriting Numeral Character Recognition System," in *Multimedia Technology (ICMT), 2010 International Conference on*, 2010, pp. 1-5.
- [8] Y. Fujisawa, S. Meng, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," in *Document Analysis and Recognition, 1999. ICDAR '99. Proceedings of the Fifth International Conference on*, 1999, pp. 277-280.
- [9] Y. Tai-Shan, T. Yong-Qing, and C. Du-wu, "Research on handwritten numeral recognition method based on improved genetic algorithm and neural network," in *Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on*, 2007, pp. 1271-1276.
- [10] W. Xin, T.-l. Huang, and X.-y. Liu, "Handwritten Character Recognition Based on BP Neural Network," in *Genetic and Evolutionary Computing, 2009. WGEC '09. 3rd International Conference on*, 2009, pp. 520-524.
- [11] R. Arnold and P. Miklos, "Character recognition using neural networks," in *Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on*, 2010, pp. 311-314.
- [12] C. Halder, J. Paul, and K. Roy, "Comparison of the classifiers in Bangla handwritten numeral recognition," in *Radar, Communication and Computing (ICRCC), 2012 International Conference on*, 2012, pp. 272-276.
- [13] P. Zhang, T. D. Bui, and C. Y. Suen, "Hybrid feature extraction and feature selection for improving recognition accuracy of handwritten numerals," in

- Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, 2005, pp. 136-140 Vol. 1.
- [14] K. Yamada, H. Kami, J. Tsukumo, and T. Temma, "Handwritten numeral recognition by multilayered neural network with improved learning algorithm," in *Neural Networks, 1989. IJCNN., International Joint Conference on*, 1989, pp. 259-266 vol.2.
- [15] B. V. S. Murthy, "Handwriting recognition using supervised neural networks," in *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, 1999, pp. 2899-2902 vol.4.
- [16] F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Handwritten numeral recognition using autoassociative neural networks," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, 1998, pp. 166-171 vol.1.