ENG499 CAPSTONE ELECTRONICS PROJECT

NAME: Lim Theng Kwang

PI NO: B0401878

PROJECT TITLE: Face Recognition Home Security System

PROJECT REFERENCE NO: JAN07/BEHE/06

Abstract

The Face Recognition Home Security System (FRHSS) is used to prevent intruders from entering any home premises. The objective is to integrate a home security system and to detect burglary and hence prevent loss of personnel belongings. There are currently many of different types of FRHSS available in the market. Although the system has a relatively high accuracy and reliability in preventing intruders, it still has its flaws. In order to determine better ways to implement a better accuracy to prevent intruders, it is important to understand this project attempt to use the working principles of existing FRHSS. Based on the knowledge gain from the studies, different processes required to improve the existing security system could be drawn out. The main aim of this project was to implement, design and find much better ways and methods to improve the performance of the Face Recognition Home Security System.

Acknowledgement

I would like to show my deepest thankfulness to the following individuals for their continuous help and support to make this project to be carried out and be implemented.

My Project Tutor, Dr. Eicher Low for his patience and guidance throughout the whole process of the project. He guided me through the development process and answered almost all my questions and doubts.

My UniSIM friends Thenna, Chang Feng, Gaph and Eric for their continuous encouragement and also gratefulness to have them around and made my studies in UniSIM so enriching, exciting and fruitful.

My manager at work, Mr. Teng Ling Foong, for his thoughtfulness and understanding of letting me to self manage both my works and my project to be carried equally and also his willingness too by letting me to take time off from work to complete the project.

My family members especially my dad for his financial and my mum for her emotional support in making my studies in UniSIM worthwhile.

Last but not least, to my dearest wife Eunice, for her constant outpouring of love, understanding, and support. In my darkest hours, she has been the wind beneath my wings.

Table of content

Page

ABSTRACT i

ACKNOWLEDGEMENTS ii

TABLE OF CONTENTS iii,iv,v,vi

CAPTIONS FOR FIGURES v

CAPTIONS FOR TABLES vi

CHAPTER 1 INTRODUCTION 1
1.1 Face Recognition Home Security System Background 1
1.2 Project Objective 2
1.3 Overall Objective 2
1.4 Proposed Approach 3
1.5 Face Recognition System Design Overview 3

CHAPTER 2 PROJECT SCOPE 5
2.1 Scope of Work 5
2.2 Project plan 5

CHAPTER 3 LITERATURE REVIEWS 9
3.1 History of Face Recognition 9
3.2 Face Recognition Techniques 10
3.3 Face Recognition Algorithms 11
3.3.1 Eigenfaces 11
3.3.2 Fisherface 12
3.3.3 Hidden Markov Model 13

CHAPTER 4 IMPLEMENTATION GUI FOR FACE RECOGNITION SYSTEM 14
4.1 Graphical User Interface 14
4.2 How this GUI system works 15

CHAPTER 5 OVERVIEWS OF HARDWARE 16

CHAPTER 6 OVERVIEWS OF LABVIEW SOFTWARE APPLICATION 17
6.1 Using NI-IMAQ-USB camera to capture face image 17
6.1.1 IMAQ Create 17
6.1.2 IMAQ USB Grab Setup.vi 18
6.1.3 IMAQ USB Grab Acquire 18
6.1.4 IMAQ USB Close.vi 18

6.1.5 While Loop 19
6.1.6 Block Diagram 20
6.1.7 Stacked Sequence Structure 21
6.1.8 Case Structure 22
6.2 Using Histogram Technique and calculated array to compare two face images 23
6.2.1 IMAQ Create 23
6.2.2 IMAQ Extract ColorPlanes 24
6.2.3 IMAQ Add 25
6.2.4 IMAQ Histogram 26
6.2.5 Unbundle By Name 28
6.2.6 Delete From Array 28
6.2.7 Add Array Elements 29
6.2.8 Block Diagram 30
6.3 Passwords Authentication Entry for final phase of the Face Recognition 30
6.3.1 Property Mode 30
6.3.2 Equal gate 31
6.3.3 OR gate 32
6.3.4 Block Diagram 33
6.4 Useful Words or Messages to the user 33
6.4.1 String Constant 33
6.4.2 Concatenate Strings 34
6.4.3 One Button Dialog 34
6.4.4 Block Diagram 34

CHAPTER 7 CIRCUIT FLOW OF THE FULL FACE RECOGNITION SYSTEM 35

CHAPTER 8 TEST RESULTS 39
8.1 Missing features on face images 40
8.1.1 Disscussion 40
8.2 Different expression on face images 41
8.2.1 Disscussion 41
8.3 Different degree postures on face images 42
8.3.1 Disscussion 42
8.4 Different turning postures on face images 43
8.4.1 Disscussion 43
8.5 Different distances on face images 44
8.5.1 Disscussion 44

CHAPTER 9 CONCLUSION 45

CHAPTER 10 FUTURE WORKS 46

CHAPTER 11 CRITICAL REVIEW AND REFLECTION 47
11.1 Summary of the Project Development 47
11.2 Skill Sets for the Project Development 48
11.3 Problems encountered and Problems solution 49

REFERENCES 51

CAPTIONS FOR FIGURES

Figure1 Proposed flowchart for the Face Recognition System 4
Figure2 GUI for Face Recognition System 14
Figure3 IMAQ Create 17
Figure4 IMAQ USB Grab Setup.vi 18
Figure5 IMAQ USB Grab Acquire 18
Figure6 IMAQ USB Close.vi 18
Figure7 While Loop 19
Figure8 Block Diagram of capturing face image using USB camera 20
Figure9 Stacked Sequence Structure 21
Figure10 Case Structure 22
Figure11 IMAQ Create 23
Figure12 IMAQ ExtractColorPlanes 24
Figure13 IMAQ Add 25
Figure14 IMAQ Histogram 26
Figure15 Unbundle By Name 28
Figure16 Delete From Array 28
Figure17 Add Array Elements 29
Figure18 Block Diagram of the Histogram Techniques 30
Figure19 Property Mode 30
Figure20 EQUAL Gate 31
Figure21 OR gate 32
Figure22 Block Diagram of the Passwords Authentication Entry 33
Figure23 String Constant 33
Figure24 Concatenate Strings 34
Figure25 One Button Dialog 34
Figure26 Block Diagram for Dialog Box 34
Figure27 Block Diagram of the 1st Web Camera activation 35
Figure28 Block Diagram of the 2nd Web Camera activation 35
Figure29 Block Diagram of the two face images for processing 36
Figure30 Block Diagram of the calculated array 36
Figure31 Block Diagram of the calculating the array variance between two face images 37
Figure32 Block Diagram of the Passwords Authentication Entry 37
Figure33 Block Diagram of showing a dialog box 38
Figure34 Block Diagram of the clearing two face images 38
Figure35 Black background with no face image 39
Figure36 Black Background with face image 39
Figure37 Face Images features 40

Figure38 **Face Images expressions** 41
Figure39 **Face Images degree postures** 42
Figure40 **Face Images turning postures** 43
Figure41 **Face Images distances** 44

CAPTIONS FOR TABLES

Table1 **Gantt Chart** 5
Table2 **Video Pro Amp** 16
Table3 **Advanced Setting** 16

CHAPTER 1 INTRODUCTION

1.1 Face Recognition Home Security System Background

In today's society, it is very common for Singaporeans to have a small family due to the competitive nature of our society. When a couple gets married, it is also quite common for both husband and wife to work in the first few years of their marriage life to save enough for the future off springs. As a result, their home will be left empty most of the day until they get back from work at night.

With 80% of our population living in the high rise government flat, most couples have very routine daily pattern of when they leave their home and when they will be back. This will make them to be the easy target for intruder to break into their home. There are many agencies in Singapore that will provide residential home security such as CISCO but their service comes with a premier price tag. With the technology becoming more mature, there are many more home security products such as X10 are available in the market for private residential home installation.

This project will take this opportunity to explore the Face Recognition Home Security System (FRHSS).In most of the systems, biometric feature such a finger print, hand geometry, Iris, voice and face pattern recognition have so far been integrated into the commercial products. This project will attempt to develop a Face Recognition Home Security System (FRHSS).It can also serve as a biometric home security system when integrated with an existing pin coded security system. The FRHSS provides a keyless entry into the home premise via checking the biometric data bank for a match and a preset pin. The alarm of the security system will sound whenever there is a mismatch with the pre-stored data bank or when wrong pin is entered.

1.2 Project Objective

In this project, a Face Recognition System (FRS) is to be developed. In this system, the owners' face image for identification purposes is used to check against a pre-loaded data base to prevent intruder who is trying to break into the house.

1.3 Overall Objective

We need to move along with the latest technology in Singapore in order to stay in touch with current technological advancement. In Singapore, home security system has become more common nowadays. The reason being today's married couples tend to live near their in-laws but not together in the same household. Most of them are pursuing their own careers, single income where the wife stays at home as housewife has become history of the past. As a result, their house will be left empty for at least about 10 hours a day. With the price of a home security dropping, we observe an increased trend in such a system being installed even in the government built Housing Development Board (HDB) flat.

The possible use of the FRHSS to enhance existing home security system and bring people with greater trust and security to the home security system. The importance of this FRHSS is to let us know that this much higher level home security system can be a great help in all houses premises in the near future so we can totally wipe out all the unnecessary home keys. There is this phase saying "Dogs are man's best friend" and with this new home security system bring into our home premises: "FRHSS are home's best friend".

1.4 Proposed Approach

This section outlines the proposed approach that will be carried out in order to implement the Face Recognition System.

1.5 Face Recognition System Design Overview

Face Recognition System Design will be implemented by integrating three main sections, which are as follow:

- Using NI-IMAQ-USB camera to capture face images
- Using Histogram Technique to compare two face images
- Using Passwords Entry Verification for the complete of Face Recognition System

The functions of each section are all controlled by a software application called LABVIEW. The implementation of each section goes through a typical flow of design and implementation, after which all the sections will be integrated to form a Face Recognition System. Once it is completed, it put to a test and scan faces images for the overall performance of the Face Recognition System.

Figure 1 below shows the proposed flowchart for the Face Recognition System

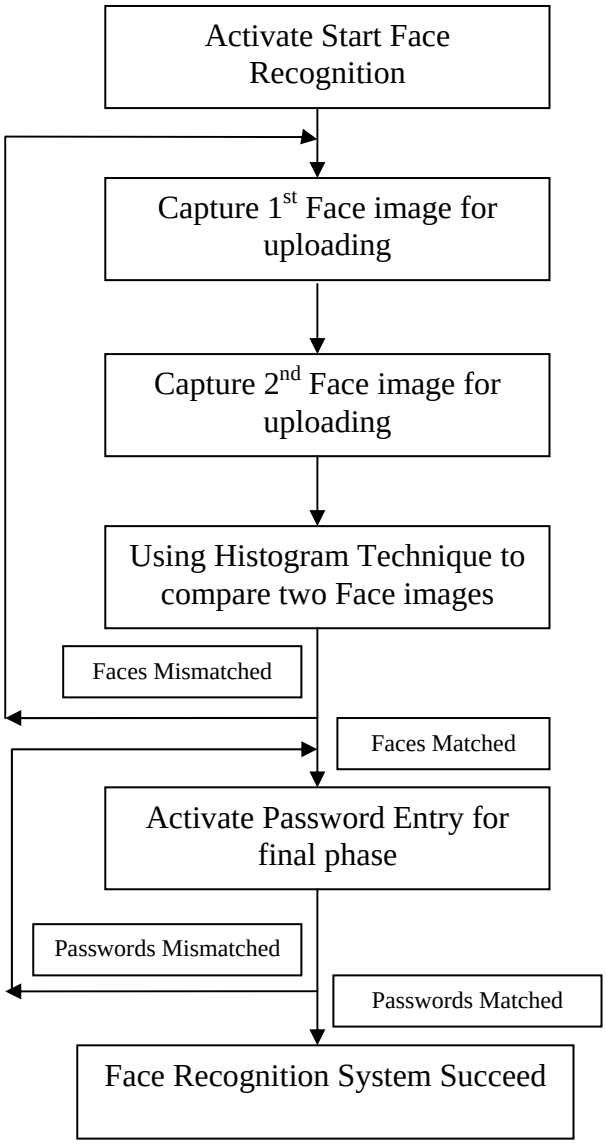


Figure 1 Proposed flowchart for the Face Recognition System

CHAPTER 2 PROJECT SCOPE

2.1 Scope of Work

The scope of work involves both hardware and software aspects. Hardware aspects include PCs and cameras. And software aspects include the implementation of the Face Recognition Algorithm and Software Application for the Face Recognition System.

2.2 Project plan

The concept of this Gantt Chart (Table 1) is taken from the Micromouse Project (HESZ341) that was done in the third year in UniSIM and indeed it reflected well as it can be able to complete the whole project in an orderly manner.

MONTH'07	APRIL	MAY	JUNE	JULY	AUGUST	SEPTEMBER	OCTOBER	NOVEMBER	DECEMBER
APPROVAL OF FPA FORM									
Received Approval from Internal Examiner									
RESEARCH AND DISCUSSIONS									
Literature Review for Face Recognition System									
HARDWARE DESIGN									
Different usage of CDD Cameras									
Selection of CDD Cameras									
SOFTWARE DESIGN									
Different kinds of Digital Image Processing Techniques									
Different kinds of Face Recognition Algorithms									
Fundamental using Matlab									
Fundamental using C++									
Fundamental using Java									
Selection of Digital Image Processing Techniques									
Selection of Face Recognition Algorithms									
Selection of Software for Face Recognition System									
SOFTWARE AND HARDWARE DESIGN									
Testing and Troubleshooting									
REPORT									
Preparation of Final Report Writing									
Submission of Final Report									
PRESENTATION									
Preparation of Slides									
Oral Presentation									

Table 1 Gantt Chart

The Gantt chart in Table 1 shows the project development schedule. The whole project has been segregated into various steps, there are:

1. Literature Review for Face Recognition System
2. Different usage of CDD Cameras
3. Selection of CDD Cameras
4. Different kinds of Digital Image Processing Techniques
5. Different kinds of Face Recognition Algorithms
6. Fundamental using Matlab
7. Fundamental using C++
8. Fundamental using Java
9. Selection of Digital Image Processing Techniques
10. Selection of Face Recognition Algorithms
11. Selection of Software for Face Recognition System
12. Testing and Troubleshooting
13. Preparation of Final Report Writing
14. Submission of Final Report
15. Preparation of Slides
16. Oral Presentation

The project proposal had been accepted since April 2007 onwards and according to the proposed method of implementation and the available time for completion, which will take about seven months to complete. Moreover, the project can only be done during any available spare time and on weekends; this is due to work, family and other commitments.

Yellow Coloured Rectangle has been set in the project schedule and achieving those Yellow Coloured Rectangle at the targeted time will mean that the progress of the project is on track. Grey Coloured Rectangle has been set too in the project schedule. And it indicated the deadline of the submission of the Final Report and the Oral Presentation Day.

And of course there will be problems and it solutions that might occur in each of these steps and may delay the progress of the project are as follow:

1. Literature Reviews on Face Recognition System

- The only research for this project is going through internet and definitely its resources will be at it minimum.
- This problem was resolved through the assistance from my tutor Dr Eicher Low who recommended a book named Digital Image Processing written by Shapiro.

2. Types of CCD Cameras use in the markets.

- Different types of CCD cameras are available everywhere in the markets but a good CCD camera is ideal and will function well in this project because Image resolutions and Pixel accuracy played the main characters for processing face images for Face Recognition System.
- Due to cost constraint in this project, a USB Web Camera is used.

3. Types of Digital Image Processing Techniques to do Face Recognition System.

- There are different types of Digital Image Processing Techniques such as histogram manipulation, pattern matching, filtering, thresholding, segmentation, edge detection.
- Each of them performed different kinds of roles and they will justify the level of successfulness in this project.
- In this project, a Histogram technique was used and it will explain in the later part.

4. Types of software and algorithms available for Face Recognition System.

- At the moment, the most commonly used software are C++ programming, Java and MATLAB and to understand all of them well is a tedious tasks because each of them have its own method of programming and advantages.
- Most commonly used Face Recognition Algorithms are Eigenfaces method, Fisherface method and Hidden Markov model and each of them its own techniques and ways of processing Face Recognition
- LABVIEW Software Application was chosen for this project including its face recognition algorithm technique which will be explained and presented in the later part too.

5. Testing and troubleshooting.

- LABVIEW Software Application may have bugs, which needs time to resolve, therefore more time is catered for this activity.

CHAPTER 3 LITERATURE REVIEWS

3.1 History of Face Recognition

Facial recognition systems are computer-based security systems that are able to automatically detect and identify human faces. The first step for a facial recognition system is to recognize a human face and extract it from the rest of the scene. Next, the system measures nodal points on the face, such as the distance between the eyes, the shape of the cheekbones and other distinguishable features. These nodal points are then compared to the nodal points computed from a database of pictures in order to find a match. Obviously, such a system is limited based on the angle of the face captured and the lighting conditions present. New technologies are currently in development to create three-dimensional models of a person's face based on a digital photograph in order to create more nodal points for comparison.**[1]**

The subject of face recognition is as old as computer vision, both because of the practical importance of the topic and theoretical interest from cognitive scientists. Despite the fact that other methods of identification (such as fingerprints, or iris scans) can be more accurate, face recognition has always remains a major focus of research because of its non-invasive nature and because it is people's primary method of person identification. **[2]**

Automated face recognition is a relatively new concept. Developed in the 1960s, the first semi-automated system for face recognition required the administrator to locate features(such as eyes, ears, nose and mouth) on the photographs before it calculated distances and ratios to a common references point, which were then compared to reference data.

In1970s, Goldstein, Harmon and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition. The problems with both of these were manually computed.

In 1988, Kirby and Sirovich applied principle component analysis, a standard linear algebra technique, to the face recognition problem. This was considered somewhat of a milestone as it showed that less than one hundred values were required to accurately code a suitably aligned and normalized face image.

In 1991, Turk and Pentland discovered that while using the eigenfaces techniques, the residual error could be used to detect face in images – a discovery that enabled reliable real-time automated face recognition systems. **[3]**

3.2 Face Recognition Techniques

Pattern recognition and matching consists of classifying, processing the input and matching it with a known pattern. Face recognition is a very complex form of pattern recognition. It consists of classifying highly ambiguous input signals, with multiple dimensions and matching them with the know ’signals’. Classifying a pattern with high dimensions requires a restrictively large number of training samples. A number of ways have been proposed to solve this problem. Finding an effective means to reduce the dimensionality is the first step in face recognition. Considering the face to be a matrix of values reduces the dimensions to a single dimension. Bit such an approach would be helpful in solving the problem if we were interested in faces that vary by certain transformations, such as magnification translation etc. But biological systems, such as faces do not vary just by transformations, for example changed expression on a face cannot be captured by simple transformations in one -dimensional space. **[4]**

Face Recognition can be classified into two different classes

1. Feature Based (Geometric)
2. Template Based (Photometric)

3.3 Face Recognition Algorithms

3.3.1 Eigenfaces

Many approaches to the overall face recognition problem have been devised over the years, but one of the most accurate and fastest ways to identify faces is to use what is called the “eigenface” technique. The eigenface technique uses a strong combination of linear algebra and statistical analysis to generate a set of basis faces--the eigenfaces--against which inputs are tested. **[5]**

Eigenface recognition derives it's name from the German prefix "eigen", meaning "own" or "individual" (Ponti, 1999). The eigenface recognition approach was developed by Turk and Pentland (1991), both colleagues from MIT, in 1987. The eigenface method of facial recognition is considered the first working facial recognition technology (Velasco, 1998).

The eigenface recognition system begins by collecting a large number of facial images in a database. The system then creates a set of eigenfaces by combining all of the facial images in the database and comparing commonalities and differences between groups of individual facial images (Velasco, 1998). The eigenfaces developed by the system appear as two-dimensional sets of light and dark areas arranged in a particular pattern. When a face is presented to the eigenface system for identification, the locations of the eyes are found first. The eye location provides a reference point so the head can be located and scaled to a standard size. Next, the system concentrates on the face only, leaving out clothing and hair, and removing brightness and contrast variations caused by the cameras settings (Lau Technologies, 1999). Then the program compares the live face's eigenface characteristics with those in the database and determines a "degree of fit" score, between -1.0 and +1.0, for the target face. If the target face produces a high enough degree of fit score when compared to a face stored in the database, it is recognized and accepted by the

system (Ruggles, 1998). Practically any individual can be identified using a database of 100 to 150 eigenfaces (Velasco, 1998).

A variation of the eigenface approach, called eigenfeatures, is also being developed. The eigenfeature approach combines facial metrics, which involves measuring the distance between specific facial features, such as the eyes, nose, and mouth, with the eigenface approach. The eigenfeatures system measures the distance between these points on a live face and compares them to the sets of eigenfeatures stored in the database to determine whether the face is a match (Randall, 1999). **[6]**

3.3.2 Fisherface

The Fisherface method was developed by Peter N.Belheumeuer, Joao P. Hespanha and David J. Kriegman of Yale University in 1997. It uses similar approached as Eigenface; both create a feature space to project images for comparison. Eigenface uses Principal Component Analysis (PCA) for better compression of the images while the Fisherface use Fisher's Linear Discriminant (FLD) for better representation of images in terms of classification.

Fisherface method is able to take advantage of within class information, minimizing variation with each class, yet still maximizing class separation. To accomplish, the training set were expanded to contain multiple images of each person, providing examples of how a person face may changes from one image to another due to variations in lighting conditions, facial expression and even small change in orientation. **[7]**

3.3.3 Hidden Markov model

Hidden Markov Models were first described in a series of statistical papers by Leonard E. Baum and other authors in the second half of the 1960s. One of the first applications of HMMs was speech recognition, starting in the mid-1970s.
In the second half of the 1980s, HMMs began to be applied to the analysis of biological sequences, in particular DNA. Since then, they have become ubiquitous in the field of bioinformatics. **[8]**

CHAPTER 4 IMPLEMENTATION GUI FOR FACE RECOGNITION SYSTEM

4.1 Graphical User Interface

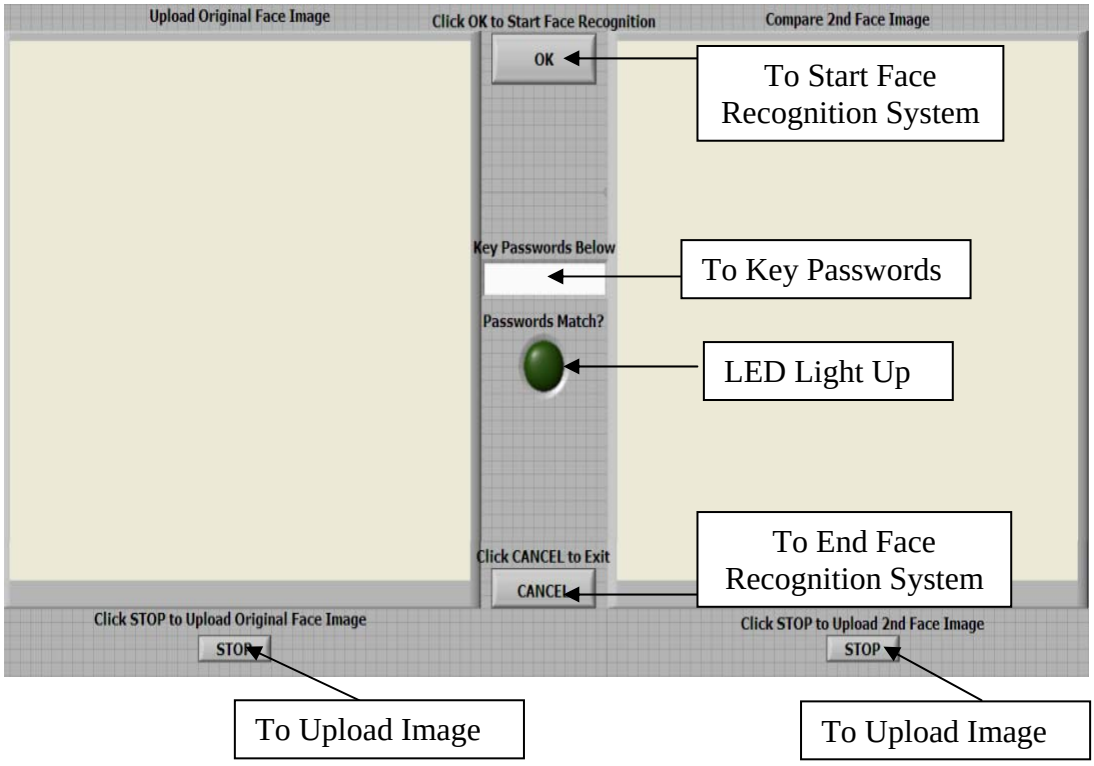


Figure 2: GUI For Face Recognition System

In this Face Recognition System, we need a GUI interface whereby the user can easily click to start the face recognition system and this is GUI interface is created directly using LABVIEW software application

4.2 How this GUI system works

In this GUI System, it had been programmed and designed that it is an user friendly for anybody to use the Face Recognition System without any difficulty. The steps sequences are as follows:-

- **Step 1**: The user will be directed to click “OK” to start the Face Recognition System
- **Step 2**: A small dialog box (will be discussed later in Chapter 6) will pop out to ask the user to position his/her face image in a central position.
- **Step3**: HP Web camera had also been activated (will be discussed later in Chapter 6) to project out the user’s face image to be captured into the screen **Step 4**: Once he/she had positioned ready, there is a button is waiting to be instructed by the user to click “STOP”
- **Step 5**: The whole process for the user’s original face image had been downloaded and uploaded into the Face Recognition System.
- **Step 6**: Now the user will be directed again to position his/her face image in a central position for the second phase and this time his/her face image will be projected out on the next screen just next to the 1st screen.
- **Step 7**: And again, there is also another button for the user to click “STOP” once he/she is ready for uploading the 2nd face image.
- **Step 8**: Once the Face Recognition System had been uploaded with two face images, they will go through a digital image processing called IMAQ Histogram VI. A dialog box showed that “Face Recognition Passed! Please Key Passwords and Press ENTER to proceed” where it will go to the next step or otherwise, it will pop out another kind of dialog box says “Face Recognition Failed! Please try again” where it will go back to Step 1.
- **Step 9:** The purpose for the entry of passwords is the next security level in the face recognition system before it really complete the whole process and not to mention a Green LED will light up if the entry passwords match with the loaded passwords.

CHAPTER 5 OVERVIEWS OF HARDWARE

Initially there were plan to work on the prototype of a Face Recognition System and a password device to be implanted into a hardware used for every household premises. After discussed with my tutor and not forgetting the main objective for this project is to design a Face Recognition System so as to be used in the Home Security System. Not only that, involving cost for the face recognition system also a constraint.

Therefore hardware used in this project are as follows:

- 1 x HP Notebook PC
  HP Genuine Intel(R) CPU T2050, 1.60GHz, 504MB of RAM
- 1 X HP Pavilion Web Camera 1000 (Default setting)

Video Pro Amp

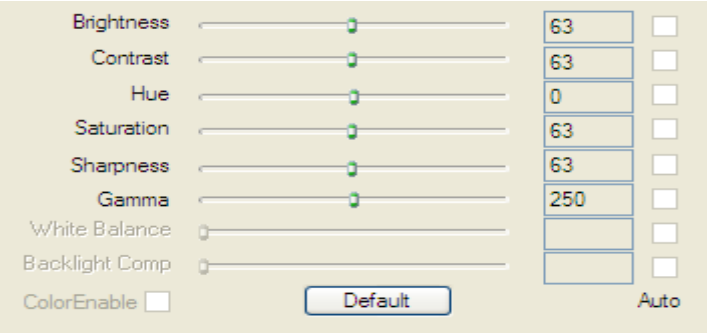


Table 2

Advanced Setting

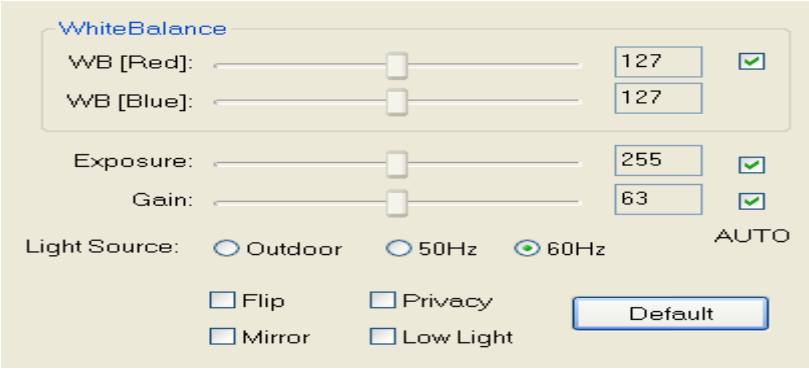


Table 3

CHAPTER 6 OVERVIEWS OF LABVIEW SOFTWARE APPLICATION

6.1 Using NI-IMAQ-USB camera to capture face image [9]

In this project HP Pavilion Web camera 1000 is used to capture face images and this is because webcam is the most cheapest and due to it availability in common Notebook PC, it is always ready for usage.

Web camera using LABVIEW software application will be explained out in details. But first we shall look at the block unit used for creating and activating Web Camera to capture an image.

6.1.1 IMAQ Create

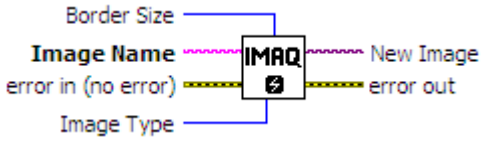


Figure 3 IMAQ Create

IMAQ Create is used to create a image.

Only two terminal lines were used for this configuration and there are as follows:-

- **Image Name** is the name associated with the created image. Each image created must have a unique name.
- **New Image** is the **Image** reference that is supplied as input to all subsequent (downstream) functions used by NI Vision. Multiple images can be created in a LabVIEW application.

6.1.2 IMAQ USB Grab Setup.vi

Figure 4 IMAQ USB Grab Setup.vi

IMAQ USB Grab Setup.vi is used to start a continuous acquisition. Once the acquisition has started, call IMAQ USB Grab Acquire to copy images from the continuous acquisition. Only one camera can be acquire at a time

6.1.3 IMAQ USB Grab Acquire

Figure 5 IMAQ USB Grab Acquire

IMAQ USB Grab Acquire is used to acquire an image during a continuous acquisition. Call IMAQ USB Grab Setup to start the continuous acquisition. Only one camera can acquire at a time. Use the New Image output from the IMAQ Create.vi for the Image in

6.1.4 IMAQ USB Close.vi

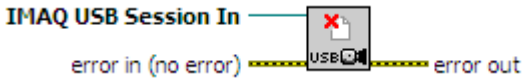


Figure 6 IMAQ USB Close.vi

IMAQ USB Close.vi is used to close a session to a USB camera that was opened with IMAQ USB init

6.1.5 While Loop [10]

Figure 7 While Loop

While Loop is used to repeat the sub diagram inside it until the conditional terminal, an input terminal, receives a particular Boolean value. The Boolean value depends on the continuation behavior of the While Loop. Right-click the conditional terminal and select **Stop if True** or **Continue if True** from the shortcut menu. We can also wire an error cluster to the conditional terminal, right-click the terminal, and select **Stop on Error** or **Continue while Error** from the shortcut menu. The While Loop always executes at least once. The iteration **(i)** terminal provides the current loop iteration count, which is zero for the first iteration.

We select a While Loop on the Execution Control Express VIs and Structures palette and place it on the block diagram, a stop button also appears on the block diagram and is wired to the conditional terminal. Therefore with the addition of a loop structure, we can continuously 'grab' images and process them, or display the camera output continuously.

6.1.6 Block Diagram

And so we shall concluded and combined as one with all the block units explained above to become a system shown below to capture images using Web Camera.

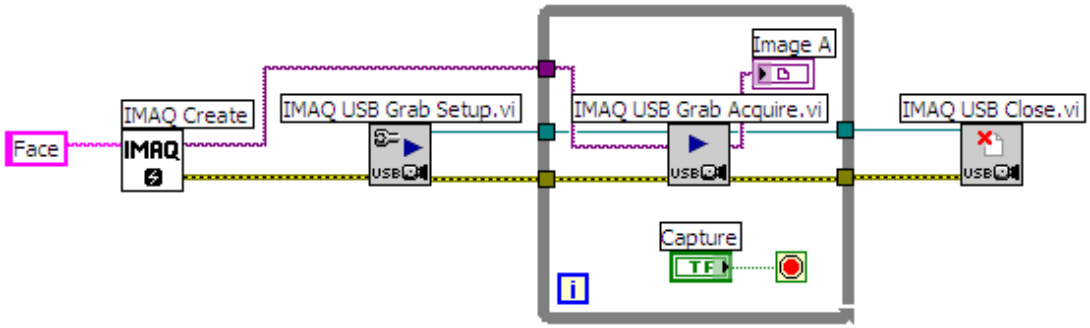


Figure 8 Block Diagram of capturing face images using USB Web Camera

A *grab* is a continuous, high-speed acquisition of data to a single buffer in host memory. This function performs an acquisition that loops continually on one buffer. We can get a copy of the acquisition buffer by grabbing a copy to a LabVIEW image buffer. We must use two VIs—IMAQ USB Grab Setup and IMAQ USB Grab Acquire—for a grab acquisition in LabVIEW. Call IMAQ USB Grab Setup once to initialize the acquisition and start capturing the image to an internal software buffer. We can call IMAQ USB Grab Acquire multiple times to copy the image currently stored in the internal buffer to a LabVIEW image buffer. After the program finishes copying images, call IMAQ USB Close once to shut down the acquisition. Figure 8 shows a simplified block diagram for using IMAQ USB Grab Setup and IMAQ USB Grab Acquire.

6.1.7 Stacked Sequence Structure

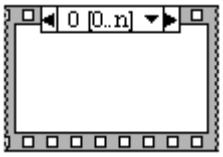


Figure 9 Stacked Sequence Structure

Stacked Sequence Structure is used to consist of one or more subdiagrams, or frames, that execute sequentially. Right-click the structure border to add and delete frames or to create sequence locals to pass data between frames. Use the Stacked Sequence structure to ensure a subdiagram executes before or after another subdiagram.

To scroll through the available subdiagrams, click the decrement and increment arrows in the selector label. You can add, duplicate, rearrange, or delete the subdiagrams. You can use a sequence local terminal to pass data from one frame to any subsequent frame.

The Stacked Sequence structure does not start to execute until all data wired to the structure arrive. The data wired from each frame leave only when all the frames complete execution.

6.1.8 Case Structure

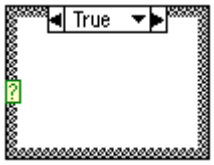


Figure 10 Case Structure

Case Structure is used because it has one or more subdiagrams, or cases, exactly one of which executes when the structure executes. The value wired to the selector terminal determines which case to execute and can be Boolean, string, integer, or enumerated type. Right-click the structure border to add or delete cases. Use the Labeling tool to enter value(s) in the case selector label and configure the value(s) handled by each case. To scroll through the available subdiagrams, click the decrement and increment arrows in the selector label.

After we create a Case structure, we can add, duplicate, rearrange, or delete the subdiagrams. For each case, use the Labeling tool to enter a single value or lists and ranges of values in the case selector label at the top of the Case structure.
We also can create multiple input and output tunnels and specify a default case for a Case structure.

Those two structures that mentioned above will be used later as part of the Face Recognition System.

6.2 Using Histogram Technique and calculated array to compare two face images

After capturing two face images Using NI-IMAQ-USB camera techniques, they will go through a process called Histogram technique to calculate the array between those two face images and of course a few selected block units were used to configure this calculated array process and there are as follows.

6.2.1 IMAQ Create

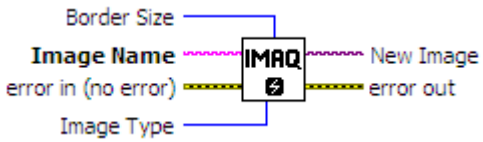


Figure 11 IMAQ Crea*te*

IMAQ Create is used to create a image.

Only two terminal lines were used for this configuration and there are as follows:-

- **Image Name** is the name associated with the created image. Each image created must have a unique name.
- **New Image** is the **Image** reference that is supplied as input to all subsequent (downstream) functions used by NI Vision. Multiple images can be created in a LabVIEW application.

6.2.2 IMAQ ExtractColorPlanes

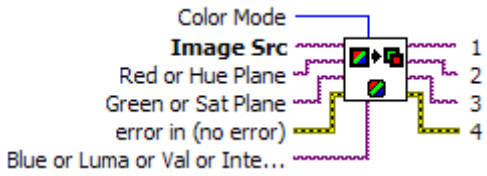


Figure 12 IMAQ ExtractColorPlanes

IMAQ ExtractColorPlanes is used to extract the three planes (RGB, HSL, HSV, or HSI) from an image.

Only seven terminal lines were used for this configuration and there are as follows:-

- **Image Src** is a reference to the source image.
- **Red or Hue Plane** is the reference to the destination image for the first color plane. This plane can be either the red plane (**Color Mode** RGB) or the hue plane (**Color Mode** HSL, HSV, or HSI). If **Red or Hue Plane** is connected, it must have as many bits per pixel as the extracted color plane. If **Red or Hue Plane** is not connected, the VI does not extract the first color plane.
- **Green or Sat Plane** is the reference to the destination image for the second color plane. This plane can be either the green plane (**Color Mode** RGB) or the saturation plane (**Color Mode** HSL, HSV, or HSI). If **Green or Saturation Plane** is connected, it must have as many bits per pixel as the extracted color plane. If **Green or Saturation Plane** is not connected, the VI does not extract the second color plane.
- **Blue or Luma or Val or Inten Plane** is the reference to the destination image for the third color plane. This plane can be either the blue plane (**Color Mode** RGB), the luminance plane (**Color Mode** HSL), the value plane (**Color Mode** HSV) or the intensity plane (**Color Mode** HSI). If **Blue or Luma or Val or Inten Plane** is connected, it must have as many bits per pixel as the extracted color plane. If **Blue or Luma or Val or Inten Plane** is not connected, the VI does not extract the second color plane.

- **Red or Hue Plane out** is the reference to the image containing the red (or hue) plane of the source (input) image.
- **Green or Sat Plane out** is the reference to the image containing the green (or saturation) plane of the source (input) image.
- **Blue or Luma or Val or Inten Plane out** is the reference to the image containing the blue (or luminance, value, or intensity) plane of the source (input) image.

6.2.3 IMAQ Add

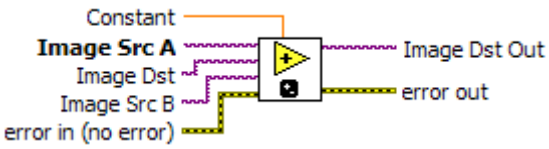


Figure 13 IMAQ Add

IMAQ Add is used to add two images or an image and a constant.
An operation between an image and a constant occurs when **Image Src B** is not connected. The two possibilities are distinguished in the following equations:
Dst(x, y) = SrcA(x, y) + SrcB(x, y) or *Dst(x, y) = SrcA(x, y) + Constant*
The different image-type combinations supported by this VI are described in the following equations. The first symbol represents the image connected to **Image Src A**, and the second symbol represents the image type connected to **Image Src B**. The third symbol represents the image type that should be connected to **Image Dst**.
To add a constant to an image, **Image Dst** must be connected to the same image type as **Image Src A**.

When an 8-bit image or constant is added to an RGB image, it is added to each plane of the RGB image. When IMAQ Add is performed on two RGB images, each color plane (red, green, and blue) of **Image Src A** is added to the corresponding color plane of **Image Src B**. Each of these additions is similar to an IMAQ Add operation on two 8-bit images.

Only three terminal lines were used for this configuration and there are as follows:-

- **Image Src A** is a reference to the first source image.
- **Image Src B** is the reference to the second source image.
- **Image Dst Out** is the reference to the destination (output) image that receives the processing results of the VI. If the **Image Dst** is connected, **Image Dst Out** is the same as **Image Dst**. Otherwise, **Image Dst Out** refers
- to the image referenced by **Image Src A**.

6.2.4 IMAQ Histogram

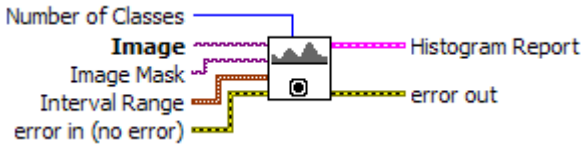


Figure 14 IMAQ Histogram

IMAQ Histogram is used to calculate the histogram of an image.

The IMAQ Histogram VI outputs the histogram for the image we specify. The actual histogram array consists of N elements (or classes). This means if we specified 256 for the "Number of classes" input, our histogram array will have 256 elements. Each element defines a specific pixel intensity range. If the intensity of a pixel in the image falls under this particular range, then this element's value will increase by 1. In the same manner, all the pixels can be grouped into the 256 elements according to their intensities. Thus, each element is a number, equal to a certain number of pixels. Adding up the value of each element will give you the total number of pixels in the image.

Only two terminal lines were used for this configuration and there are as follows:-

- **Image** is a reference to the source image.
- **Histogram Report** is a cluster that returns the histogram values. This cluster contains the following elements:

- **histogram** returns the histogram values in an array. The elements found in this array are the number of pixels per class. The *n*th class contains all pixel values belonging to the interval [(*Starting Value* + (*n* – 1) × *Interval Width*), (*Starting Value* + *n* × (*Interval Width* – 1))].
- **Minimal Value** returns the smallest pixel value used in calculating the histogram.
- **Maximal Value** returns the largest pixel value used in calculating the histogram.
- **Starting Value** returns the smallest pixel value from the first class calculated in the histogram. It can be equal to the **Minimum** value from the **Interval Range** or the smallest value found for the image type connected.
- **Interval Width** returns the length of each class.
- **Mean Value** returns the mean value of the pixels used in calculating the histogram.
- **Standard Variation** returns the standard deviation from the histogram. A higher value corresponds to a better distribution of the values in the histogram and the image.
- **Area (pixels)** returns the number of pixels used in the histogram calculation. This is influenced by the values specified in **Interval Range** and the contents of **Image Mask**.

6.2.5 Unbundle By Name

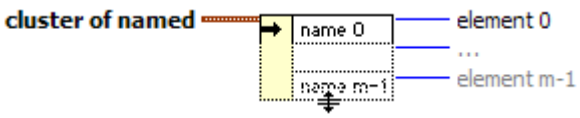


Figure 15 Unbundle By Name

Unbundle By Name is used to return the cluster elements whose names we specify. We do not have to keep track of the order of the elements within the cluster. This function does not require the number of elements to match the number in the cluster. After we wire a cluster to this function, We can select an individual element from the function.

Only two terminal lines were used for this configuration and there are as follows:-

- **cluster of named** is the cluster whose elements We want to access.
- **element 0..m-1** is the element of **cluster of named** called name 0..m-1.We can access only elements with owned labels. Select a named element by clicking a name terminal and selecting a name from the shortcut menu.

6.2.6 Delete From Array

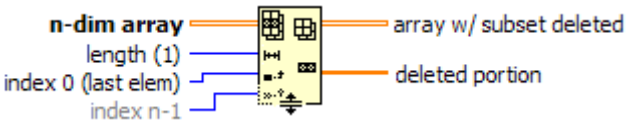


Figure 16 Delete From Array

Delete From Array is used to delete an element or subarray from **n-dim array** and returns the edited array in **array w/ subset deleted** and the deleted element or subarray in **deleted portion**. When we wire an array to this function, the function resizes automatically to display **index** inputs for each dimension in the array.

Delete From Array Details

This function reduces the array in one dimension only, therefore, we can wire only one **index** input. For example, to delete a row in a 2D array, wire only the **row** index. To delete a column, wire only the **column** index. We can delete multiple consecutive subarrays at once by wiring **length**.

Only two terminal lines were used for this configuration and there are as follows:-

- **n-dim array** is the array from which we want to delete element(s), row(s), column(s), page(s), and so on. This input can be an n-dimension array of any type.
- **array w/ subset deleted** is the array returned with the deleted element(s), row(s), column(s), or page(s).

6.2.7 Add Array Elements

Figure 17 Add Array Elements

Add Array Elements is used to return the sum of all the elements in **numeric array**. The connector pane displays the default data types for this polymorphic function.

Only two terminal lines were used for this configuration and there are as follows:-

- **numeric array** can have any number of dimensions.
- **sum** is of the same data type and structure as the elements in **numeric array**.

6.2.8 Block Diagram

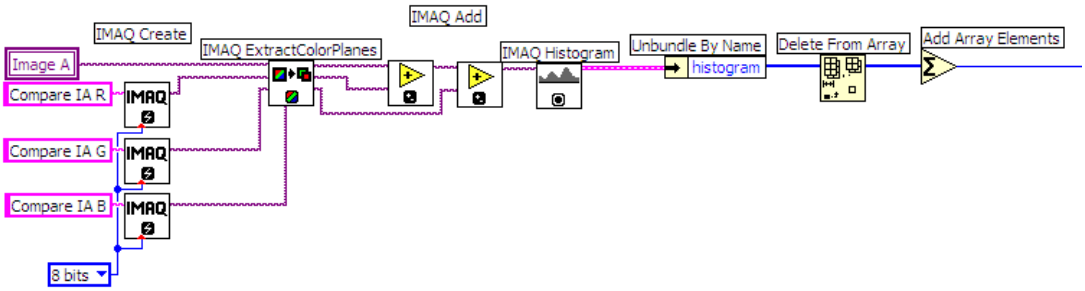


Figure 18 Block Diagram of the Histogram Techniques

There will be two block diagram of Histogram Techniques needed for two face images that were captured by the Web Camera.

6.3 Passwords Authentication Entry for final phase of the Face Recognition System

Upon succeeding the face images matching through the calculated array, it will direct the user to the Passwords Authentication Entry whereby the user will key in the secret passwords so as to match the passwords data that was already loaded into the Face Recognition system. Some of the block units were chosen for this configuration and they are:-

6.3.1 Property Mode [11]

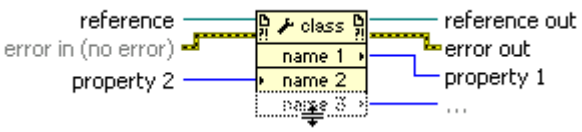


Figure 19 Property Mode

Property Mode Gets (reads) and/or sets (writes) properties of a reference

Two Property Modes were used to initialize the inputs for the user to key in. The 1st property mode is referring to the Round LED named “Password Matched?” and in order to initialize the Button to OFF mode during every start of the Face Recognition. A False Constant was inserted at the beginning of the 1st property 2.

Another 2nd Property Mode was also used and it is referring to the String Control named “Password” and in order to initialize the string to clear mode during every start of the Face Recognition. An Empty String Constant was inserted at the beginning of the 2nd property 2 to clear away whatever inputs that was previously keyed by the user.

6.3.2 Equal gate

Figure 20 EQUAL gate

OR Returns TRUE if **x** is equal to **y**. Otherwise, this function returns FALSE.

Only three terminal lines were used for this configuration and there are as follows:-

- **X** was connected to String Control named “Password” waiting for the user to key the secret password
- **Y** was connected String Constant named “String Constant” whereby the password data was loaded as “Face Recognition”
- **Output** indicated the result whether x = y? which mean whether the passwords keyed by the user tally with the passwords that was already uploaded in the system.

6.3.3 OR gate

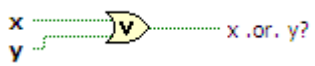


Figure 21 OR gate

OR computes the logical OR of the inputs. Both inputs must be Boolean or numeric values. If both inputs are FALSE, the function returns FALSE. Otherwise, it returns TRUE.

Only three terminal lines were used for this configuration and there are as follows:-

- **X** was connected to output of the OR gate waiting for the user to complete the whole Face Recognition System.
- **Y** was connected to Cancel button 

named “Cancel” waiting for the user to cancel/restart the whole Face Recognition System.
- **Output** indicated the result whether x .or. y? which mean either the user keyed the password correctly or pressed the Cancel button 

, it will exit out the Passwords Authentication Entry and directed back to the Face Recognition System.

6.3.4 Block Diagram

The block diagram below show the full Password Authentication Entry after the two face images founded to be matched.

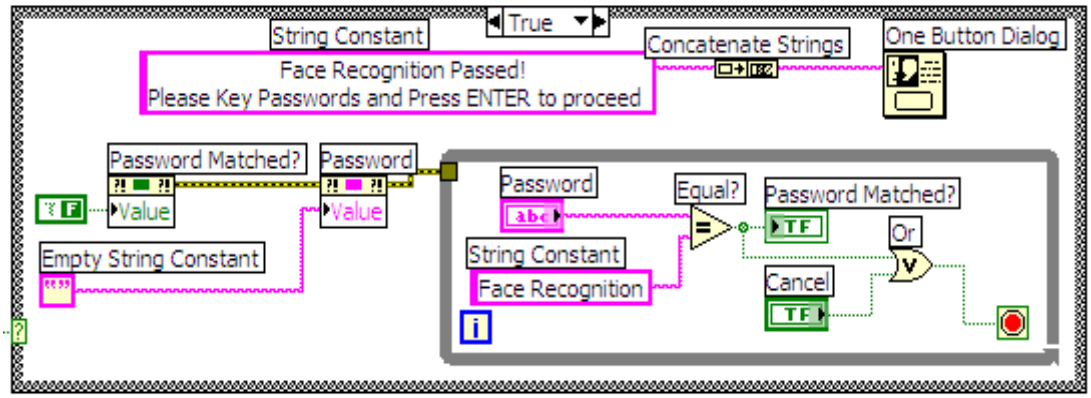


Figure 22 Block Diagram of the Passwords Authentication Entry

6.4 Useful Words or Messages to the user

In the Face Recognition System, it is user friendly as it will pop out dialog and direct the user instructions and ways to complete the whole Face Recognition System. The block units are as follows:

6.4.1 String Constant

Figure 23 String Constant

In this String Constant, it can have any characters or numbers so that it can clearly show what it wants from the user.

6.4.2 Concatenate Strings

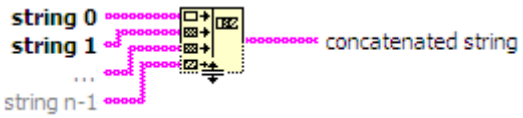


Figure 24 Concatenate Strings

Concatenates input strings and 1D arrays of strings into a single output string. For array inputs, this function concatenates each element of the array

Only two terminal lines were used for this configuration and there are as follows:-

- **string 0..n-1** are the strings you want to concatenate.
- **concatenated string** contains the concatenated input strings in the order you wire them to the node from top to bottom.

6.4.3 One Button Dialog

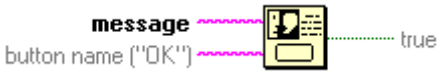


Figure 25 One Button Dialog

Displays a dialog box that contains a message and a single button.

Only two terminal lines were used for this configuration and there are as follows:-

- **message** is the text to display in the dialog box.
- **true** contains a value of TRUE when you click the button.

6.4.4 Block diagram

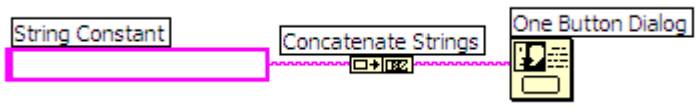


Figure 26 Block Diagram for Dialog Box

CHAPTER 7 CIRCUIT FLOW OF THE FACE RECOGNITION SYSTEM

With all block diagrams that explained, we shall combine all together to become a full Face Recognition System. But we will need them and placed in a orderly manner/sequence. And shown below are the step sequences of the circuit flow for whole Face Recognition System and they are explained in details as follows:-

Step 1: A Stacked Sequence Structure was used and the 1st Web Camera will be activated to upload original face image.

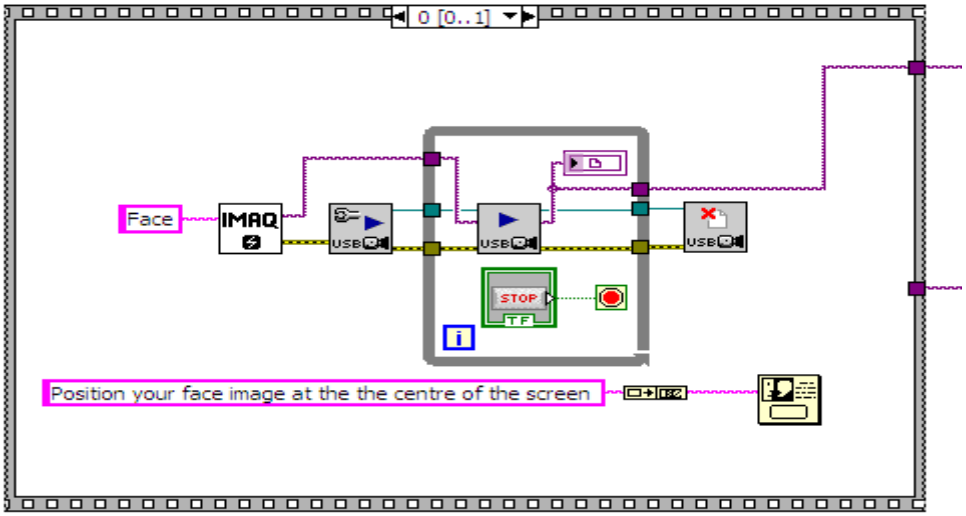


Figure 27: Block Diagram of the 1st Web Camera activation

Step 2: Another round Stacked Sequence Structure was used and this time it activated the 2nd Web Camera to upload another face image.

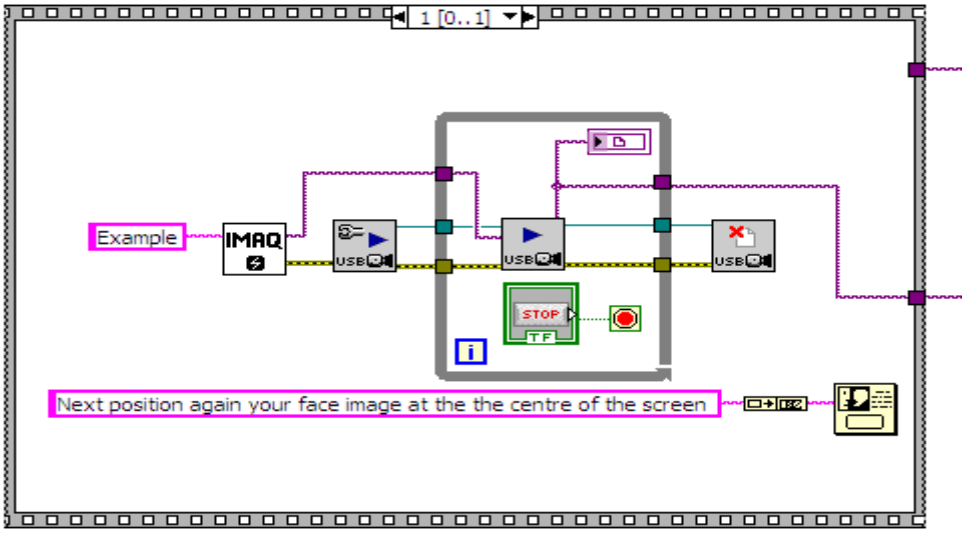


Figure 28: Block Diagram of the 2nd Web Camera activation

Step 3: A Stacked Sequence Structure was used also and after uploading those two face images, both of them will gone through a calculated array shown below and then compare the difference between them before it goes to the final phase which I mentioned earlier the password authentication entry.

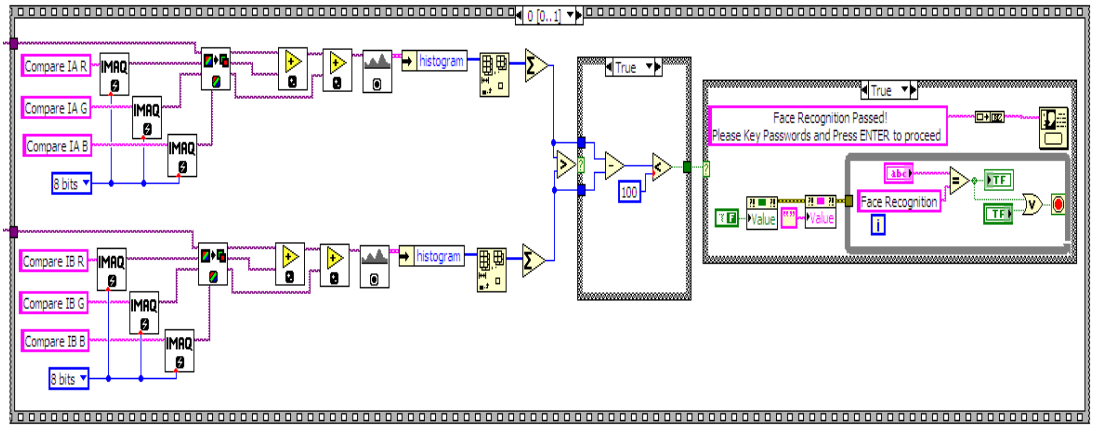


Figure 29: Block Diagram of two face images for processing

Step4: This block diagram shown below the detail process of the two face images undergo calculated array.

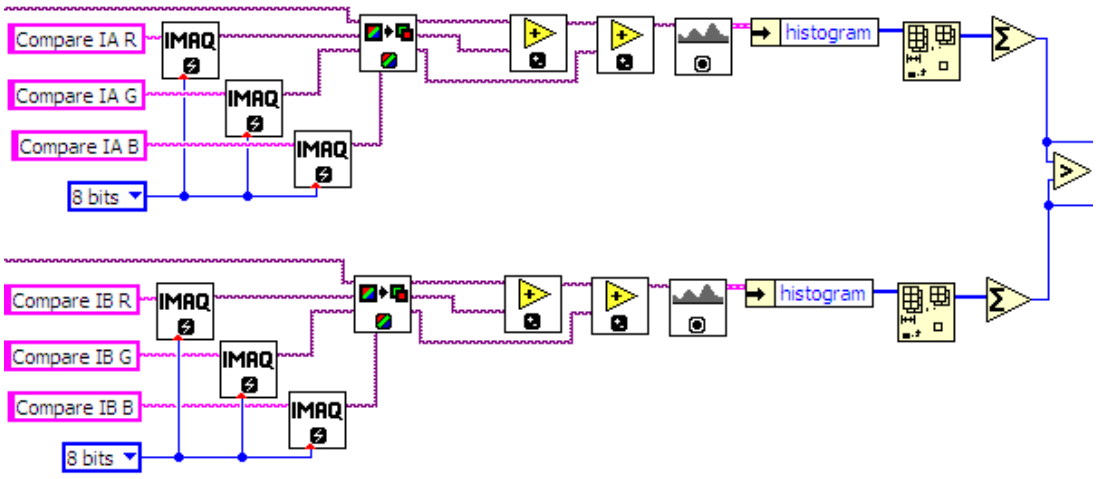


Figure 30: Block Diagram of the calculated array

Step5: This time round a Case Structure was used instead. This block diagram shown below the circuit process of the two face images undergo calculated the array variance.

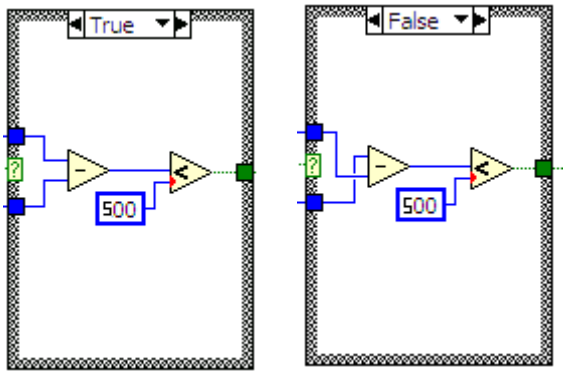


Figure 31: Block Diagram of the calculating the array variance between two face images

Step6: A Case Structure was also used. This block diagram shown below the circuit process for passwords authentication entry once the two face images' array variance difference is less than 500.

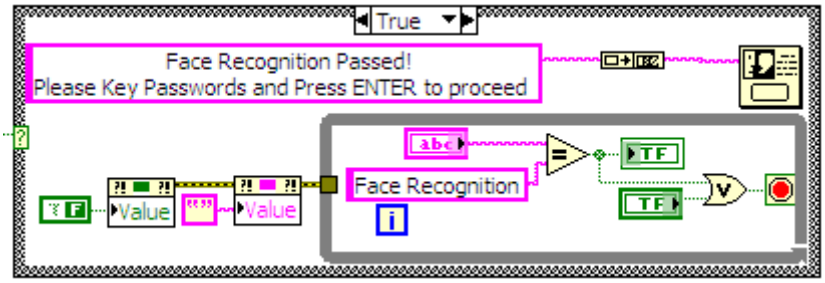


Figure 32: Block Diagram of the Passwords Authentication Entry

Step7: Another Case Structure was used. This block diagram shown below the circuit process of popping out a dialog box once the Face Recognition System failed.

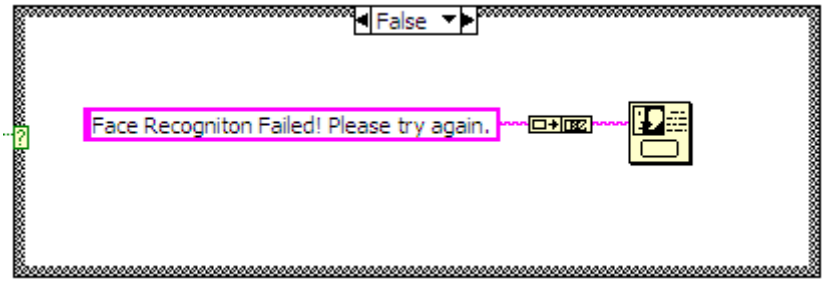


Figure 33: Block Diagram of the showing a dialog box

Step7: This block diagram shown below indicated the action of clearing two face images once the face recognition had either completed or failed.

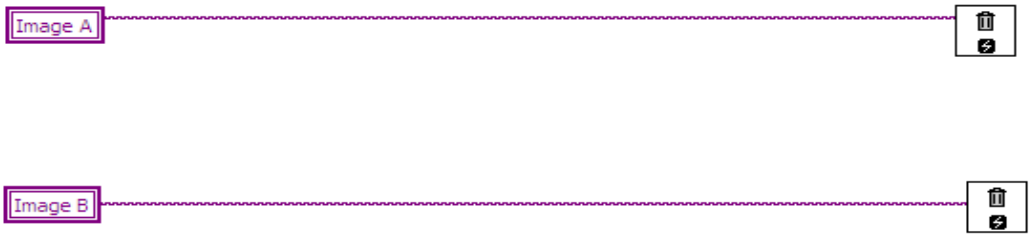


Figure 34: Block Diagram of the clearing two face images

CHAPTER 8 TEST RESULTS

The Face Recognition system was designed with my own personal HP laptop together with an installed USB Web Camera whereby it is used and based on 2D face colour images to be captured for face recognition. Hence, the face image is captured with a black background behind so as to maximize the face image for data analyzing using LABVIEW application. Examples are shown below

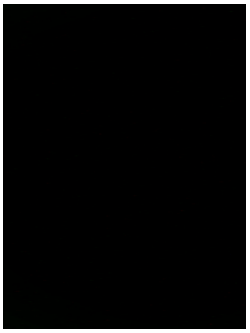


Figure 35: Black background with no face image

Figure 36: Black background with face image

There will be a few conducts of test experiments running the Face Recognition System and they will be shown and explained in each section as follows:

- Missing features on face images
- Different expressions on face images
- Different degree postures on face images
- Different turning postures on face images
- Different distances on face images

8.1 Missing features on face images

Objective: To test the system accuracy to react accordingly for missing features from the face images.

Original Face Image

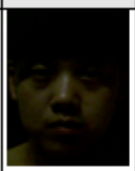

Feature	Face Image1	Face Image2	Face Image3	Face Image4	Face Image5	Face Image6	Face Image7
Image							
Result	Passed	Failed	Passed	Passed	Failed	Failed	Failed

Figure 37: Face Images features

8.1.1 Discussion

There are a total of 7 face images that had been processed through LABVIEW to compare with the original face image. As we can see Face Images 2, 5, 6 and 7 obtained a “Failed” whereas the rest of the Face Images obtained a “Passed.” The reason is quite simple as this is regarding the process of the LABVIEW software application which is comparing calculated arrays between two face images. Once that calculated array from the particular face image that was being uploaded to compare is getting lesser than what was expected than the original and definitely, it will obtained a “Failed” result.

8.2 Different expressions on face images

Objective: To test the system accuracy to react accordingly for different expression from the face images.

Original Face Image


Expression	Face Image1	Face Image2	Face Image3	Face Image4	Face Image5	Face Image6	Face Image7
Image							
Result	Passed	Passed	Passed	Passed	Passed	Passed	Passed

Figure 38: Face Images expressions

8.2.1 Discussion

There are a total of 7 face images that had been processed through LABVIEW to compare with the original face image. And we see all the Face Images obtained a “Passed.” This is because there was not much change in the face images that were uploaded into the process of the LABVIEW software application. Same theory applied, once that calculated array from the particular face image that was being uploaded to compare with the original face image is getting a little more than what was expected than the original and definitely, it will obtained a “Passed” result.

8.3 Different degree postures on face images

Objective: To test the system accuracy to react accordingly for different degree posture features from the face images (Up, down, diagonal right, diagonal left, off left & off right).

Posture	Face Image1	Face Image2	Face Image3	Face Image4	Face Image5	Face Image6
Image						
Result	Pass	Pass	Failed	Pass	Pass	Failed

Figure 39: Face Images degree postures

8.3.1 Discussion

There are a total of 6 face images that had been processed through LABVIEW to compare with the original face image. As we can see Face Images 3 and 6 obtained a "Failed" whereas the rest of the Face Images obtained a "Passed." Take note there were not much missing features or expression changes. These experiments are more towards the posture of the face image that was going to be positioned. Lighting do played an important part in this Face Recognition System because some of the areas might be dark whereas some of the areas might be bright.

8.4 Different turning postures on face images

Objective: To test the system accuracy to react accordingly for different turning postures from the face images. (left to right and right to left)**.**

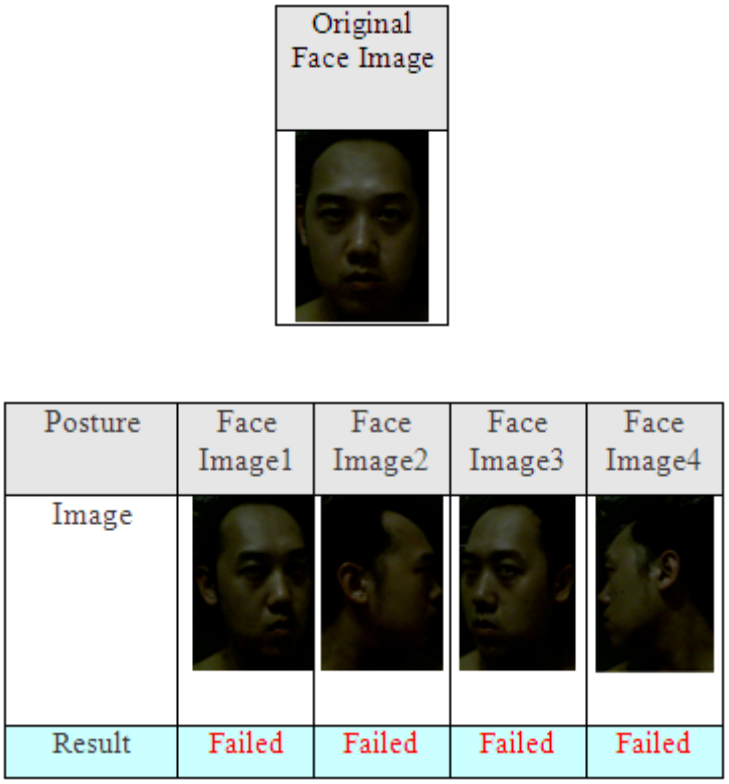


Figure 40: Face Images turning postures

8.4.1 Discussion

All the face images failed and it is probably due to more or less calculated array after comparing the difference between the original face image just like the missing features and lighting that might affected the face recognition system.

8.5 Different distances on face images

Objective: To test the system accuracy to react accordingly for different distances from the face images (from camera to face).

Distance	Face Image1	Face Image2	Face Image3	Face Image4
Image				
Result	Failed	Passed	Passed	Failed

Figure 41: Face Images distances

8.5.1 Discussion

All face images obtained a “Passed” except for face image 1 and 4. Smaller face image resulted much more smaller calculated array after compared with the original face image. And truly, bigger face image should result much greater calculated array and most probably it will give a “Failed” outcome. That why in this face recognition system, LABVIEW was programmed to enable accepted a “Passed” outcome if the calculated array variation is still within the range of 500.

CHAPTER 9 CONCLUSION

This project set out to implement a Face Recognition System to be used widely in every household premise. The main objective for this project to develop a Face Recognition System and while developing this Face Recognition System, they will be problems surfaced and need to be solved in order to proceed furthermore. It is not really a full proof Face Recognition System and this is because it is very vulnerable to lighting and posture variation.

Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a powerful and flexible instrumentation and analysis software system [12, 13, 14]. It uses a graphical programming environment that uses the construction of virtual instruments as a metaphor [15]. The programming aspect of LabVIEW is similar to a creating flowcharts where icons are wired together to perform various functions.

Using LABVIEW software application is a challenging task for me and this is a new software being introduced to me after attending one of the Capstone Workshop 3. Basically this software application is good at vision teaching that why I chosen LABVIEW for my Face Recognition System. There are so many way of using vision teaching in LABVIEW and each has it own advantages. And it took me quite a time before I really decide what kind of vision teaching and how to use it correctly before it can really work fine on the Face Recognition System.

The initial phase was a touch one too because I don’t have any knowledge on the Face Recognition System. And the reason I took this as my final year project is simply that it will be an interesting research and topic to work on it and even though there are obstacles while working on this project. Truly I felt that it is still worthwhile as it was fun and has provided a great learning experience.

CHAPTER 10 FUTURE WORKS

The Face Recognition System that I am working on which I feel it can be further improved. They are as follows:-

- If only the program can automatic detect only one of the features (face, mouth, nose or eyes) from the original face image and compare it with the uploaded face image rather than taking reference from the whole features original face image and compare with the uploaded face image.
- Because the program is unable to locate only the face image (head) for vision teaching that why it is used in a black background so as it can able to concentrate more on the face image (head).
- At this time, the program is only able to do face recognition with one-to-one person. It will be useful that it can able to can able to upload more original face images and compare with many uploaded face images for face recognition.
- Instead of using Web Camera for the capturing of face image, a CDD camera will definitely work well in Face Recognition System

The whole ideas in this Face Recognition System is not new in the society
There are many kinds of Face Recognition System available out there that had been invented and used by many researchers. Each has it own way of doing the recognition process and also achieved very good results.

So I am not surprised if the Face Recognition System become a commercialize thing to be used in every part of the world.

CHAPTER 11 CRITICAL REVIEW AND REFLECTIONS

11.1 Summary of the Project Development

From the start till the end of this project, there are bound to have difficulties in achieving the project to be a complete Face Recognition System.

In the PPA form stated, the plan was to implement and build a hardware/prototype of this system to be processed into a Face Recognition Home Security System. A password entry device was also included to increase the security level for the project. After discussion with my tutor and due to cost constraint, furthermore the main objective is to implement a Face Recognition System. Therefore no hardware/prototype was involved in this project.

Initially in project plan, the installation for a CDD camera was also needed to be carried and implanted onto the hardware device so as to capture face images for recognition purpose. Therefore resulting, my personal PC notebook to implement the system for the Face Recognition and my integrated Web Camera to capture face images were mainly involved for this project.

The criteria and targets set in the first TMA01 were as follow:-

- To list out the various different type of Face Recognition Algorithms to be used in the project.
- To list out the various different type of Software Application to be used in the project.
- To list out the various CDD cameras in the available markets to be used in the project.
- To list out hardware available for Face Recognition System and Password Entry Hardware device.

There are so many types of Face Recognition Algorithms available. There is the EigenFaces method, Fisherface method, Hidden Markov model and so on. And to study each of them in details were the tedious parts because it has it owns way of detecting face images for recognition and advantages too. Understanding the concepts of the algorithms takes some time but it is the implementation of the algorithms into the selected software application to be programmed in it takes most of my time for this project.

Furthermore it is the toughest and critical part for this project in which the success level will depend on it. Not to mention, I am also not very well versed in programming.

All the targets mentioned above still met except the last two pointers were not really carried out due to costs constraints. Still I do have the Web camera to capture face images and using the LABVIEW Software Application to come out both the Face Recognition System and the Password Authentication Entry.

11.2 Skill Sets for the Project Development

For the completion and success level of this project, three main skills were required and they are as follows:-

- Implementation and Design of the Project
- Time Management
- Report Writing

There are so many researches about Face Recognition System in the internets and the implementation and the design for Face Recognition System need to be carried concurrently in order to become a successful one. I kept concentrating too much on the implementation and on the design. That why there is a need to have a fixed algorithm and chosen software application for the whole project development so that it will no turn out to be no ending project.

In order for the implementation and the design for the project to be carried out smoothly, time management need to play an important part too. That why, time was wasted and not to mention limited time is left for Report Writing.

Luckily for Report Writing portion, I did complete my HESZ331 Electronic Engineering Innovation and Design during my third year in UniSIM and truly I do benefit a lot from it when writing final report for this project.

11.3 Problems encountered and Problems solution

Problems encountered were involved in software only and solutions were also mentioned and there are as follows:-

Software

- LABVIEW Software Application is chosen for the Face Recognition System and downloading is required from the website and but to use it and purchase it is very costly luckily there is a trial period of 30 days period before it expired.

- Initially, pattern matching algorithms is selected to be implemented. But understanding from matching concepts used in LABVIEW, it is required that object is always at fixed location. That why Histogram Technique was chosen for this project because it calculate the overall array of the face image and I can compare the overall array with another face image and compare the difference easily.

- One Web camera is used to capture two face images for Histogram Techniques processing. A block diagram was to activate only one Web camera at one time but somehow even though with two different face images being uploaded, it always showed a matched result. Therefore, with the help of the LABVIEW manual downloaded, I found out that there is a need to indicate two different variable names to the two face images so that the system can able to recognize two different face images and can send for Histogram Techniques processing independently.

- The Histogram Techniques processing calculated the overall array of the face image but somehow it also calculated the background of face image behind and thus resulting incorrect matching face image. There is a need to make sure that the background is always constant so it will concentrate more on the face image rather than the background itself. Initially white background was used but resulting the calculated array tends to be inconsistent and in the end black colour is selected.

In conclusion, this project indeed is an interesting topic and challenging due to the LABVIEW Software Application is very new to me.

REFERENCES

1. http://www.epic.org/privacy/facerecognition/

2. http://vismod.media.mit.edu/tech-reports/TR-516/node7.html

3. http://www.biometrics.gov/Docs/facerec.pdf

4. http://www.cs.unm.edu/~sumanth/FinalProj.pdf

5. http://cnx.org/content/m12534/latest/

6. http://et.wcu.edu/aidc/BioWebPages/eigenfaces.htm

7. http://www.eng.newcastle.edu.au/~c3019989/Literature%20Review.html

8. http://en.wikipedia.org/wiki/Hidden_Markov_model#History

9. http://www.me.utexas.edu/~lotario/me244L/labs/basic_vision/ex2.html

10. http://www.uccs.edu/~aglushch/PES%20318%20Instrumentation%20Laboratory%20II_files/LabVIEW%20Basics%20I%20Course.PDF

11. http://www.uccs.edu/~aglushch/PES%20318%20Instrumentation%20Laboratory%20II_files/LabVIEW%20Basics%20II%20Course.PDF

12. LabVIEW 3.1, Charles seiter. MacWorld, Nov 1995

13. LabVIEW 4.0, Charles seiter. MacWorld, Aug 1996

14. National Instruments, http://www.ni.com

15. LabVIEW for everyone : graphical programming made even easier, Lisa Wells and Jeffrey Travis, Prentice Hall PTR, 1997