REAL TIME FACE RECOGNITION USING RASPBERRY PI FOR CLASS ATTENDANCE SYSTEM

HO HWA SHENG

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REAL TIME FACE RECOGNITION USING RASPBERRY PI FOR CLASS ATTENDANCE SYSTEM

HO HWA SHENG

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Engineering
Universiti Teknikal Malaysia Melaka

2020 / 2021

DECLARATION

I declare that this report entitled "Real Time Face Recognition Using Raspberry Pi For Class Attendance System" is the result of my own work except for quotes as cited in the references.

Signature :

Author: HO HWA SHENG

Date: 23/6/2021

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

Signature	:	
Supervisor Name	:	DR. NORIZAN MOHAMAD
Date	:	23 JUN, 2021

DEDICATION

I would like to dedicate this work to my project supervisor, family members and friends who willing to guide me and support me throughout this final year project.

ABSTRACT

Face recognition is one of the preferred biometric authentication that widely used in various applications nowadays. Some application such as in law enforcement systems, border control management systems, smartphone unlock systems and etc. One of the key advantage is provide faster verification and identification process that run in real-time. Therefore, can implement this technology to attendance system due to this advantage. First, it can reduce the time taken for attendance marking process when consider in the case of having large amount of attendant. Furthermore, eliminate the chances of any proxy attendance due to real-time processing. In this paper, the project use face recognition algorithm implemented in Raspberry Pi for class attendance system. Local binary patterns histograms (LBPH) is used as face recognizer in this study.

ABSTRAK

Teknologi pengecaman wajah adalah salah satu pengesahan biometrik pilihan yang digunakan secara meluas dalam pelbagai aplikasi pada masa kini. Beberapa aplikasi seperti dalam sistem penguatkuasaan undang-undang, sistem pengurusan kawalan sempadan, sistem buka kunci telefon pintar dan lain-lain.Salah satu kelebihan utamanya ialah proses pengesahan dan pengenalan yang lebih pantas serta berlangsung dalam masa nyata. Oleh itu, dapat menerapkan teknologi ini ke sistem kehadiran disebabkan kelebihan ini. Pertama, ia dapat mengurangkan masa yang diperlukan untuk proses menandakan kehadiran apabila mengambil kira terdapat jumlah atendan yang banyak. Di samping itu, mengurangkan kemungkinan adanya proksi kehadiran kerana berlangsung dalam masa nyata. Dalam kajian ini, projek ini menggunakan algoritma pengecaman wajah yang dilaksanakan di Raspberry Pi untuk sistem kehadiran kelas. Histogram corak binari tempatan (LBPH) digunakan sebagai pengecam wajah dalam kajian ini.

ACKNOWLEDGEMENTS

Firstly, I would like to express my gratitude to my project supervisor, Dr. Norizan Bin Mohamad for his invaluable advice and guidance throughout this project. I highly appreciate all his advice and valuable comments for this project.

Furthermore, I would like to express my gratitude to my family members for their unconditional support and encouragement throughout this project. Last but not least, I would like to thank my friends as the volunteer for testing in this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

For examples:

IoT	:	Internet of Things
OS	:	Operating system
QR	:	Quick Response
IMEI	:	International Mobile Equipment Identity
GPS	:	Global Positioning System
SMS	:	Short message service
HDMI	:	High-Definition Multimedia Interface
SD	:	Secure digital

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background of project

In general, attendance represent the marking record of someone attends or present at that particular time. Attendance is consider important because it can influence many factors mainly the performance. For instance, a company can review attendance rates to evaluate employee’s performance in the workplace. It also determines whether the characteristics such as responsibility, punctuality have on the employee.

Attendance is mandatory and necessary in almost every sector especially in educational institutions. Attendance is compulsory as students need to reach the certain percentage of attendance rate required to pass the subject and eligibility to sit for final exam. In addition, the academic performance of students have less or more are relate to student’s attendance. Students that actively attend the class show that attendance is related to learning outcomes[1]. Therefore, managing students’s attendance are vital task for administrators of educational institute to track and check student’s academic performance. In short, a good attendance system can help to track and manage attendance efficiency, besides make workload easy.

Regarding the types of attendance system, there are manual attendance system that still used nowadays. Typical method used in this system are educator calls student’s names one by one and when that student responds then it indicate he or she attend the class. Then, attendance will be mark on the attendance-list document. Another method is educator pass the attendance-list document to students and ask them to sign their attendance. The common characteristics of manual attendance system no require any specific equipment or tools implement on system. However, this type of system maybe suitable only for a small amount of students in a class. Moreover, it is time-consuming and ineffective when execute entire system.

Nowadays, due to advances and the evolution of technology, there are various of attendance systems have been developed. For example, there are IoT based attendance system, biometric based system, web based attendance system, smartphone based attendance system and many more. These systems are more efficient and advances compare to manual attendance systems. Reduce human based error on taking attendance and time-saving are some of the advantages. However, not all but some of these technologies exhibit some certain functional limitations. Taking student ID smart card as an example. The smart card based system work by having students scan their ID smart card to record their presence during class session. The limitation is smart card could be abused by having other student to scan the card for him/her. The system unable to track the attendance well as his/ her attendance has been recorded but without that student physically attend the class session. Biometric identification might be the better choice on this issue. Everyone has a unique set of biometrics that differs from each other. This is our own personal ID.

Therefore, this proposed project will use a biometric method: face recognition algorithm implemented in Raspberry Pi to recognize the student name based on their picture or video taken in real time during class session. Face recognition technology provides many benefits such as reduce human error, provides quickly verification of users, stand as the reliable system and etc.

1.2 Problem statement

Time-consuming is one of the problems that exist in attendance system. Action of attendance taken need to repeated in every single class. It is better to save as much time as possible on this attendance issue. Futhermore, against the fake attendance or proxy attendance might be the biggest issue to concern when develop the attendance system. There are many factors for students to create proxy attendance. Some of the factors are students might be lack of interest to particular subject. Lost of motivation to attend class will start to affect academic performance of student. The helping from friend is the main cause of this issue. As being mentioned, some class attendance system using student ID smart card, student can fake attendance by having other student to scan the card for him or her. Eventually, this will cause not reliable to accuracy of the attendance system.

Other than that, some case for example some of the students sometimes is absent mind and forget to bring their student smart ID card to attend class. Extra tools or requirements like smart card must be carried with students for some of attendance system. Although students still can join the class, but not attendance will be given. Moreover, the cost of student smart ID card as every student will be assigned one card. The educational institution has to take over the cost when these cards snapped off or broken by student from time to time. This is quite a lot of expenses by consider there are lot of students in institution.

1.3 Objective

There are two objectives that need to achieve in this project. Following are the objectives:

1. To develop class attendance system using face recognition with Raspberry Pi.
2. To track and manage student attendance efficiently.

1.4 Scope of project

The following are the scope of the project:

- This project focuses on use face recognition algorithm implemented in Raspberry Pi.
- Raspberry Pi 3 model B is the single-board computers used in this project.
- A 5-megapixel Raspberry Pi camera module V1 will used for image capturing purpose such as capture images of attendant.
- A monitor will be connected with Raspberry Pi as a display device.
- The programming language used is Python.
- The Python IDE from Raspbian OS will be used as code editor.
- OpenCV, library of programming functions integrate with python is used for image processing and plays a major role in real-time operation.

1.5 Thesis Outline

This document outline consists of total five chapters. Chapter 1 is introduction and brief about project which include the background of project, problem statement, objectives of project and scope of project. Chapter 2 is literature review which focus on background studies on attendance systems that developed early and other more research related to this field. Chapter 3 discussed about the methodology used in this project. It include the flowchart, brief methodology, material or components used. In short, all the method and overall approach to the project will be listed. Next, the final outcome of project and its discussion are stated in chapter 4. Lastly, the final conclusion and future work about the project will be carried out in chapter 5.

CHAPTER 2

LITERATURE REVIEW

.

There are various types of attendance systems developed nowadays which based on different technology and method. Some of attendance system have been proposed and tested in educational institutions field especially in university or higher education state while some other system not being used before. Manual attendance system will not be review in this chapter due to its limitations and not adapted. Therefore, this paper will review on existing attendance systems that have been developed and other research related to this field. The relevant sources that reviewed in this chapter were from year 2014 till 2020.

For the non-biometric based attendance system for instance Sudha et al. purposed the system that used barcode scanner to scan student's ID card and store attendance database into computer. Email will be sent to student if their attendance percentage are less than requirement[2]. Next, QR scanning based system[3,4] via smartphone implement in low cost and ease to use. Likewise, Hooi et al. used the same technology but used extra smartphone features like IMEI, timestamp and GPS to verify attendance[5]. Lodha et al. used Bluetooth smart or Bluetooth Low Energy technology that connects electronic tags with smartphone to record attendance. In that system, a application on smartphone used to check validity of electronic tags[6]. System from Honnalli and Prasad composed of Bluetooth module and Arduino connected with computer. Main purpose of Bluetooth module is to scan Bluetooth MAC addresses of student's smartphone while process of matching with database and record attendance are conducted in Arduino device. Then, attendance details will be email to respective educator[7]. Moreover, the system proposed by Mrabet and Moussa utilizes RFID technology, Arduino board and IoT technology. Processing time of the system was under 5 minutes on average. Result of system showed that absence between students were decreased[8]. However, the issues of proxy attendance still exists in the system. As review, there are more attendance system that based on smartphone or use it as one of tools in system have been developed and the number keep on increasing. This due to convenient and popularity of smartphone usage nowadays[9].

For the biometric based attendance system for example Abubakar Adamu purposed both method which was fingerprint and iris technique into the system[10]. O.Shoewu et al. purposed system that worked by fingerprint method and microcontroller. Additional features such as embedded thermal printer to print attendance list and Bluetooth feature used to send softcopy of attendance data to smartphone were included. Authors stated that the system can make available in job sectors where the system also can calculate workers's paycheck[11]. Furthermore, Djoanna Marie Vasquez Salac discussed system that used three different technologies. There are face recognition, Android operating system device and SMS. The system was totally portable. System used the camera feature that available on the smartphone to do the recognition process. An application developed in Android device have functions such as compute the attendance grade, allow students to view their attendance record. SMS service is used to notify parents about students's attendance record[12]. Moreover, Noradila Nordin and Nurul Husna described web-based attendance system with face recognition features. JavaScript API with TensorFlow library used for face detection and recognition process. A webcam used for capture image. However, there was feedback from end-user stated that system should be in form of application and guidance from developer was needed when used the system[13]. Another study from Prangchumpol (2019) focused on the technique of Android Face Recognition with Deep Learning on his system. The database stored in web server using cloud storage. Based on experimental result, author claim that his system has up to 97% on average accuracy of classifier compare to Euclidean distance(ED) has 96.21% and Linear discriminant analysis (LDA) has 96%. However, training time was taken longer which was 182.01 seconds compare to 50.47 seconds and 45.04 seconds on other two classifier[14]. P.Reddy and N.Raju (2020) proposed face recognition system using Eigenfaces method based on Principal Component Analysis (PCA) algorithm and implemented by MATLAB software. Image Acquisition and Image Processing toolbox from MATLAB were used[15]. Eigenfaces method converted face image to vector form that reduced the dimension on input image while PCA algorithm helped system to extract facial features efficiently[16]. Furthermore, Mehta and Tomar purposed Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG) for features extractions and Support Vector Machine (SVM) classifier for face recognition. The system used MATLAB to implemented these algorithms and process them. Pi camera connects with Raspberry Pi 2 was place in classroom to capture image of student. Then, Raspberry Pi connect to MATLAB via Wi-Fi dongle that plug into Pi. Authors reported that system got accuracy of face recognition up to 92%[17].

In conclusion, among various types of technology used in attendance system listed and many other more system that not covered in this chapter, so face recognition method was utilize and implemented with Raspberry Pi in this paper. Face recognition method is a common method in category of biometric based for attendance system due to real-time processing, convenience and time-saving.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter discussed the appropriate method and overall procedure that executed in this project. There are few sections that listed such as the flowchart of project, overall system development which divided into hardware and software section, method and algorithms used for face detection and recognition, material used in project and etc. The detail information listed in each sections below.

3.2 Flowchart of project

The flowchart shown in figure 3.1 illustrated the workflow of attendance taking process of the system. First and foremost, student needed to register to the system by enter their name, student id and facing the camera to capture portrait. It aimed to create the dataset on Raspberry Pi. To create the dataset, there were several process need to be executed. Once the image being captured, it will undergo a face detection process to identify whether has a face in every portrait. After these captured images with confirmed got face detected will undergo pre-processing process like cropping the face region then converted into grayscale image. Next, these face data and their respective id's of each face will feed into the recognizer which is provided by OpenCV library to learn and do the training. Once the training process is done, these set of trained data will be store for further face recognition process. When student taking attendance, they have to face the camera to capture portrait. These new capture portrait will undergo the same face detection, pre-processing process. Then, face recognition process will be happening in next state. New captured portrait will be compared against trained data. If results do not match, student's name will not be displayed on screen and no attendance will be recorded otherwise if results are match, student's name will be display on screen and attendance will be recorded by system.

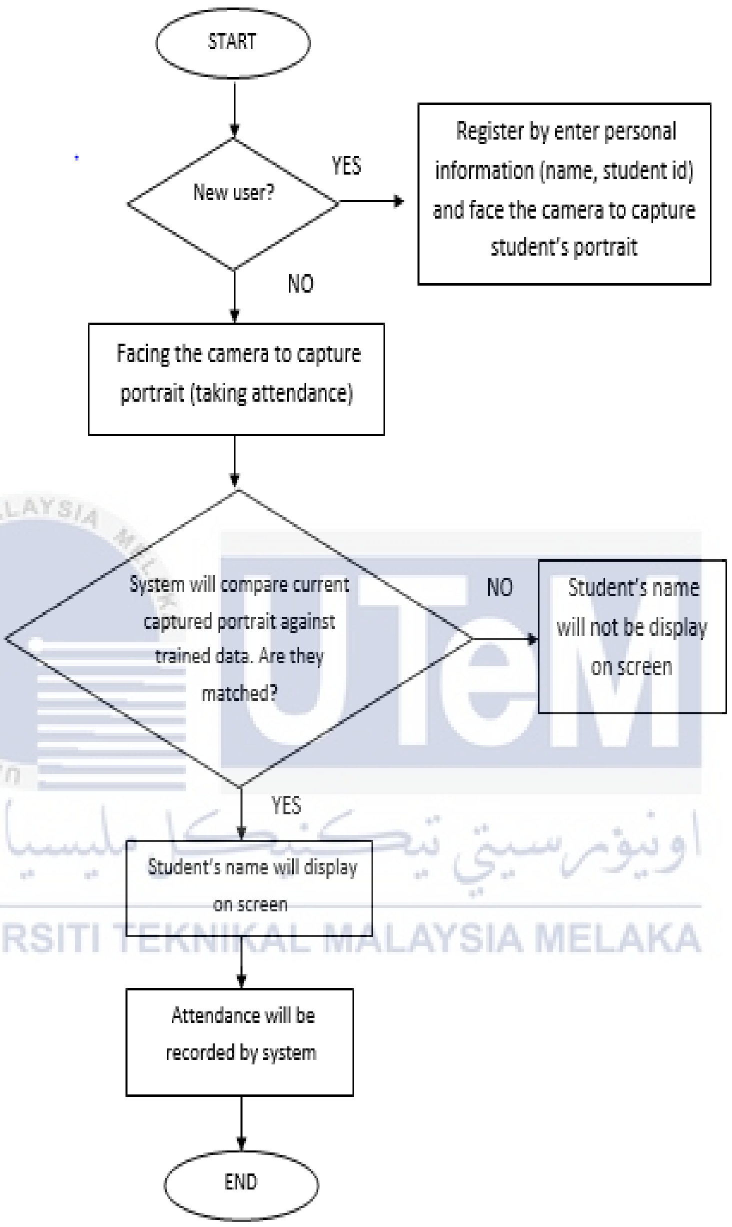

START
New user?
YES
Register by enter personal information (name, student id) and face the camera to capture student's portrait
NO
Facing the camera to capture portrait (taking attendance)
System will compare current captured portrait against trained data. Are they matched?
NO
Student's name will not be display on screen
YES
Student's name will display on screen
Attendance will be recorded by system
END

Figure 3.1: The flowchart of attendance taking process of system

3.3 Proposed system

The system consists of hardware and software part. For the hardware development consists of several components which are power supply adapter, a micro SD card, a Raspberry Pi camera module V1, both USB keyboard and mouse are interfaced with Raspberry Pi 3 model B. The Raspberry Pi connect to monitor via standard HDMI cable. The block diagram of hardware setup was shown in figure 3.2.

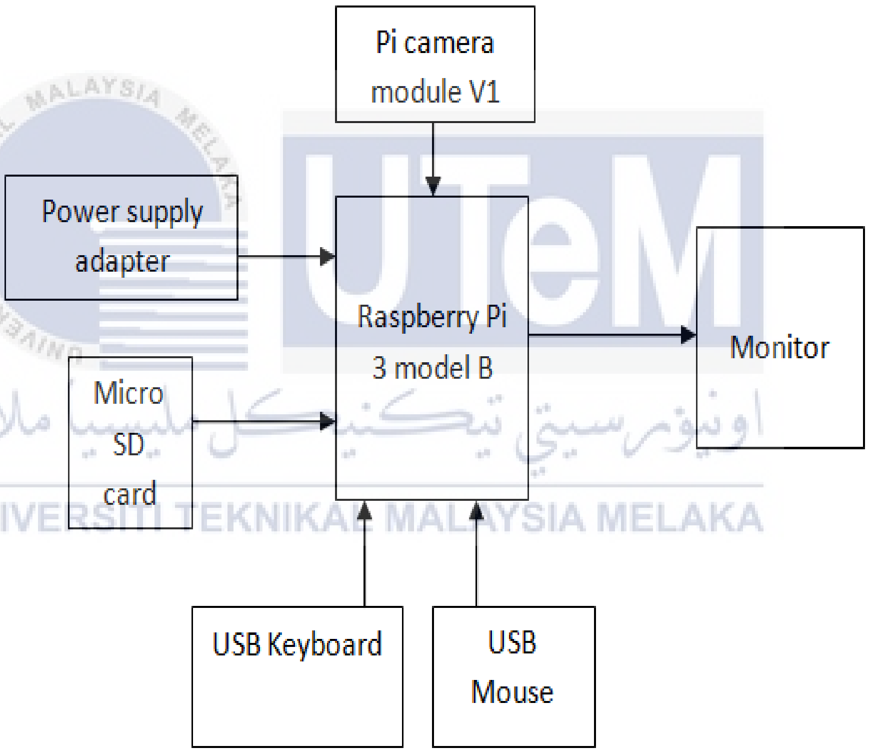


Figure 3.2: Block diagram of hardware setup

3.3.1 Hardware setup

Board description of Raspberry Pi shown in figure 3.3. The power supply adapter was plug into Micro USB connector on board of Raspberry Pi. Micro SD card was inserted to Micro SD card slot which locate at underside of board. Then, the camera module installed by insert the ribbon cable to CSI camera connector. Both the keyboard and mouse connected to USB 2.0 port. For the display, a standard HDMI cable was used to plug into HDMI connector on board and other end plugged into HDMI connector on monitor side. Connection of hardware was shown in figure 3.4.

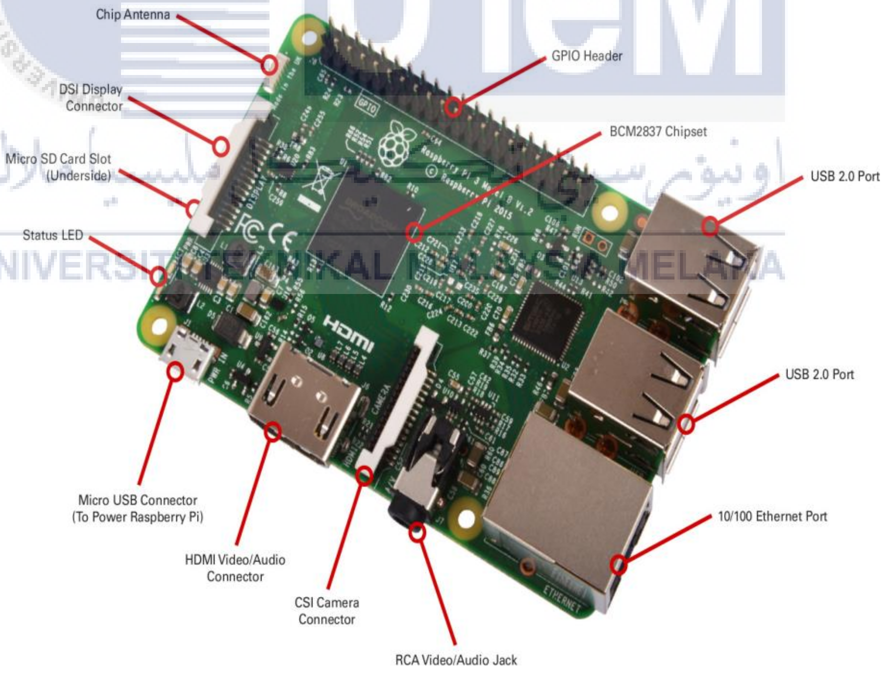


Figure 3.3: Board description of Raspberry Pi[18]

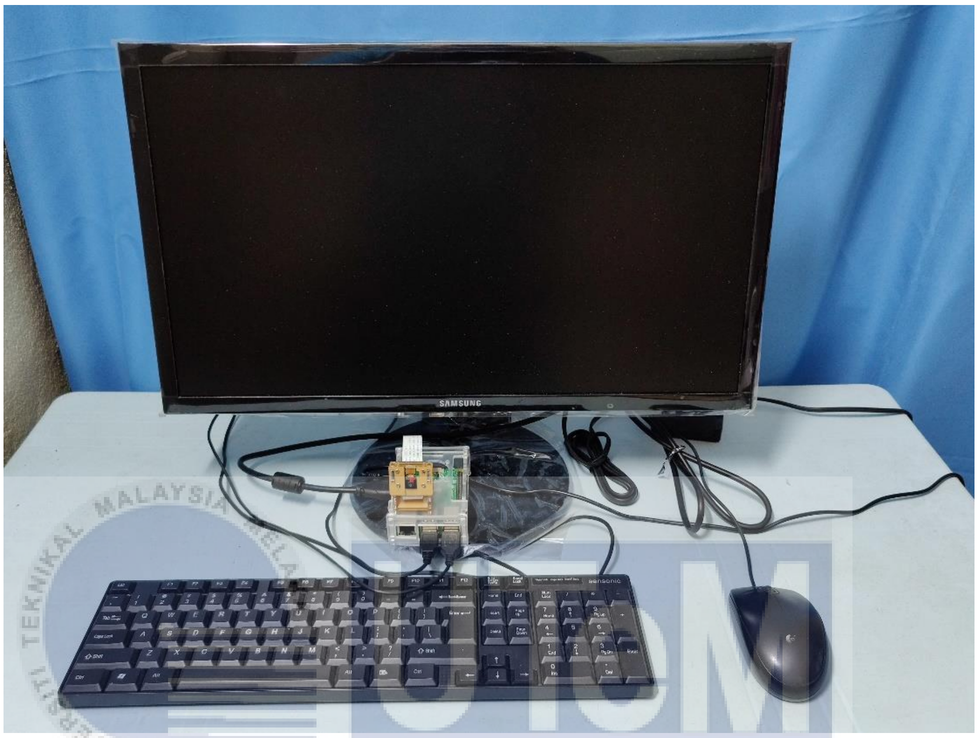
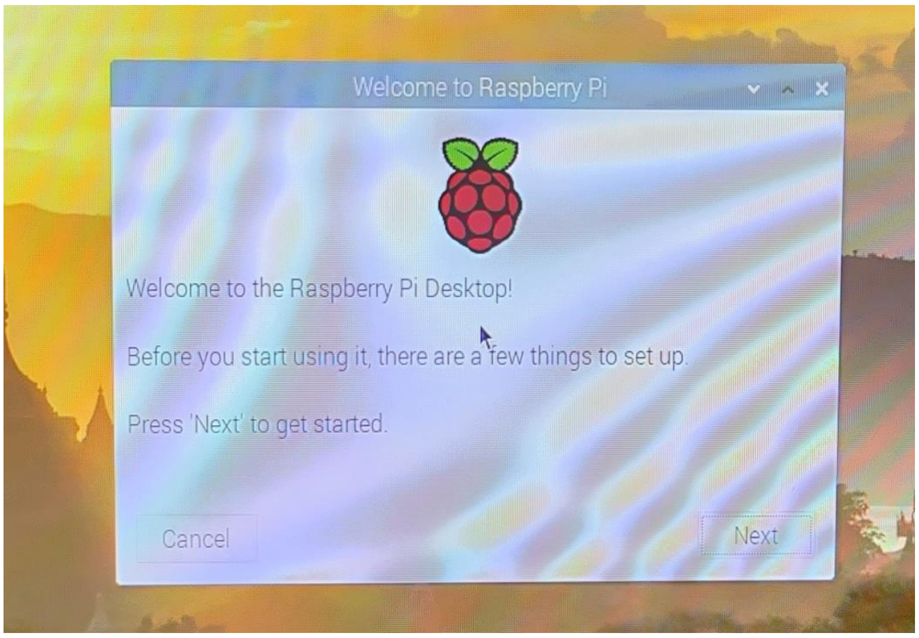


Figure 3.4: Connection of hardware part

3.3.2 Software setup

First things first, it needed to install the operating system onto a Micro SD card for Raspberry Pi. To install the operating system, browse to the official website of Raspberry Pi and download the New Out Of Box Software (NOOBS)[19]. NOOBS is an operating system install manager for Raspberry Pi. Once downloaded the file, extract the files and move all the files onto Micro SD card. Next, insert the Micro SD card into Raspberry Pi and boot it. The window will display and choose the Raspbian. After a normal reboot and system is ready to be used. Figure 3.5 shown the welcome screen of Raspberry Pi.



Welcome to Raspberry Pi

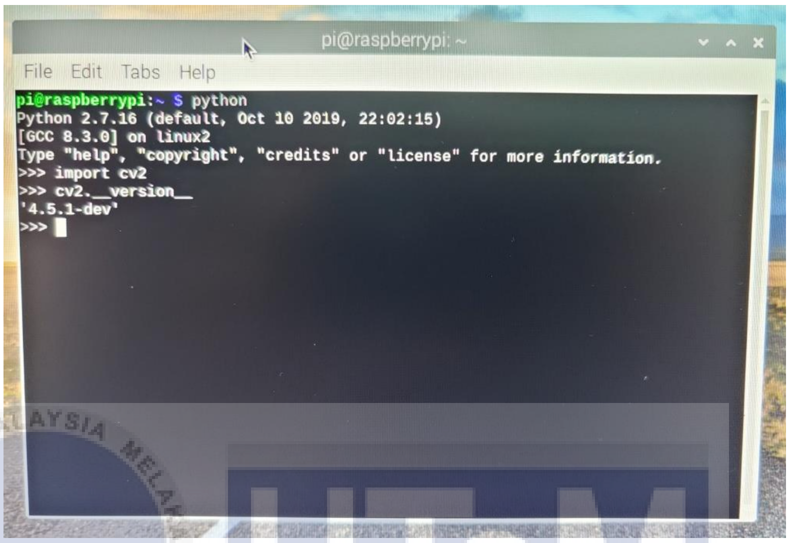
Welcome to the Raspberry Pi Desktop!

Before you start using it, there are a few things to set up.

Press 'Next' to get started.

Figure 3.5: Welcome screen of Raspberry Pi

To install OpenCV to Raspberry Pi, there are some of the steps and need to wait some period of time to install it successfully. All the installation process are run on command . First of all, installed packages are needed for OpenCV. These packages include tools to compile the code, package-management system(pip), NumPy library. Then, changed the size on swap file before the process of compiling OpenCV. Due to the limited RAM on Raspberry Pi, increase of swap file can use as additional memory to support the process of compiling OpenCV. Furthermore, clone the OpenCV repositories and get the latest version. After that, use CMake to compile the OpenCV. Functions of CMake manage the build process in operating system, compile the source code and etc. This step takes most of the time of the entire process which around 6 hours. Once this step is done, run an install command. A 100% message will be shown at output indicated all process finished. Last step is to change back initial size on swap file. To check whether OpenCV install on Pi, run some command and output shown in figure 3.6. The 4.5.1 version of OpenCV install successfully on the Raspberry Pi.



```
pi@raspberrypi:~ $ python
Python 2.7.16 (default, Oct 10 2019, 22:02:15)
[GCC 8.3.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.5.1-dev'
>>> 
```

Figure 3.6: Result of OpenCV installation

3.4 Method used for face detection and recognition

For the overview process of face recognition, first stage must undergo the face detection process, then pre-processing process, dataset creation and lastly face recognition process. Face detection was a different concept from face recognition. Face detection is technique to identify and detect whether there is face presence in digital image, frames or portrait. There are many difference technique and algorithms used to perform the process. Haar classifier is used as face detection classifier in this study. The Haar classifier is based on algorithm by Paul Viola and Michael Jones in their research[20]. This algorithm processed on grayscale image. Figure 3.7 shown the steps of Viola-Jones algorithm.

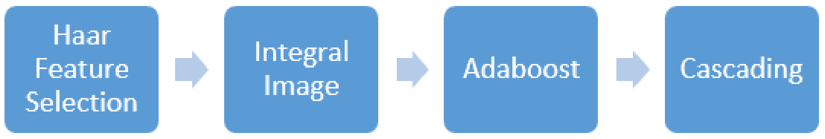


Figure 3.7: Steps of Viola-Jones algorithm

First step is Haar feature selection and used to extract features from image by using three types of features. The three types of features are edge features, line features and four-rectangle features as shown in figure 3.8. These features are scalable and not fixed by any size. By apply these features to determine the relevant parts of face from input image. Then, it obtained a value by computed the difference between sum of pixels under black rectangle and sum of pixels under white rectangle. This computation process is using Integral image method. The relevant parts are based on common properties of our face. For instance, our eye region are darker than upper cheeks and nose bridge region.

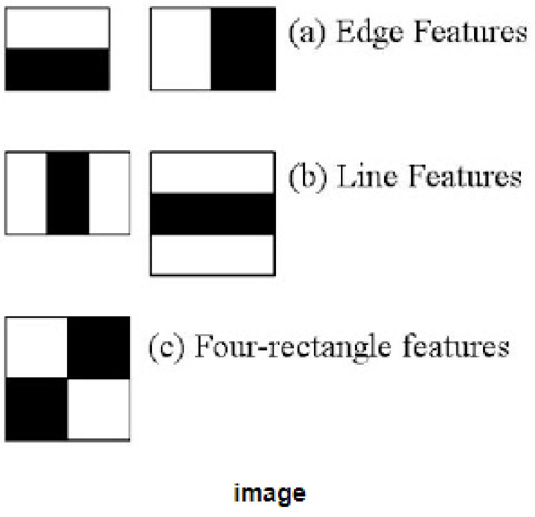


Figure 3.8: Three types of Haar feature[21]

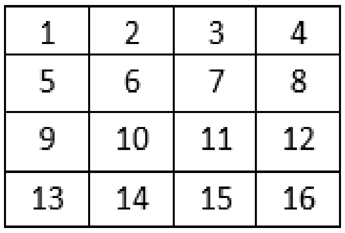
Second step is used Integral image to compute the sum of pixel value inside these features in a quicker way. It is based on summed-area table algorithm. This works by computed the sums of pixels above and toward left of that point. The equation is defined in equation 3.1 below and it can calculate recursively in equation 3.2.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{3.1}$$

$$ii(x, y) = i(x, y) - ii(x-1, y-1) + ii(x, y-1) + ii(x-1, y) \tag{3.2}$$

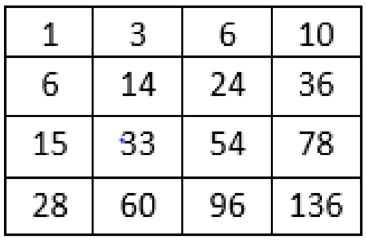
where $i(x, y)$ is value of pixel at its position (x, y), $ii(x, -1)$ and $ii(-1, y) = 0$.

For example, a input image shown in figure 3.8 and its corresponding integral image shown in figure 3.9.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Figure 3.9: Input image, i



1	3	6	10
6	14	24	36
15	33	54	78
28	60	96	136

Figure 3.10: Integral Image, ii

Next, Adaboost or full name adaptive boosting is used to eliminate irrelevant features so that it would not be consider for further evaluation. The term boost is to increase the classification performance of weak learners. It combines multiple weak classifiers to form a strong classifier. Procedure of Adaboost are firstly given example images (x_1, y_1) and initialize the weights to their number of negative and positive respectively. Then, normalize weights for the classifiers. Calculate the error using equation 3.3 below. After that, choose the classifier with the lowest error. Update the weights of classifiers by adding more weight on incorrectly classified and reduce weight on correctly classified. This process will be repeated until accuracy or number of features are found. Final step is to form a strong classifiers by combine these weak classifiers. Equation for strong classifiers shown in equation 3.4.

$$\sum_i w_i \left|h_{j(x_i)-y_i}\right| \qquad (3.3)$$

$$F(x) = a_1 f_1(x) + a_2 f_2(x) +.. a_n f_n(x) \qquad (3.4)$$

where a_n represent weight and $f_1(x)$ represent weak classifier.

Last step is set up a cascading system to reject non-face image that are selected from the previous Adaboost process. It has multiple stages in cascading system. The main purpose is to saving computation time and speed up for face detection in real time. Figure 3.11 shown the process of cascading. All sub-windows loaded to stage and window that pass through all the stage detected as face region. If any of these sub-window fails within stage, the system will automatically reject or discarded it. It is better to put the strong classifier in the early stage of system. Therefore, it able to reject as many negative images as possible. Based on results obtained by Viola-Jones in their paper, they use the total number of 6061 features in 38 layers cascaded classifiers to detect frontal upright faces[20].

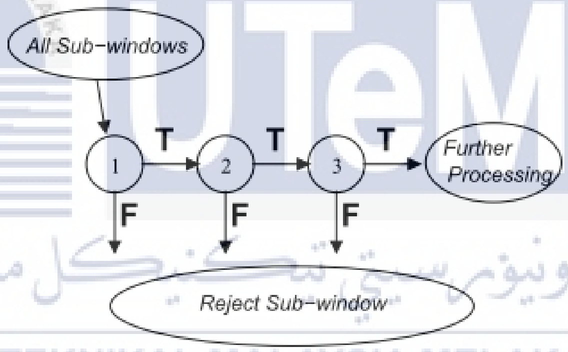


Figure 3.11: Cascading process [20]

Face recognition is the process that able to identify and match who is that person in digital image or frames by comparing against its database. Local binary patterns histograms(LBPH) face recognizer is used as face recognizer in this study. This method purposed by Ojala et al. in their research[22]. Once the dataset with respective ID are feed into the recognizer, it start to do the training. So, the steps of operation of LBPH are first when the input image is convert to grayscale, it look at

3x3 pixels on image. Compare center pixel with its neighbor pixel value. There are two conditions listed, if neighbor pixels is equal or larger than center pixel, set the value as 1 otherwise set as 0. Hence, new pixel should be in form of 0 or 1. Next, update these new pixels in clockwise order. Then, it should form as binary number. Convert these binary number to decimal number. The decimal number is the new value of the center matrix for that 3x3 pixels. Repeat the procedure for every 3x3 pixels in image. Figure 3.12 shown the LBP operator. Equation of LBP operator is expressed in equation 3.5. Furthermore, extract into histograms by using grid x and y parameter. Therefore, one histogram represents each image. If the distance between histogram of input image closer to histogram inside dataset, then it will be recognized. Equation to calculate distance between histograms shown in equation 3.6. Figure 3.13 shown trained image and its corresponding histogram.

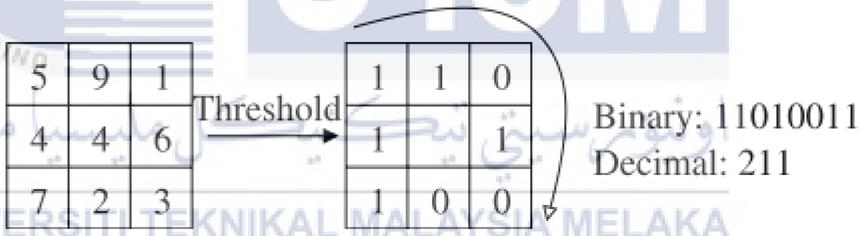


Figure 3.12: LBP operator[23]

$$\text{LBP} = \sum_{n=0}^{7} S(i_n - i_c)2^n \tag{3.5}$$

where i_n = Neighbour pixel value and i_c = Centre pixel value

$$S(z) = \begin{cases} 1, & z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$\boldsymbol{D} = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{3.6}$$

where q and p are two LBP histograms from new input image and dataset.

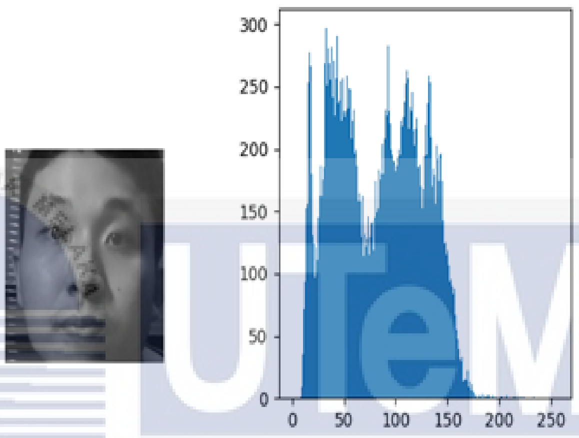


Figure 3.13: Trained image and its histogram

3.5 Material description/used in project

a) Raspberry Pi 3 model B

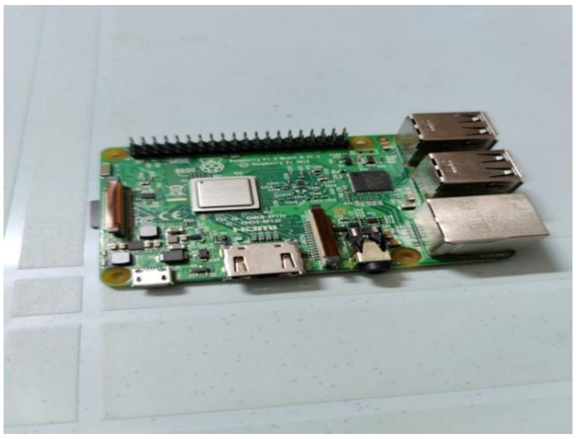


Figure 3.14: Raspberry Pi 3 model B

Raspberry Pi that used in this project is its third-generation model that released in early 2016. This model has features like quad-core 64-bit ARM Cortex-A53 CPU running at 1.2GHz, run on 1GB LPDDR2 Ram,4 USB 2.0 ports, wired and wireless networking, HDMI and composite video output, and a 40-pin GPIO connector for the physical interfacing projects.

a) Pi camera module v1

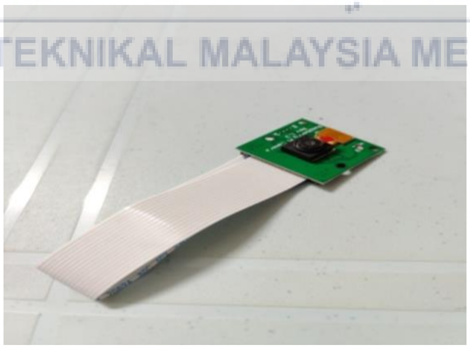


Figure 3.15: Pi camera module v1

The camera that used is Pi camera module v1 that have a resolution of 5 Megapixels. For the video modes can support up to 1080p 30fps. It is small in size for around 25 × 24 × 9mm.

a) Micro SD card

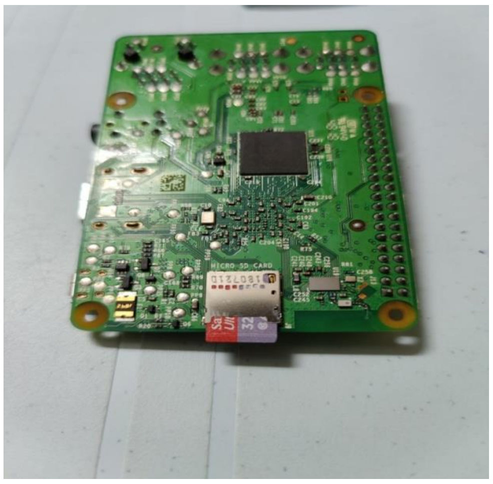


Figure 3.16: Micro SD card

A micro SD card is necessary in any Raspberry Pi as it functions to be the storage for operating system and memory. Capacity of 32GB micro SD card is used in this project.

b) Power supply adapter

Figure 3.17: Power supply adapter

The 5V 2.5A AC/DC adapter is used to power the Raspberry Pi. It is micro USB output connector.

c) USB Wired Keyboard and Mouse

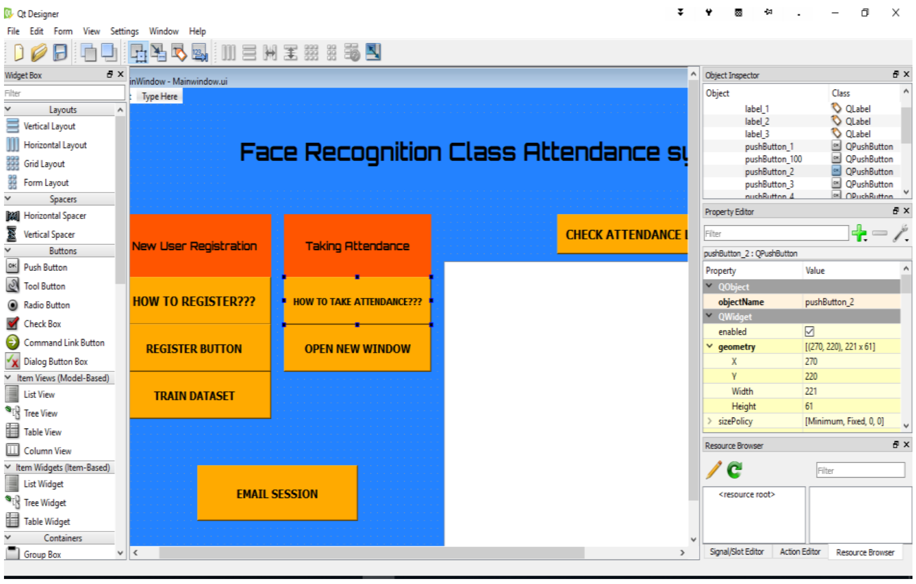
d) Standard HDMI cable (Type A)

e) LED Monitor

Figure 3.18: LED Monitor

3.6 GUI software

In order to make user friendly, this system used PyQt5 to form GUI (graphical user interface). Qt is a multi-platform GUI library that works on many platforms such as Android, Linux, Windows, iOS and etc. PyQt5 is the one that compatible with Python. From its software, Qt Designer provides user to build and design own GUI using components provided such as item widgets, display widgets, buttons and etc. The working procedure is described as follow. Once finished design the user interface in Qt Designer, save the file. File will be saved as UI file extension. Then, convert file to Python file extension using pyuic5 utility through command window. Next, coding on that Python file and assigned the function to these widgets or buttons in order to execute command. For the sub-window that have separate script, then link these script to the main window script. Last but not least, do GUI testing to ensure functions that assigned work as expected. Figure 3.19 shown this system's GUI design on Qt Designer.



Object	Class
label_1	QLabel
label_2	QLabel
label_3	QLabel
pushButton_1	QPushButton
pushButton_100	QPushButton
pushButton_2	QPushButton
pushButton_3	QPushButton
pushButton_4	QPushButton

Property Editor

pushButton_2 : QPushButton

Property	Value
QObject	
objectName	pushButton_2
QWidget	
enabled	✓
geometry	[(270, 220), 221 x 61]
X	270
Y	220
Width	221
Height	61
sizePolicy	[Minimum, Fixed, 0, 0]

Figure 3.19: System's GUI design on Qt Designer

3.7 Conclusion

At the end of chapter, method of the project has been explained. The hardware setup has been done. An operating system for Raspberry Pi and OpenCV library has also been installed. The further result and discussion will be explored in next following chapter.

CHAPTER 4

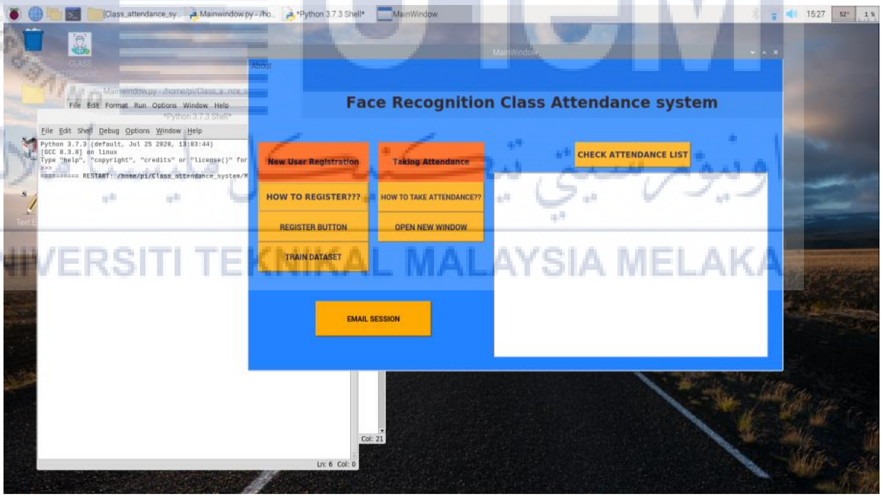
RESULTS AND DISCUSSION

4.1 Introduction

By using method as mentioned in previous chapter, the result is obtained and discussed in following section. Then, analysis has been carried out on how different settings affected the confidence measurement of image histogram.

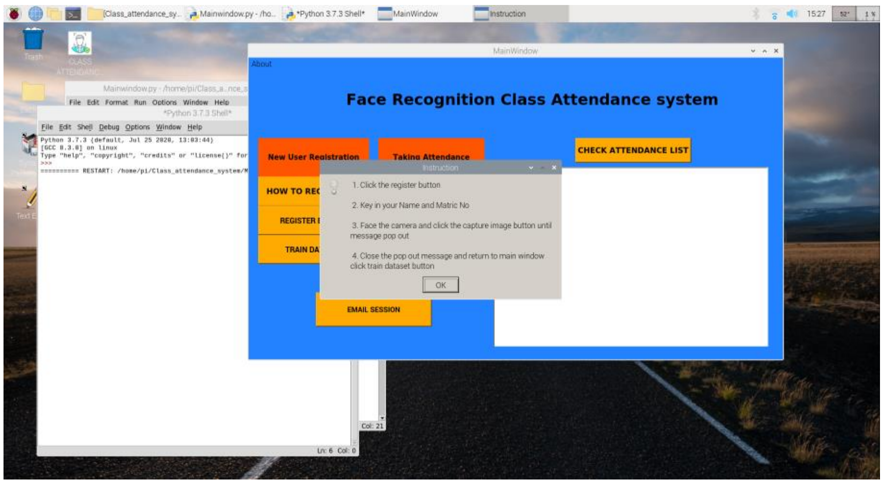
4.2 Results from system

Start the system by running the main script. Figure 4.1 shown the main window of system. There are three main parts on this system. Followed the flowchart in figure 3.1. First, start from new user registration part. There are three buttons under this part. By clicked how to register button, it pops up the instruction window to guide new user. New user need to follow in order to complete the registration process. Figure 4.2 shown the instruction window for registration.



```
Python 3.7.3 (default, Jul 25 2020, 13:03:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for
>>>
========== RESTART: /home/pi/Class_attendance_system/M
```

Figure 4.1: Main window of system



```
Python 3.7.3 (default, Jul 25 2020, 13:03:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license()" for
>>>
========== RESTART: /home/pi/Class_attendance_system/M
```

1. Click the register button

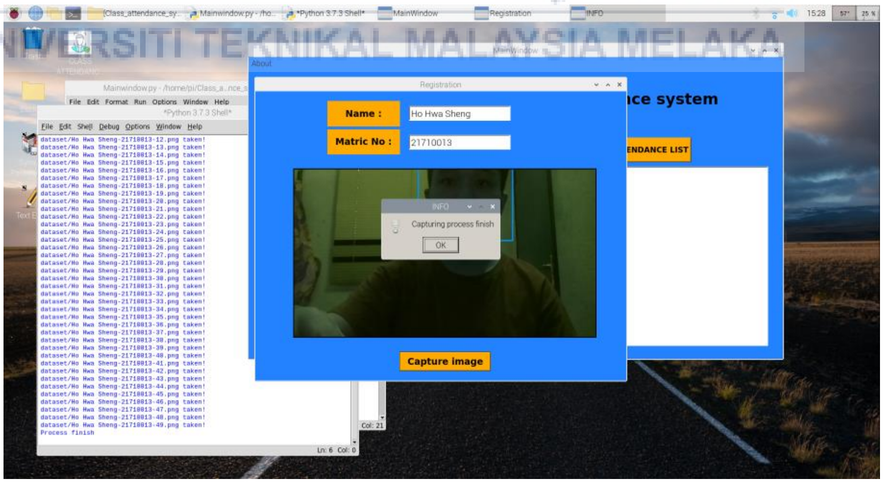
2. Key in your Name and Matric No

3. Face the camera and click the capture image button until message pop out

4. Close the pop out message and return to main window click train dataset button

Figure 4.2: Instruction window for registration

Next, click the register button and fill in the name and matric number. Then, click the capture image button and face the camera. The capturing process done when the info message pop out. Figure 4.3 shown the message when capturing process done. All the trained images saved in dataset folder sort by user's name as shown in figure 4.4. As soon as process done, click the train dataset button to complete the registration. Figure 4.5 illustrated the message when training process done.



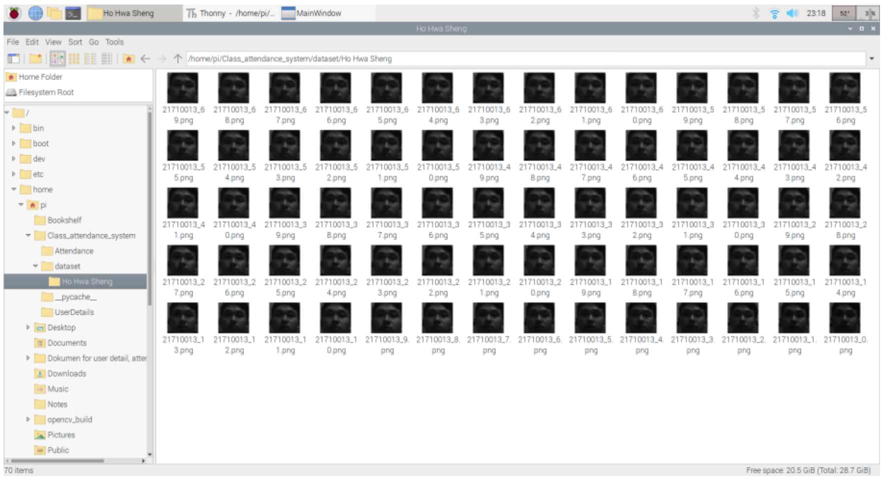
Name : Ho Hwa Sheng

Matric No : 21710013

Capturing process finish

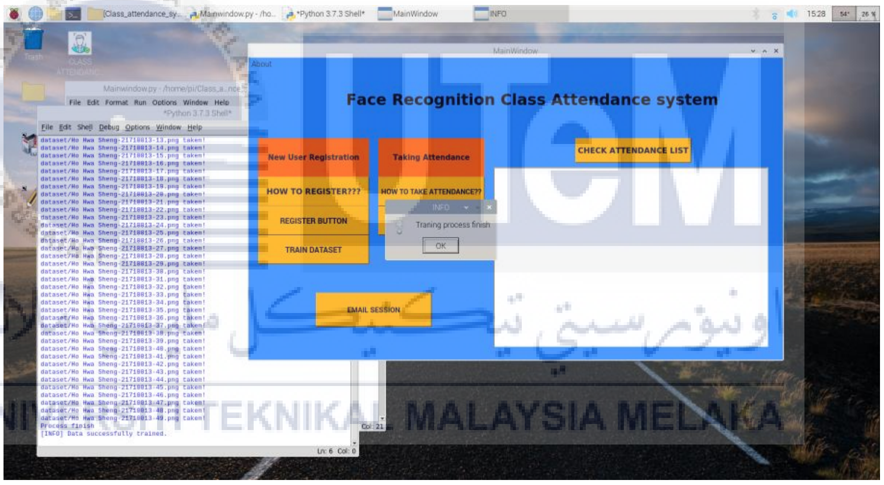
```
dataset/Ho Hwa Sheng-21710013-12.png taken!
dataset/Ho Hwa Sheng-21710013-13.png taken!
dataset/Ho Hwa Sheng-21710013-14.png taken!
dataset/Ho Hwa Sheng-21710013-15.png taken!
dataset/Ho Hwa Sheng-21710013-16.png taken!
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
```

Figure 4.3: Capturing process done



Ho Hwa Sheng
File Edit View Sort Go Tools
/home/pi/Class_attendance_system/dataset/Ho Hwa Sheng
Home Folder
Filesystem Root
70 items
Free space: 20.5 GiB (Total: 28.7 GiB)

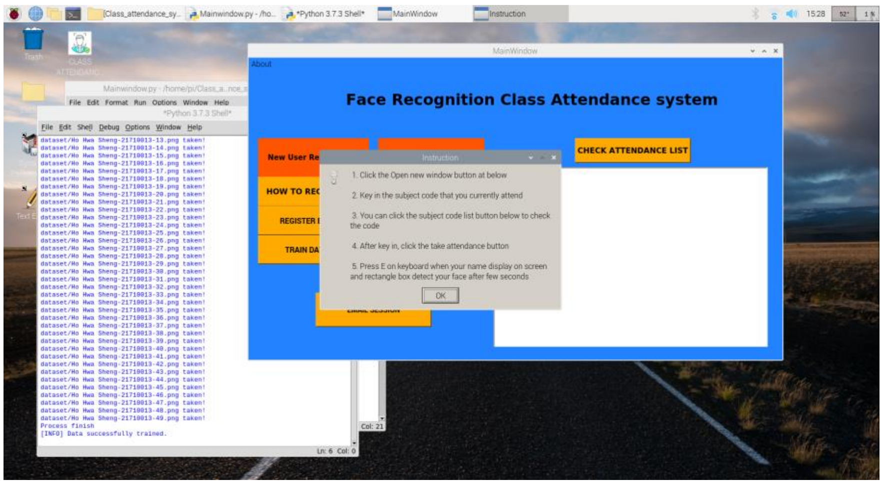
Figure 4.4:Dataset folder



```
dataset/Ho Hwa Sheng-21710013-13.png taken!
dataset/Ho Hwa Sheng-21710013-14.png taken!
dataset/Ho Hwa Sheng-21710013-15.png taken!
dataset/Ho Hwa Sheng-21710013-16.png taken!
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
```

Figure 4.5: Training process done

Moreover, move on to taking attendance part. There are two buttons under this part. Similar to registration part, the how to take attendance button also guide the user on step to take attendance from this system. Figure 4.5 shown the instruction window for take attendance. Furthermore, click the open new window button and key in the subject code that user currently need to attend. There is one subject code list button below to show the entire subject and code as the reference for user. This is shown in figure 4.6.



1. Click the Open new window button at below

2. Key in the subject code that you currently attend

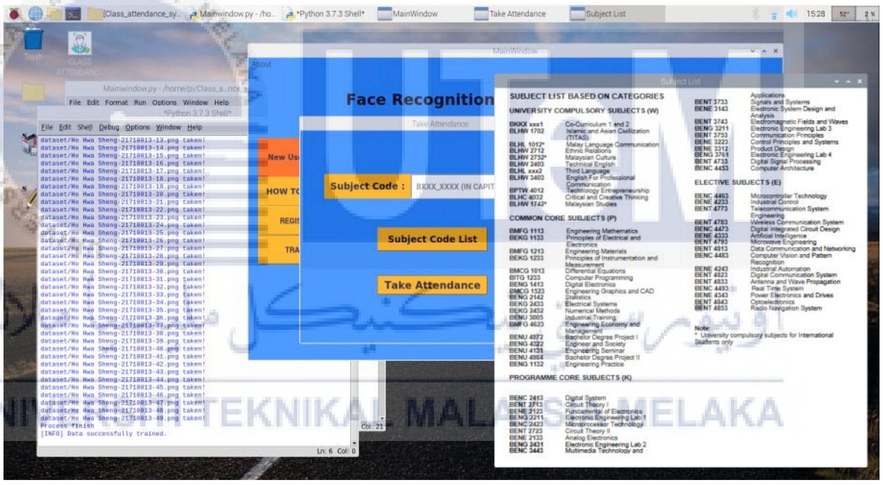
3. You can click the subject code list button below to check the code

4. After key in, click the take attendance button

5. Press E on keyboard when your name display on screen and rectangle box detect your face after few seconds

```
dataset/Ho Hwa Sheng-21710013-13.png taken!
dataset/Ho Hwa Sheng-21710013-14.png taken!
dataset/Ho Hwa Sheng-21710013-15.png taken!
dataset/Ho Hwa Sheng-21710013-16.png taken!
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
```

Figure 4.6: Instruction window for take attendance



SUBJECT LIST BASED ON CATEGORIES

UNIVERSITY COMPULSORY SUBJECTS (W)

BKKX xxx1	Co-Curriculum 1 and 2
BLHW 1702	Islamic and Asian Civilization (TITAS)
BLHL 1012*	Malay Language Communication
BLHW 2712	Ethnic Relations
BLHW 2752*	Malaysian Culture
BLHW 2403	Technical English
BLHL xxx2	Third Language
BLHW 3403	English For Professional Communication
BPTW 4012	Technology Entrepreneurship
BLHC 4032	Critical and Creative Thinking
BLHW 1742*	Malaysian Studies

COMMON CORE SUBJECTS (P)

BMFG 1113	Engineering Mathematics
BEKG 1133	Principles of Electrical and Electronics
BMFG 1213	Engineering Materials
BEKG 1233	Principles of Instrumentation and Measurement
BMCG 1013	Differential Equations
BITG 1233	Computer Programming
BENG 1413	Digital Electronics
BMCG 1523	Engineering Graphics and CAD
BENG 2142	Statistics
BEKG 2433	Electrical Systems
BEKG 2452	Numerical Methods
BENU 3005	Industrial Training
BMFG 4623	Engineering Economy and Management
BENU 4972	Bachelor Degree Project I
BENG 4322	Engineer and Society
BENU 4131	Engineering Seminar
BENU 4984	Bachelor Degree Project II
BENG 1132	Engineering Practice

PROGRAMME CORE SUBJECTS (K)

BENC 2413	Digital System
BENT 2713	Circuit Theory I
BENE 2123	Fundamental of Electronics
BENG 2211	Electronic Engineering Lab 1
BENC 2423	Microprocessor Technology
BENT 2723	Circuit Theory II
BENE 2133	Analog Electronics
BENG 2431	Electronic Engineering Lab 2
BENC 3443	Multimedia Technology and Applications
BENT 3733	Signals and Systems
BENE 3143	Electronic System Design and Analysis
BENT 3743	Electromagnetic Fields and Waves
BENG 3211	Electronic Engineering Lab 3
BENT 3753	Communication Principles
BENE 3223	Control Principles and Systems
BENE 3312	Product Design
BENG 3761	Electronic Engineering Lab 4
BENT 4733	Digital Signal Processing
BENC 4453	Computer Architecture

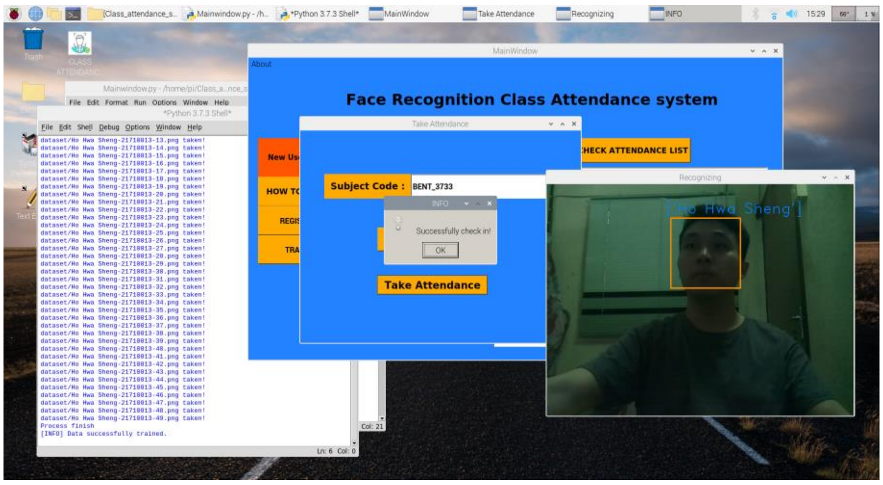
ELECTIVE SUBJECTS (E)

BENC 4463	Microcontroller Technology
BENE 4233	Industrial Control
BENT 4773	Telecommunication System Engineering
BENT 4783	Wireless Communication System
BENC 4473	Digital Integrated Circuit Design
BENE 4333	Artificial Intelligence
BENT 4793	Microwave Engineering
BENT 4813	Data Communication and Networking
BENC 4483	Computer Vision and Pattern Recognition
BENE 4243	Industrial Automation
BENT 4823	Digital Communication System
BENT 4833	Antenna and Wave Propagation
BENC 4493	Real Time System
BENE 4343	Power Electronics and Drives
BENT 4843	Optoelectronics
BENT 4853	Radio Navigation System

Note:
* University compulsory subjects for International Students only

Figure 4.7: Subject code list window

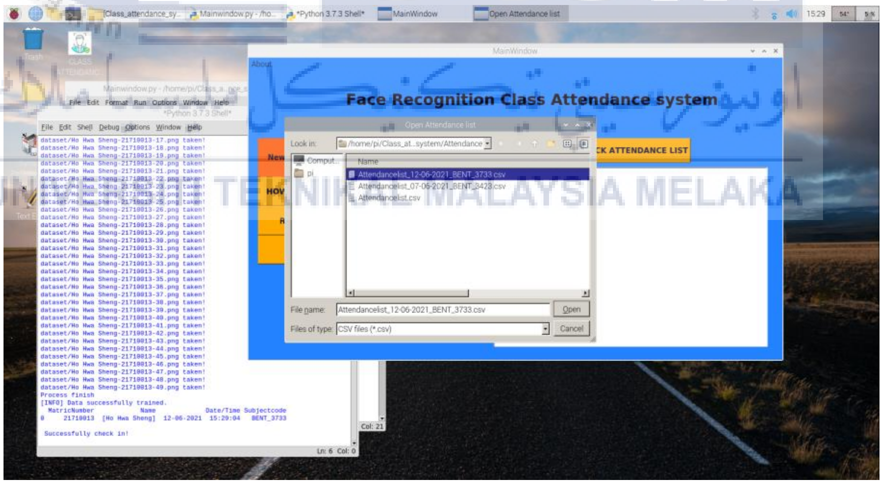
After key in subject code and click the take attendance button, a recognizing window will pop out. It will start to scan through frame and displayed user’s name once the user’s face has been recognized by system. Meanwhile, user need to press the e button on keyboard once their name displayed on window. One info message will pop out to inform user has successfully check-in. Figure 4.7 illustrated the situation mentioned. The recognized user will be automatically added into attendance list.



```
dataset/Ho Hwa Sheng-21710013-13.png taken!
dataset/Ho Hwa Sheng-21710013-14.png taken!
dataset/Ho Hwa Sheng-21710013-15.png taken!
dataset/Ho Hwa Sheng-21710013-16.png taken!
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
```

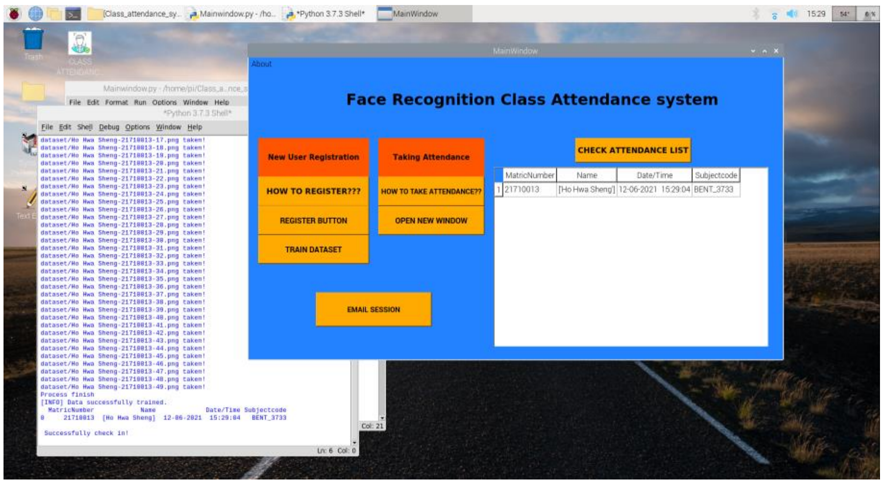
Figure 4.8: Success check in message

User can check and preview their attendance status by click the check attendance list button on main window. Select and open the attendance file with current date and subject code as shown in figure 4.8. Then, all the detail will be displayed on table widget as shown in figure 4.9.


Face Recognition Class Attendance system
Open Attendance list
Look in: /home/pi/Class_at...system/Attendance
Name
Attendancelist_12-06-2021_BENT_3733.csv
Attendancelist_07-06-2021_BENT_3423.csv
Attendancelist.csv
File name: Attendancelist_12-06-2021_BENT_3733.csv
Files of type: CSV files (*.csv)
Open
Cancel

```
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
  MatricNumber           Name            Date/Time Subjectcode
0     21710013  [Ho Hwa Sheng]  12-06-2021  15:29:04   BENT_3733

 Successfully check in!
```

Figure 4.9: Open attendance file

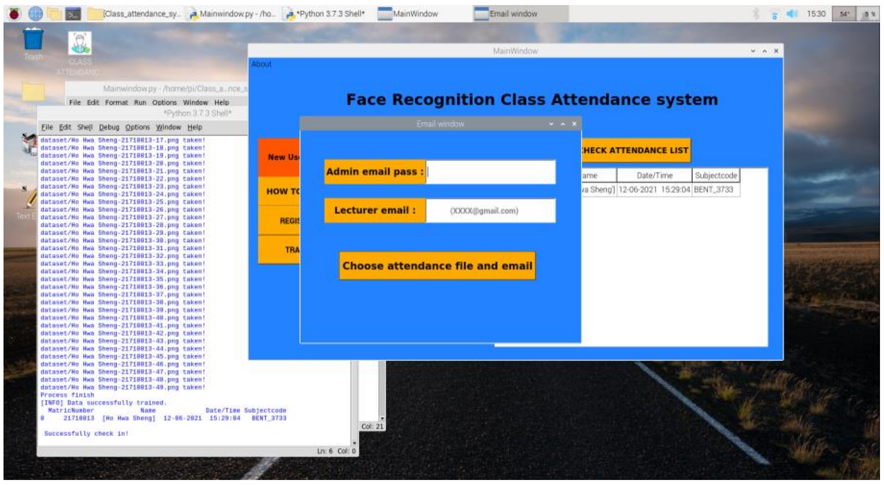


	MatricNumber	Name	Date/Time	Subjectcode
1	21710013	[Ho Hwa Sheng']	12-06-2021 15:29:04	BENT_3733

```
dataset/Ho Hwa Sheng-21710013-17.png taken!
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
  MatricNumber           Name          Date/Time Subjectcode
0    21710013  [Ho Hwa Sheng]  12-06-2021  15:29:04   BENT_3733

 Successfully check in!
```

Figure 4.10: File display on table widget

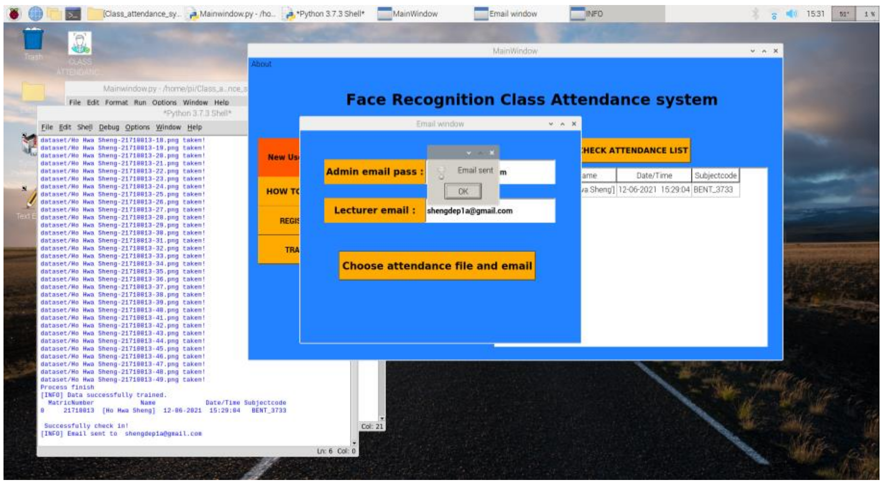
Last step is to email the relevant attendance list to lecturer. This part can only operate by lecturer. Click the email button and email window will open up as shown in figure 4.10. For the admin email pass, it is password for the admin email or sender email by this system. Only lecturer will be assigned the password. This system used the Gmail SMTP (Simple Mail Transfer Protocol) server so the receiver need to key in their Gmail. After that, click the button below and choose the attendance file that want to email. System will start email to receiver. This process needs to wait few seconds and a message will pop out once email sent as shown in figure 4.11.


Face Recognition Class Attendance system
Email window
Choose attendance file and email
CHECK ATTENDANCE LIST

Admin email pass :

Lecturer email : (XXXX@gmail.com)

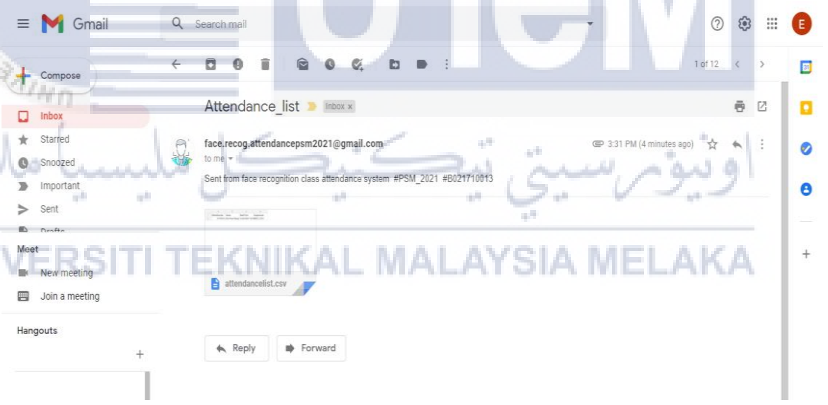
Figure 4.11: Email window



```
dataset/Ho Hwa Sheng-21710013-18.png taken!
dataset/Ho Hwa Sheng-21710013-19.png taken!
dataset/Ho Hwa Sheng-21710013-20.png taken!
dataset/Ho Hwa Sheng-21710013-21.png taken!
dataset/Ho Hwa Sheng-21710013-22.png taken!
dataset/Ho Hwa Sheng-21710013-23.png taken!
dataset/Ho Hwa Sheng-21710013-24.png taken!
dataset/Ho Hwa Sheng-21710013-25.png taken!
dataset/Ho Hwa Sheng-21710013-26.png taken!
dataset/Ho Hwa Sheng-21710013-27.png taken!
dataset/Ho Hwa Sheng-21710013-28.png taken!
dataset/Ho Hwa Sheng-21710013-29.png taken!
dataset/Ho Hwa Sheng-21710013-30.png taken!
dataset/Ho Hwa Sheng-21710013-31.png taken!
dataset/Ho Hwa Sheng-21710013-32.png taken!
dataset/Ho Hwa Sheng-21710013-33.png taken!
dataset/Ho Hwa Sheng-21710013-34.png taken!
dataset/Ho Hwa Sheng-21710013-35.png taken!
dataset/Ho Hwa Sheng-21710013-36.png taken!
dataset/Ho Hwa Sheng-21710013-37.png taken!
dataset/Ho Hwa Sheng-21710013-38.png taken!
dataset/Ho Hwa Sheng-21710013-39.png taken!
dataset/Ho Hwa Sheng-21710013-40.png taken!
dataset/Ho Hwa Sheng-21710013-41.png taken!
dataset/Ho Hwa Sheng-21710013-42.png taken!
dataset/Ho Hwa Sheng-21710013-43.png taken!
dataset/Ho Hwa Sheng-21710013-44.png taken!
dataset/Ho Hwa Sheng-21710013-45.png taken!
dataset/Ho Hwa Sheng-21710013-46.png taken!
dataset/Ho Hwa Sheng-21710013-47.png taken!
dataset/Ho Hwa Sheng-21710013-48.png taken!
dataset/Ho Hwa Sheng-21710013-49.png taken!
Process finish
[INFO] Data successfully trained.
  MatricNumber           Name           Date/Time Subjectcode
0     21710013  [Ho Hwa Sheng]  12-06-2021  15:29:04   BENT_3733

 Successfully check in!
[INFO] Email sent to  shengdep1a@gmail.com
```

Figure 4.12: Info message when the email sent

Figure 4.12 shown the email received and figure 4.13 shown the detail of attendance sheet which have student's matric number, name, date and time signed in and subject code that attend.



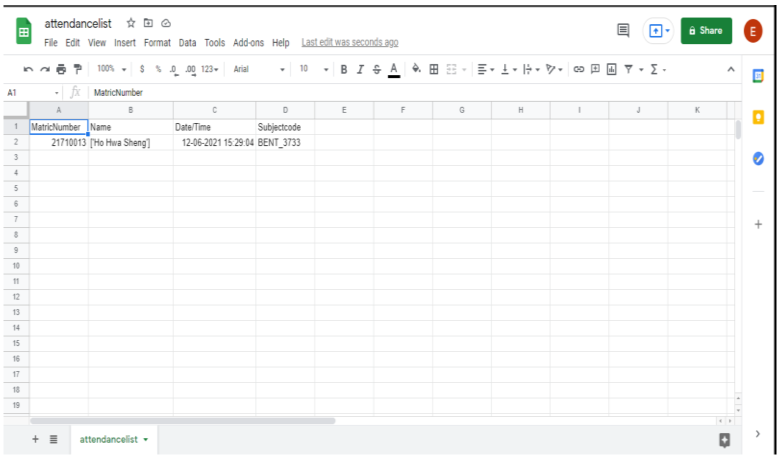
Attendance_list Inbox

face.recog.attendancepsm2021@gmail.com
to me
3:31 PM (4 minutes ago)

Sent from face recognition class attendance system #PSM_2021 #B021710013

attendancelist.csv

Figure 4.13: Email received



attendancelist
File Edit View Insert Format Data Tools Add-ons Help
Last edit was seconds ago
Share

MatricNumber	Name	Date/Time	Subjectcode
21710013	['Ho Hwa Sheng']	12-06-2021 15:29:04	BENT_3733

Figure 4.14: Attendance sheet

4.3 Analysis on confidence measurement

LBPH recognizer calculate the distance between histograms of both input image and trained image. The calculated distance is referring to confidence measurement. According to the OpenCV documents, confidence is parameter used for face recognizer predict function[24]. Confidence also mean distance for the predicted label. Due to the closer distance measured, value of confidence is as lower as better. Ideal value is closer to 0.

The analysis has been carried out on how different settings affected the confidence measurement. These settings are different number of trained image, distance between camera and people to capture and recognize. There are total of 10 subjects which divided into subject A, B, C, D, E wear glasses and subjects F, G, H, I, J not wear glasses. For the number of trained images, it tested with 10, 20, 40, 60, 80 and 100 images while for the distance between camera, with distance of 50cm, 100cm and 200cm. Then, for the value of confidence, take the average value from 3 attempts. The camera module used is 5 megapixels. Other than that, training time for dataset also recorded.

Table 4.1 shown subject A with distance of 50cm. Table 4.2 with distance of 100cm. Table 4.3 with distance of 200cm. Figure 4.15 shown the overall data of subject A. For the subject B, table 4.4 shown with distance of 50cm. Table 4.5 and Table 4.6 shown with distance of 100cm and 200cm respectively. Figure 4.16 shown the overall data of subject B. Next, Table 4.7, Table 4.8 and Table 4.9 displayed the subject C with distance of 50cm, 100cm and 200cm. Figure 4.17 shown the overall data of subject C. Table 4.10, Table 4.11 and Table 4.12 shown the subject D with distance of 50cm, 100cm and 200cm. The overall data of subject D illustrated in figure 4.18. Moreover, Table 4.13, Table 4.14 and Table 4.15 shown the subject E with distance of 50cm, 100cm and 200cm. Figure 4.19 shown overall data of subject E.

For subject F, its data with distance of 50cm, 100cm and 200cm shown in Table 4.16, Table 4.17 and Table 4.18 respectively. Figure 4.20 shown overall data of subject F. Continued with subject G, Table 4.19, Table 4.20 and Table 4.21 shown the subject E with distance of 50cm, 100cm and 200cm. The overall data of subject G illustrated in figure 4.21. Table 4.22 shown subject H with distance of 50cm. Table 4.23 with distance of 100cm. Table 4.24 with distance of 200cm. Figure 4.22 shown overall data of subject H. For subject I, its data with distance of 50cm, 100cm and 200cm displayed in Table 4.25, Table 4.26 and Table 4.27. Overall data of subject I shown in figure 4.23. In addition, Table 4.23, Table 4.29 and Table 4.30 displayed the last subject J with distance of 50cm, 100cm and 200cm respectively. Figure 4.24 shown the overall data of subject J.

Table 4.1: Subject A with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	58.390	46.261	53.820	60.555	47.287	46.509
2nd attempt	58.598	52.938	51.688	46.181	51.770	45.565
3rd attempt	49.336	52.062	50.216	46.681	45.346	43.396
Average	55.441	50.420	51.908	51.139	48.134	45.157
Training time(s)	0.88	0.95	1.67	1.96	3.01	3.77

Table 4.2: Subject A with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	68.957	68.777	58.475	61.421	68.036	64.220
2nd attempt	67.661	63.651	71.905	69.888	64.272	58.811
3rd attempt	66.865	59.544	63.264	65.492	63.212	55.431
Average	67.828	63.990	64.548	65.600	65.173	59.487
Training time(s)	0.64	1.01	1.52	2.03	2.55	3.02

Table 4.3: Subject A with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	83.922	80.175	78.282	78.589	81.113	79.489
2nd attempt	79.809	84.906	82.368	79.782	79.825	83.072
3rd attempt	85.165	83.514	78.531	79.017	80.380	77.257
Average	82.965	82.865	79.727	79.129	80.439	79.939
Training time(s)	0.67	0.93	1.27	1.67	2.06	2.57

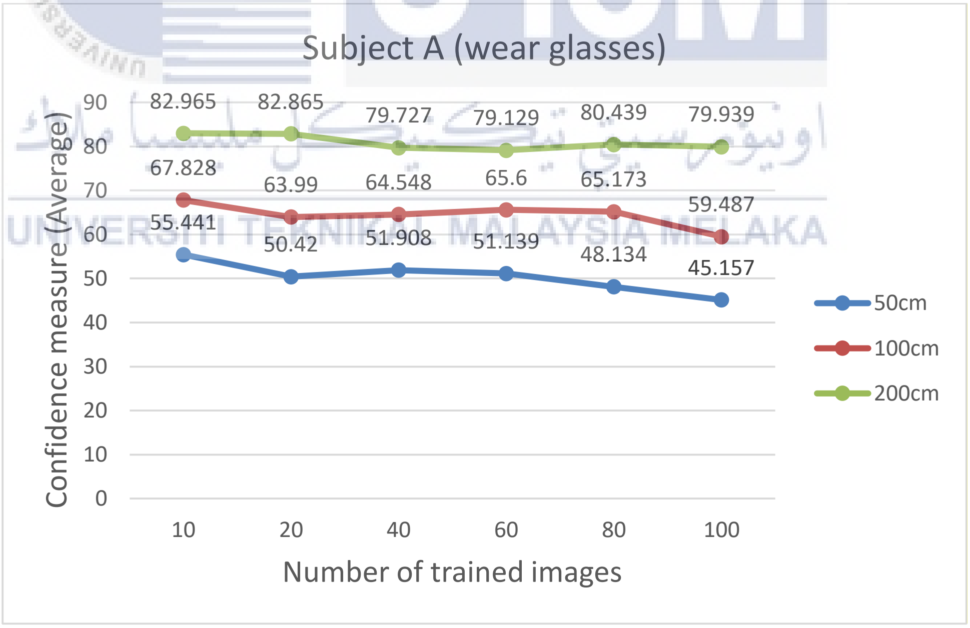


Figure 4.15: Overall data of subject A

Table 4.4: Subject B with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	61.320	51.552	50.621	61.238	44.129	46.708
2nd attempt	56.721	520..3	51.745	47.328	48.355	42.215
3rd attempt	49.215	44.231	50.942	48.631	50.903	41.083
Average	55.752	49.262	51.103	52.399	47.796	43.575
Training time(s)	0.87	0.96	1.67	1.97	3.00	3.79

Table 4.5: Subject B with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	66.034	64.008	65.264	62.007	67.036	57.467
2nd attempt	68.553	67.432	69.834	65.301	62.558	56.067
3rd attempt	67.144	60.784	58.550	69.100	64.990	64.381
Average	67.244	64.075	64.549	65.469	64.861	59.305
Training time(s)	0.65	1.02	1.52	2.05	2.55	3.03

Table 4.6: Subject B with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	84.765	83.120	77.465	80.930	79.468	78.988
2nd attempt	80.157	79.948	82.224	78.064	77.469	82.702
3rd attempt	82.467	83.460	79.551	78.520	78.013	77.752
Average	82.463	82.176	79.747	79.171	78.317	79.814
Training time(s)	0.67	0.94	1.27	1.68	2.08	2.57

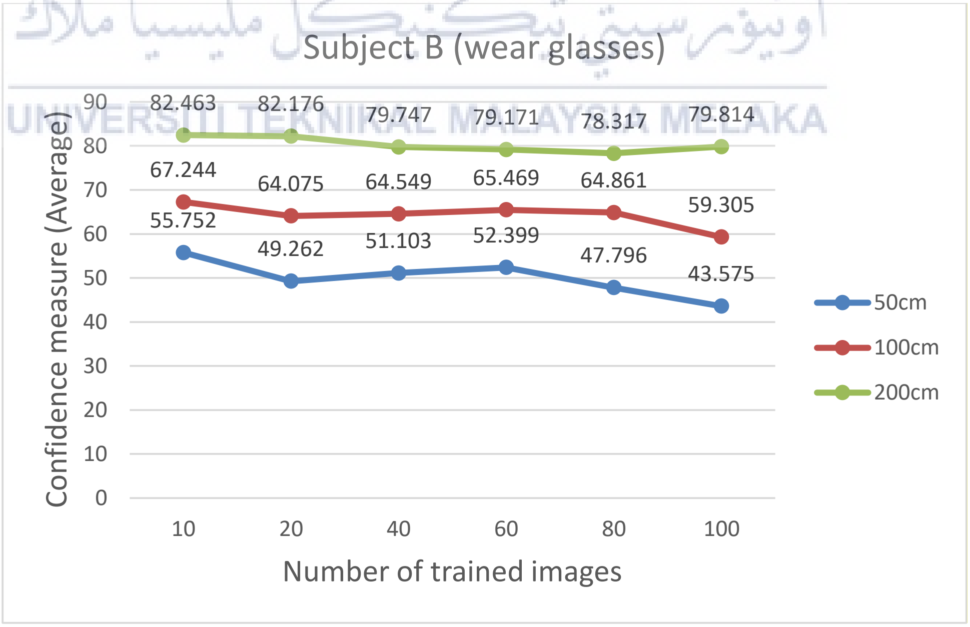


Figure 4.16: Overall data of subject B

Table 4.7: Subject C with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	50.468	45.786	51.426	50.127	50.364	45.037
2nd attempt	54.657	53.003	54.260	46.975	45.762	44.168
3rd attempt	62.275	51.160	50.008	58.168	45.879	44.037
Average	55.8	49.983	51.898	51.757	47.335	44.414
Training time(s)	0.88	0.96	1.67	1.97	3.01	3.77

Table 4.8: Subject C with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	66.058	67.199	68.885	69.664	63.337	57.813
2nd attempt	68.880	60.802	65.106	62.246	68.973	55.374
3rd attempt	65.667	64.381	58.554	66.037	65.204	63.890
Average	66.868	64.127	64.182	65.982	65.838	59.026
Training time(s)	0.65	1.01	1.52	2.03	2.56	3.04

Table 4.9: Subject C with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	84.760	82.417	78.012	78.530	78.132	79.330
2nd attempt	85.134	81.882	78.554	78.113	78.412	76.895
3rd attempt	85.934	84.608	81.997	80.759	79.465	81.430
Average	85.276	82.969	79.521	79.134	78.670	79.218
Training time(s)	0.68	0.94	1.27	1.68	2.07	2.58

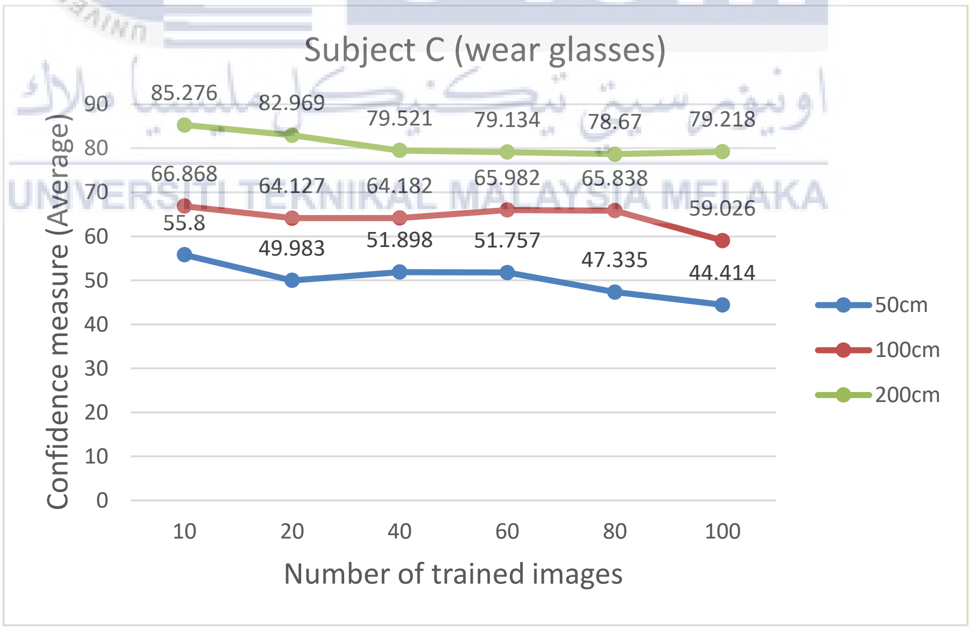


Figure 4.17: Overall data of subject C

Table 4.10: Subject D with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	58.356	47.562	50.986	58.067	46.983	45.976
2nd attempt	59.468	52.348	52.130	47.132	50.883	45.883
3rd attempt	50.994	52.006	51.784	48.586	45.355	40.327
Average	56.273	50.637	51.633	51.261	47.740	44.062
Training time(s)	0.88	0.96	1.68	1.98	3.02	3.78

Table 4.11: Subject D with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	68.948	61.380	69.430	63.039	64.859	55.706
2nd attempt	67.435	60.547	68.265	69.337	63.034	62.174
3rd attempt	67.438	68.005	60.881	60.438	65.271	59.880
Average	67.940	63.311	66.192	64.271	64.388	59.253
Training time(s)	0.64	1.02	1.53	2.04	2.56	3.03

Table 4.12: Subject D with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	86.021	81.005	81.776	80.346	79.558	83.573
2nd attempt	79.324	80.730	80.137	78.024	79.468	78.643
3rd attempt	84.796	81.534	79.333	78.080	79.130	79.458
Average	83.380	81.090	80.415	78.817	79.385	80.558
Training time(s)	0.68	0.94	1.28	1.68	2.08	3.00

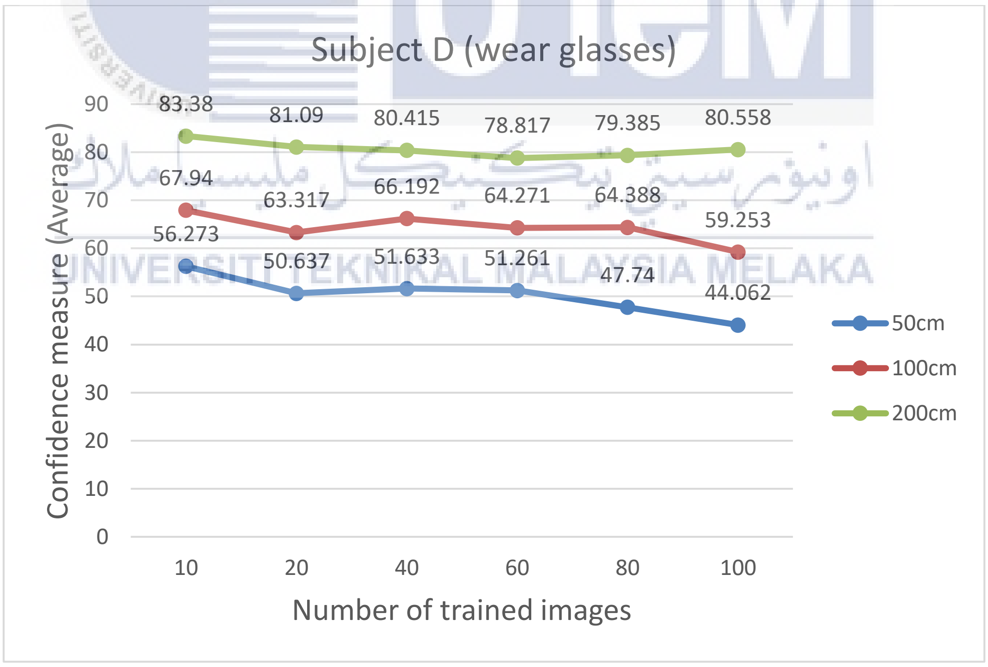


Figure 4.18: Overall data of subject D

Table 4.13: Subject E with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	59.924	44.203	51.994	48.135	51.680	45.199
2nd attempt	49.580	52.934	50.831	49.673	45.550	44.516
3rd attempt	56.542	52.711	51.887	57.003	47.009	45.003
Average	55.349	49.949	51.504	51.604	48.080	44.906
Training time(s)	0.88	0.96	1.67	1.96	3.02	3.78

Table 4.14: Subject E with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	67.144	61.437	67.958	61.247	63.803	62.467
2nd attempt	66.785	60.471	60.457	68.733	68.049	52.820
3rd attempt	67.301	68.221	65.276	66.105	65.766	55.467
Average	67.077	63.376	64.564	65.362	65.873	58.918
Training time(s)	0.65	1.01	1.53	2.04	2.56	3.02

Table 4.15: Subject E with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	79.468	80.367	81.666	79.824	78.310	77.964
2nd attempt	85.157	78.223	79.988	78.369	81.002	78.752
3rd attempt	81.001	85.110	81.218	77.148	79.446	82.006
Average	81.875	81.233	80.957	78.447	79.586	79.574
Training time(s)	0.68	0.92	1.28	1.68	2.06	2.58

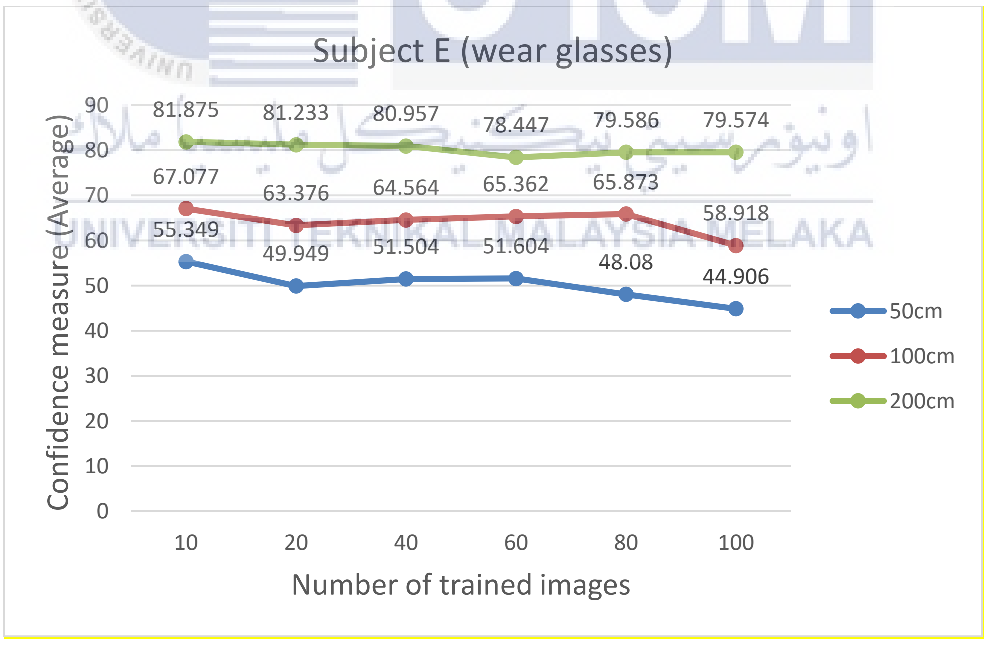


Figure 4.19: Overall data of subject E

Table 4.16: Subject F with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	46.678	41.581	39.886	39.346	40.451	37.322
2nd attempt	47.390	41.710	39.659	40.812	39.300	38.794
3rd attempt	43.582	39.045	41.160	39.402	40.143	37.650
Average	45.883	40.778	39.901	39.853	39.964	37.922
Training time(s)	0.64	1.07	1.92	2.55	3.31	4.05

Table 4.17: Subject F with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	79.977	62.958	61.286	68.504	59.924	61.389
2nd attempt	77.015	62.620	62.245	67.604	61.969	58.590
3rd attempt	61.616	62.494	59.786	64.997	60.771	63.040
Average	72.869	62.691	61.106	67.035	60.888	61.006
Training time(s)	0.58	0.69	1.41	1.97	2.32	2.89

Table 4.18: Subject F with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	113.932	106.377	79.645	87.961	83.386	82.941
2nd attempt	107.807	89.636	85.938	90.564	84.324	85.690
3rd attempt	84.072	95.639	84.266	92.567	80.523	80.042
Average	101.937	97.217	83.283	90.364	82.744	82.891
Training time(s)	0.67	0.87	1.23	1.66	2.15	2.49

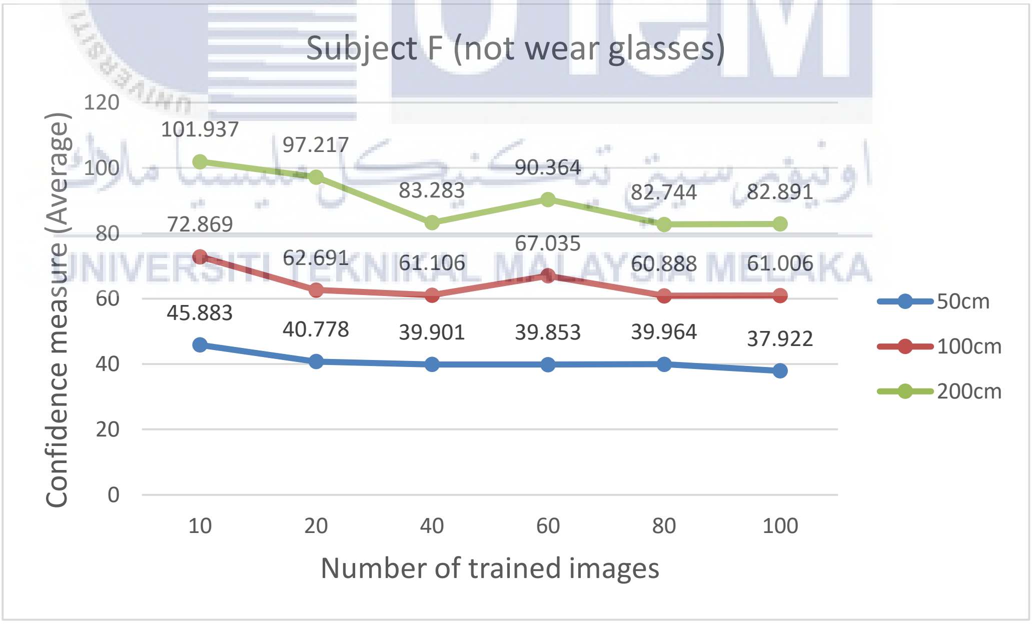


Figure 4.20: Overall data of subject F

Table 4.19: Subject G with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	47.136	40.981	39.167	40.167	39.876	37.448
2nd attempt	44.135	41.773	40.800	40.138	41.037	38.230
3rd attempt	46.078	39.258	41.670	38.699	39.224	37.596
Average	45.883	40.671	40.546	39.668	40.046	37.758
Training time(s)	0.64	1.08	1.92	2.56	3.32	4.07

Table 4.20: Subject G with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	65.942	62.784	62.334	68.243	59.321	63.754
2nd attempt	78.945	62.134	60.256	64.758	60.336	62.732
3rd attempt	79.775	63.002	59.763	66.325	60.590	62.413
Average	74.887	62.64	60.784	66.442	60.082	62.966
Training time(s)	0.58	0.69	1.42	1.98	2.33	2.90

Table 4.21: Subject G with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	106.334	100.684	83.657	90.465	81.346	85.558
2nd attempt	111.678	90.331	80.761	88.619	81.400	81.673
3rd attempt	98.667	95.467	85.205	89.673	84.367	85.523
Average	105.560	95.494	83.208	89.586	82.371	84.251
Training time(s)	0.68	0.88	1.24	1.66	2.15	2.51

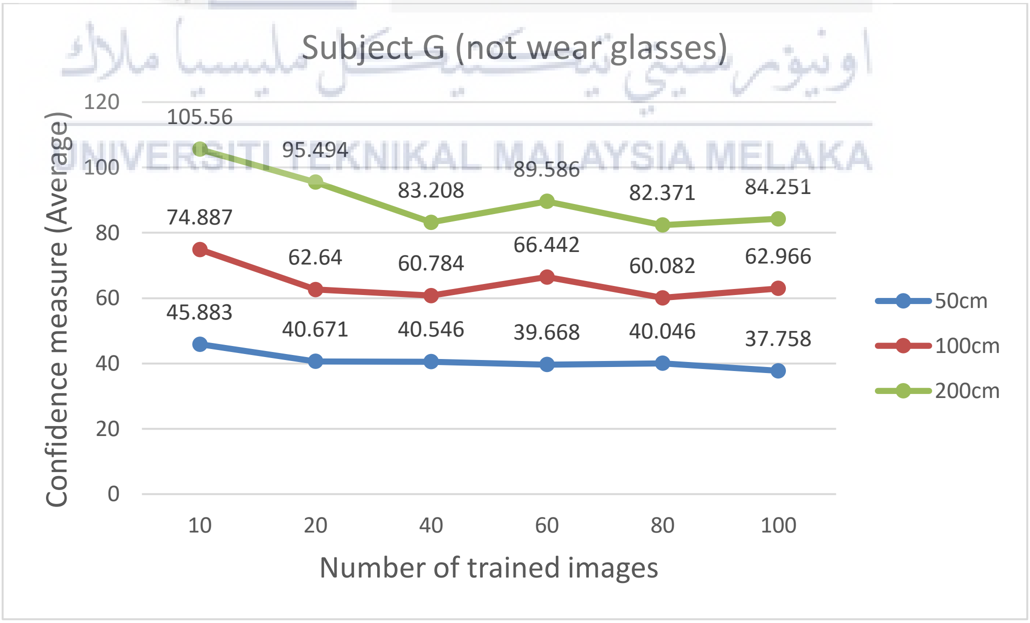


Figure 4.21: Overall data of subject G

Table 4.22: Subject H with distance of 50cm

Confidence value \ Number of trained images	10	20	40	60	80	100
1st attempt	45.376	40.367	38.009	38.201	39.778	38.556
2nd attempt	48.320	40.377	38.164	39.637	40.372	37.886
3rd attempt	47.930	41.850	41.130	41.325	39.658	37.342
Average	47.209	40.865	39.101	39.721	39.936	37.928
Training time(s)	0.63	1.06	1.92	2.55	3.32	4.07

Table 4.23: Subject H with distance of 100cm

Confidence value \ Number of trained images	10	20	40	60	80	100
1st attempt	78.779	62.887	61.668	66.332	61.430	57.980
2nd attempt	62.900	61.330	61.866	68.405	60.791	63.558
3rd attempt	75.510	62.681	60.542	66.664	60.437	62.467
Average	72.396	62.300	61.359	67.134	60.886	61.335
Training time(s)	0.58	0.69	1.42	1.97	2.32	2.89

Table 4.24: Subject H with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	111.002	105.764	84.302	88.572	81.430	81.778
2nd attempt	86.324	90.663	80.663	91.664	82.300	82.337
3rd attempt	112.734	95.144	81.224	90.231	80.453	84.961
Average	103.353	97.190	82.063	90.156	81.394	83.025
Training time(s)	0.68	0.87	1.23	1.66	2.16	2.50

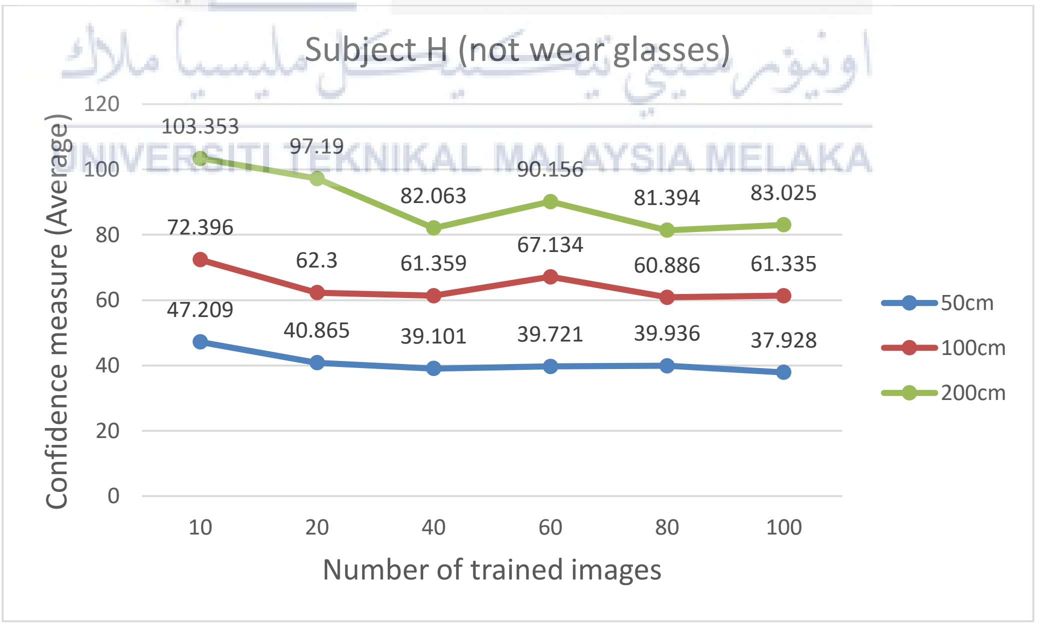


Figure 4.22: Overall data of subject H

Table 4.25: Subject I with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	46.008	41.883	39.556	39.415	39.776	37.160
2nd attempt	45.370	40.796	39.695	41.563	39.428	36.866
3rd attempt	46.573	40.273	38.134	39.638	41.302	37.460
Average	45.984	40.984	39.128	40.205	40.169	37.162
Training time(s)	0.64	1.08	1.92	2.56	3.31	4.06

Table 4.26: Subject I with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	77.462	62.260	60.543	67.352	58.466	59.950
2nd attempt	76.164	63.889	59.556	65.870	60.354	63.112
3rd attempt	79.201	61.437	61.667	67.456	60.399	59.673
Average	77.609	62.529	60.589	66.893	59.740	60.912
Training time(s)	0.59	0.69	1.42	1.98	2.32	2.90

Table 4.27: Subject I with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	108.764	90.663	84.422	91.672	80.060	83.753
2nd attempt	110.127	102.325	85.004	88.956	83.371	85.680
3rd attempt	112.467	89.773	80.365	90.988	83.127	81.035
Average	110.453	94.254	83.264	90.539	82.186	83.489
Training time(s)	0.68	0.87	1.24	1.66	2.15	2.49

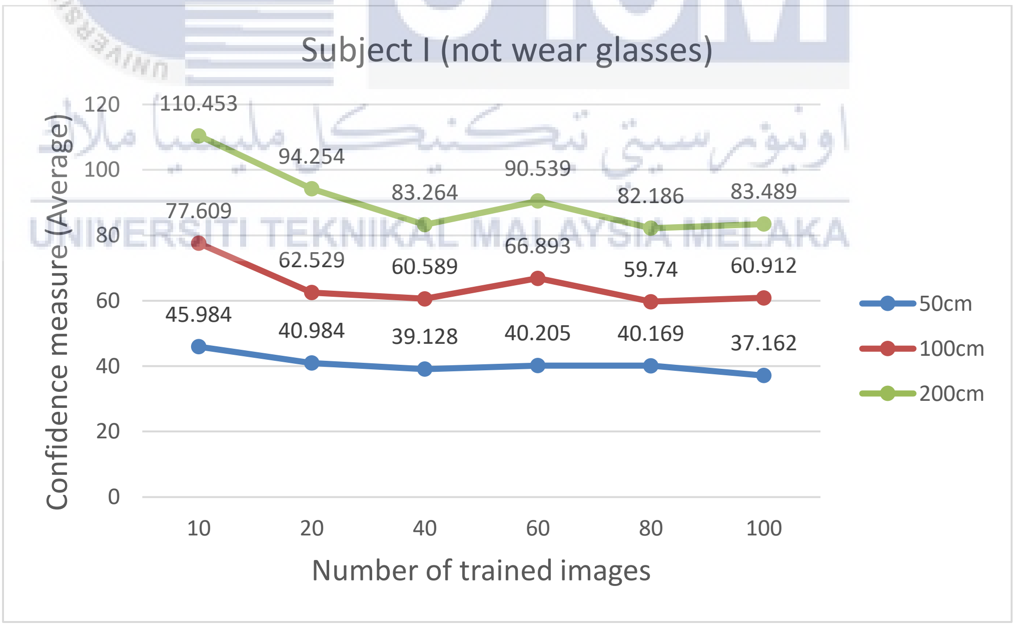


Figure 4.23: Overall data of subject I

Table 4.28: Subject J with distance of 50cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	47.300	39.520	39.389	39.664	40.243	37.550
2nd attempt	44.721	41.333	40.768	38.761	40.733	36.579
3rd attempt	48.315	39.875	40.307	41.532	38.990	37.111
Average	46.779	40.243	40.155	39.986	39.989	37.08
Training time(s)	0.63	1.06	1.92	2.56	3.31	4.05

Table 4.29: Subject J with distance of 100cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	78.333	63.253	62.520	65.764	60.221	60.305
2nd attempt	76.896	61.467	61.887	67.332	59.370	59.763
3rd attempt	80.044	62.548	58.981	67.320	61.699	63.458
Average	78.424	62.423	61.129	66.805	60.43	61.175
Training time(s)	0.58	0.70	1.42	1.97	2.32	2.90

Table 4.30: Subject J with distance of 200cm

Number of trained images / Confidence value	10	20	40	60	80	100
1st attempt	100.270	94.936	84.673	91.273	83.570	84.204
2nd attempt	109.708	105.473	80.556	88.691	82.004	83.240
3rd attempt	112.880	90.005	86.021	90.673	83.024	81.933
Average	107.619	96.805	83.75	90.212	82.866	83.126
Training time(s)	0.67	0.87	1.23	1.64	2.15	2.50

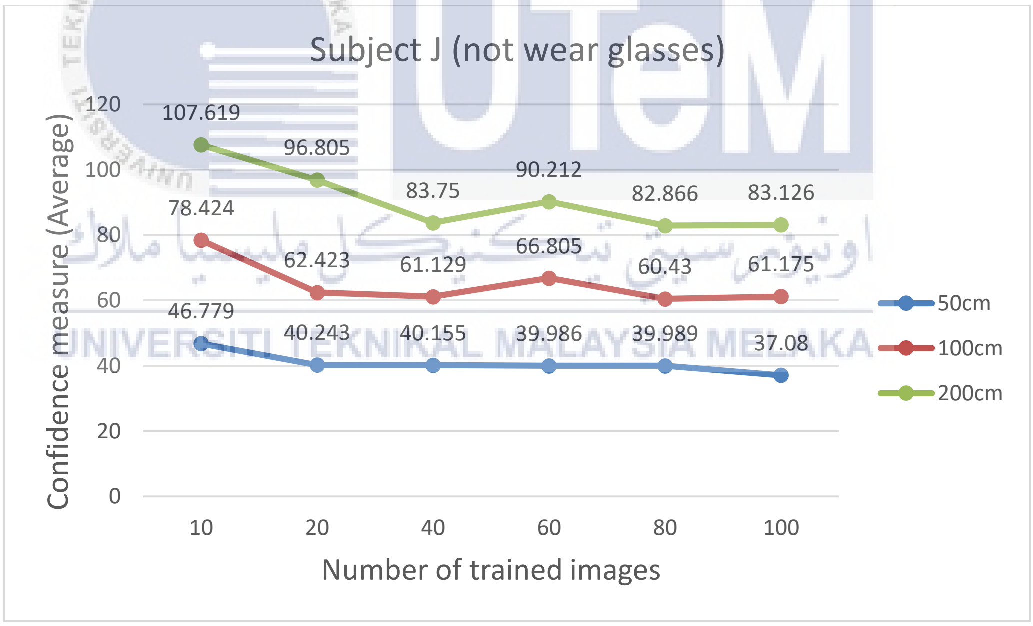


Figure 4.24: Overall data of subject J

Based on figure 4.15, it can observed that lower confidence obtained is 45.157 with 100 trained images and higher confidence is 55.441 with 10 trained images within distance of 50cm. For the distance of 100cm, lower confidence is 59.487 with 100 trained images and higher confidence is 67.828 with 10 trained images. Then, lower confidence is 79.129 with 60 trained images and higher confidence is 82.965 with 10 trained images within distance of 200cm. For subject A, all the lower confidence measurement achieved with 60 and 100 trained images. Based on figure 4.16, the lower confidence is 43.575 and higher confidence is 55.752 in distance of 50cm. For 100cm, the lower confidence is 59.305 and higher confidence is 67.244. For 200cm, the lower confidence is 78.317 and higher confidence is 82.463. All the lower confidence measurement of subject B achieved with 80 and 100 trained images. Next, the lower confidence of subject C in distance of 50cm is 44.414 and higher value is 55.8. For distance of 100cm, lower value is 59.026 and higher value is 66.868. For distance of 200cm, lower value is 78.67 and higher value is 85.276. Based on figure, 4.18, lower confidence value are 44.062, 59.253 and 78.817 while higher value are 56.273, 67.94 and 83.38 for distance of 50cm, 100cm and 200cm.With the trained images between 60 to 100 images, it obtained lower confidence. For subject E, the lower confidence value are 44.906, 58.918 and 78.447 while higher value are 55.349, 67.077 and 81.875 for distance of 50cm, 100cm and 200cm.

Furthermore, analysis with subject F. By referring to figure 4.20, lower confidence is 37.922 with 100 trained images and higher confidence is 45.883 with 10 trained images for distance of 50cm. For the distance of 100cm, lower confidence is 60.888 with 80 trained images and higher confidence is 72.869 with 10 trained images. Then, lower confidence is 82.744 with 80 trained images while higher confidence is 101.937 with 10 trained images for distance of 200cm. The lower confidence measurement achieved with 80 and 100 trained images for subject F. For subject G, the lower confidence value are 37.758, 60.082 and 82.371 while higher value are 45.883, 74.887 and 105.56 for distance of 50cm, 100cm and 200cm. All the lower confidence obtained with trained images of 80 and 100 images and higher confidence with 10 images. After that, subject H also obtained lower confidence with trained images of 80 and 100 images. These values are 37.928, 60.886 and 81.394 while for higher value are 47.209, 77.396 and 103.353 for distance of 50cm, 100cm and 200cm. Based on figure 4.23, the lower confidence value are 37.162, 59.74 and 82.186 while higher value are 45.984, 77.609 and 110.453 for distance of 50cm, 100cm and 200cm. For subject J, the lower confidence value are 37.08, 60.43 and 82.866 while higher value are 46.779, 78.424 and 107.619 for distance of 50cm, 100cm and 200cm. The higher confidence obtained with 10 trained images.

As a conclusion, it can observe that as the distance increase, confidence measurement also increase for any size of trained images. Thus, distance to train and recognizer is one of the factor that can affect the confidence measurement. Moreover, number of trained images should be set in range of 60 to 100 images in order to achieve the lower confidence measurement for distance from 50cm to 200cm. In contrast, having the less number of trained images for example 10 images, much higher the confidence measurement will obtained. However, lightning conditions or illumination inside room, face expression, resolution of camera that used and etc also affect the measurement. These data that obtained just as a reference. Another observation is the difference between higher confidence and lower confidence in certain distance for both types of subjects is opposite as distance increased. Let expressed in value for subject A that wear glasses and subject F that not wear glasses. The value for subject A in distance of 50cm, 100cm, 200cm are 10.284, 8.341 and 3.836 respectively. Then, for subject F in distance of 50cm, 100cm, 200cm are 7.961, 11.981 and 19.193 respectively. Therefore as distance increase, the difference between higher confidence and lower confidence become more smaller for subject A. For subject F, the difference become larger as distance increase. Moreover, the training process of algorithm is considering fast enough. It takes less than 5 seconds to train the 100 trained images.

4.4 Discussion on results and system

During the development, coding is an essential part to build up this system. Therefore, learn the python library and function is a must process. Most of the time will spend on testing and debugging code. The part on the automatically update user's detail on the attendance list after being recognized are one of the problems faced during development. However, this problem has been solved by used pandas.dataframe function. Email server used by this system is Gmail but it can change depending on the requirement. The system used Gmail due to its popularity among users. Next, during the camera access may experience a bit of lagging in this system. This may due to the limited processing power on raspberry pi 3. In addition, it also loads the GUI element at the same time. However, it still functional to perform the task given.

4.5 Project Cost

Total project cost is listed in Table 4.31. As seen, the total cost is less than RM300.

Table 4.31: Project cost

Component	Price (RM)
Raspberry Pi 3 model B	155
Adapter cable	20
Micro SD card	46
Pi camera	18
Casing (Camera + Pi)	21
Total	260

4.6 Conclusion

At the end of chapter, results obtained and the system has been discussed. Then, the analysis was also presented and discussed. Project cost also included in this chapter. In the next chapter, the conclusion and any future work /recommendation will be discussed.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

5.1 Conclusion

At the end, the real time face recognition class attendance system has been successfully developed. This system is easy to use and able to speed up the attendance record process. In order to test the functionality, the system need proceed to testing phases. At that time, it able to receive feedback from actual users to help improve the system. Besides that, Raspberry Pi board that used is very cost-effective as it can be used for many other applications. With the affordable price and many features provided, Raspberry Pi can be a microcontroller choice for academic projects as long as it meets the project requirement. Since this system is fully software-based, it considers as paperless system which saving resources and environmentally friendly through reducing paper consumption. Besides that, Raspberry Pi is an energy-efficient device as it only consumes very low-voltage power supply to operate. Lastly, project objectives have also been achieved.

5.2 Future works

As for future development, there are few improvements to the system. First, change the local database to online database. It comes with advantages such as more security, more efficient in accessing data and etc. This also consider that increase of user in future time as storage in Raspberry Pi will not sufficient. Other than that, the system can be use as attendance system in other fields such as the workplace. If then, the system need to improve and add more features that based on working purpose such as leave requests, shift scheduling and etc. Furthermore, can upgrade the equipment used like Raspberry Pi and camera module in future time. This improve efficiency for attendance taking process with updated equipment.

REFERENCES

[1] A. Lukkarinen, P. Koivukangas, and T. Seppälä, “Relationship between Class Attendance and Student Performance,” *Procedia - Soc. Behav. Sci.*, vol. 228, pp. 341–347, 2016, doi: 10.1016/j.sbspro.2016.07.051.

[2] K. L. Sudha, S. Shinde, T. Thomas, and A. Abdugani, “Barcode based Student Attendance System,” 2015.

[3] M. H. M. Baban, “Attendance Checking System Using Quick Response Code For Students At The University Of Sulaimaniyah,” *J. Math. Comput. Sci.*, vol. 10, no. 03, pp. 189–198, 2014, doi: 10.22436/jmcs.010.03.04.

[4] W. B. Al-Kendi and H. H. Al-Nayyef, “Data Management via QR Code Using Android Smart Devices,” *Al-Mustansiriyah J. Sci.*, vol. 31, no. 3, p. 95, Aug. 2020, doi: 10.23851/mjs.v31i3.853.

[5] Y. K. Hooi, K. Shafee Kalid, and S. Tachmammedov, “MULTI-FACTOR ATTENDANCE AUTHENTICATION SYSTEM,” *Int. J. Softw. Eng. Comput. Syst.*, vol. 4, no. 2, pp. 62–79, 2018, doi: 10.15282/ijsecs.4.2.2018.5.0049.

[6] R. Lodha, S. Gupta, H. Jain, and H. Narula, “Bluetooth Smart based attendance management system,” in *Procedia Computer Science*, 2015, vol. 45, no. C, pp. 524–527, doi: 10.1016/j.procs.2015.03.094.

[7] S. Honnalli and G. Vara Prasad, “Attendance Monitoring Model for Range Surveillance Using Bluetooth,” *Int. J. Sensors Sens. Networks*, vol. 7, no. 4, p. 56, 2019, doi: 10.11648/j.ijssn.20190704.12.

[8] H. El Mrabet and A. A. Moussa, “IoT-school attendance system using RFID technology,” *Int. J. Interact. Mob. Technol.*, vol. 14, no. 14, pp. 95–108, 2020, doi: 10.3991/IJIM.V14I14.14625.

[9] “Smartphone Popularity -- Security Today.” https://securitytoday.com/Articles/2018/05/01/Smartphone-

Popularity.aspx?Page=1 (accessed Dec. 19, 2020).

[10] A. Adamu, "ATTENDANCE MANAGEMENT… ATTENDANCE MANAGEMENT SYSTEM USING FINGERPRINT AND IRIS BIOMETRIC," 2019. Accessed: Dec. 20, 2020. [Online]. Available: https://www.researchgate.net/publication/338536515.

[11] O. O. Shoewu, L. A. Akinyemi, R. A. Lawal, and O. R. Otagburuagu, "Enhanced Smart Biometric Based Attendance (ES2BASYS) System Interfaced with POS Facility for a Smart Academic Institution," 2020. Accessed: Dec. 20, 2020. [Online]. Available: http://www.akamaiuniversity.us/PJST.htm.

[12] D. M. V. Salac, "PRESENT: An Android-Based Class Attendance Monitoring System Using Face Recognition Technology," *Int. J. Comput. Sci. Res.*, vol. 2, no. 3, pp. 102–115, 2018, doi: 10.25147/ijcsr.2017.001.1.28.

[13] N. Nordin and N. H. M. Fauzi, "A web-based mobile attendance system with facial recognition feature," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 5, pp. 193–202, 2020, doi: 10.3991/IJIM.V14I05.13311.

[14] D. Prangchumpol, "Face Recognition for Attendance Management System Using Multiple Sensors," in *Journal of Physics: Conference Series*, 2019, vol. 1335, no. 1, p. 12011, doi: 10.1088/1742-6596/1335/1/012011.

[15] K. Sai, P. Reddy, and K. N. Raju, "Design and Implementation of an Algorithm for Face Recognition by using Principal Component Analysis (PCA) in MATLAB," 2016. Accessed: Dec. 20, 2020. [Online]. Available: www.ijarcsse.com.

[16] F. Almudhafer Abd, Z. A. Khalaf -, M. Azriansyah, N. Hartuti, M. Fachrurrozi, and B. Adhi Tama, "A Study about Principle Component Analysis and Eigenface for Facial Extraction Recent citations Review of Different Combinations of Facial Expression Recognition System A Study about Principle Component Analysis and Eigenface for Facial Extraction," doi: 10.1088/1742-6596/1196/1/012010.

[17] P. Mehta and P. Tomar, "An Efficient Attendance Management Sytem based on Face Recognition using Matlab and Raspberry Pi 2," *Int. J. Eng. Technol. Sci. Res.*, vol. 3, no. 5, pp. 2394–3386, 2016, Accessed: Dec. 20, 2020. [Online]. Available: www.ijetsr.com.

[18] J.-L. Aufanc, "Raspberry Pi 3 Board is Powered by Broadcom BCM2837 Cortex A53 Processor, Sells for $35," 2016. https://www.cnx-software.com/2016/02/29/raspberry-pi-3-board-is-powered-by-broadcom-bcm2827-cortex-a53-processor-sells-for-35/ (accessed Dec. 23, 2020).

[19] "Download NOOBS for Raspberry Pi." https://www.raspberrypi.org/downloads/noobs/ (accessed Dec. 23, 2020).

[20] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, doi: 10.1109/cvpr.2001.990517.

[21] “OpenCV: Cascade Classifier.” https://docs.opencv.org/master/db/d28/tutorial_cascade_classifier.html (accessed Jun. 13, 2021).

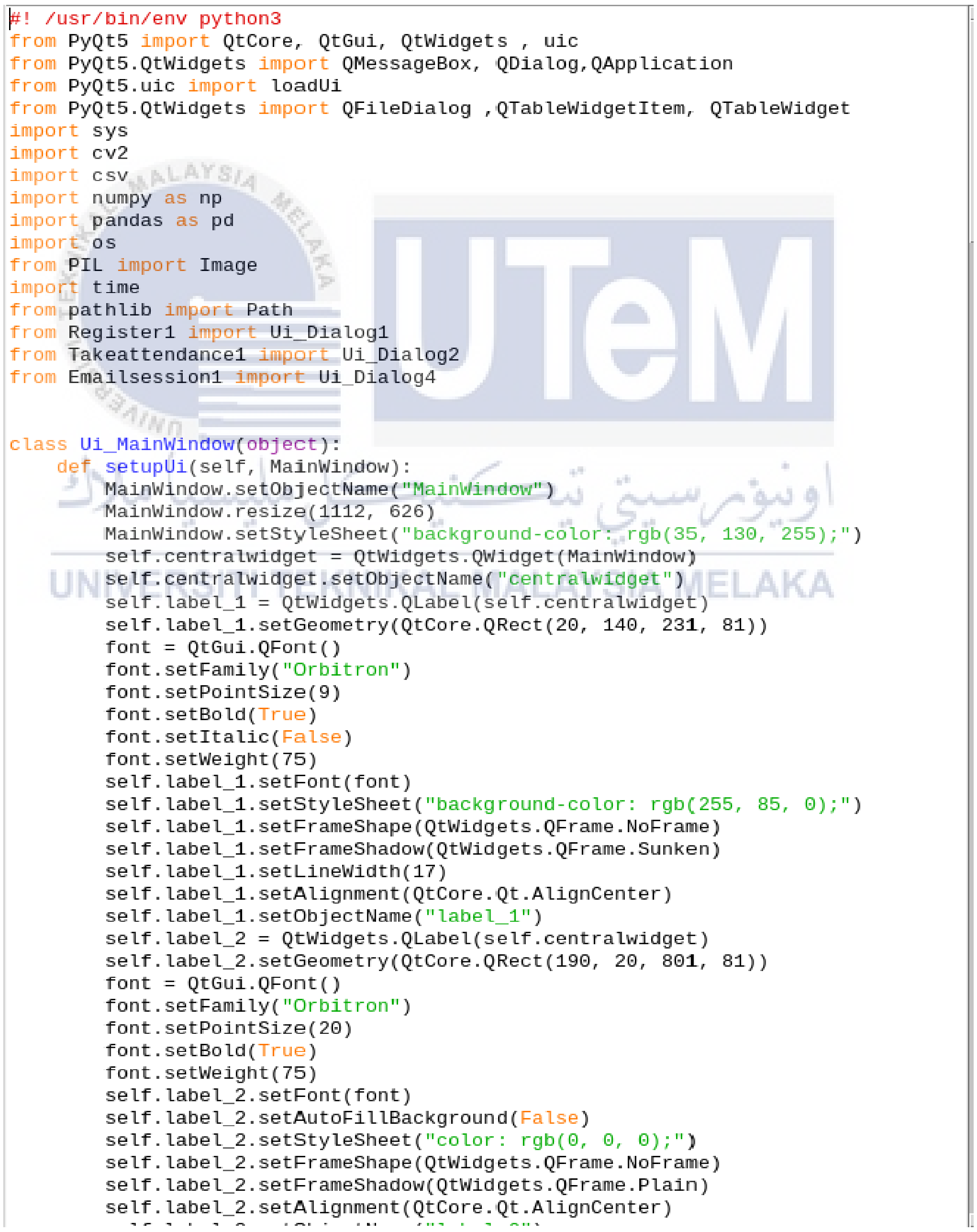
[22] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002, doi: 10.1109/TPAMI.2002.1017623.

[23] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3021, pp. 469–481, 2004, doi: 10.1007/978-3-540-24670-1_36.

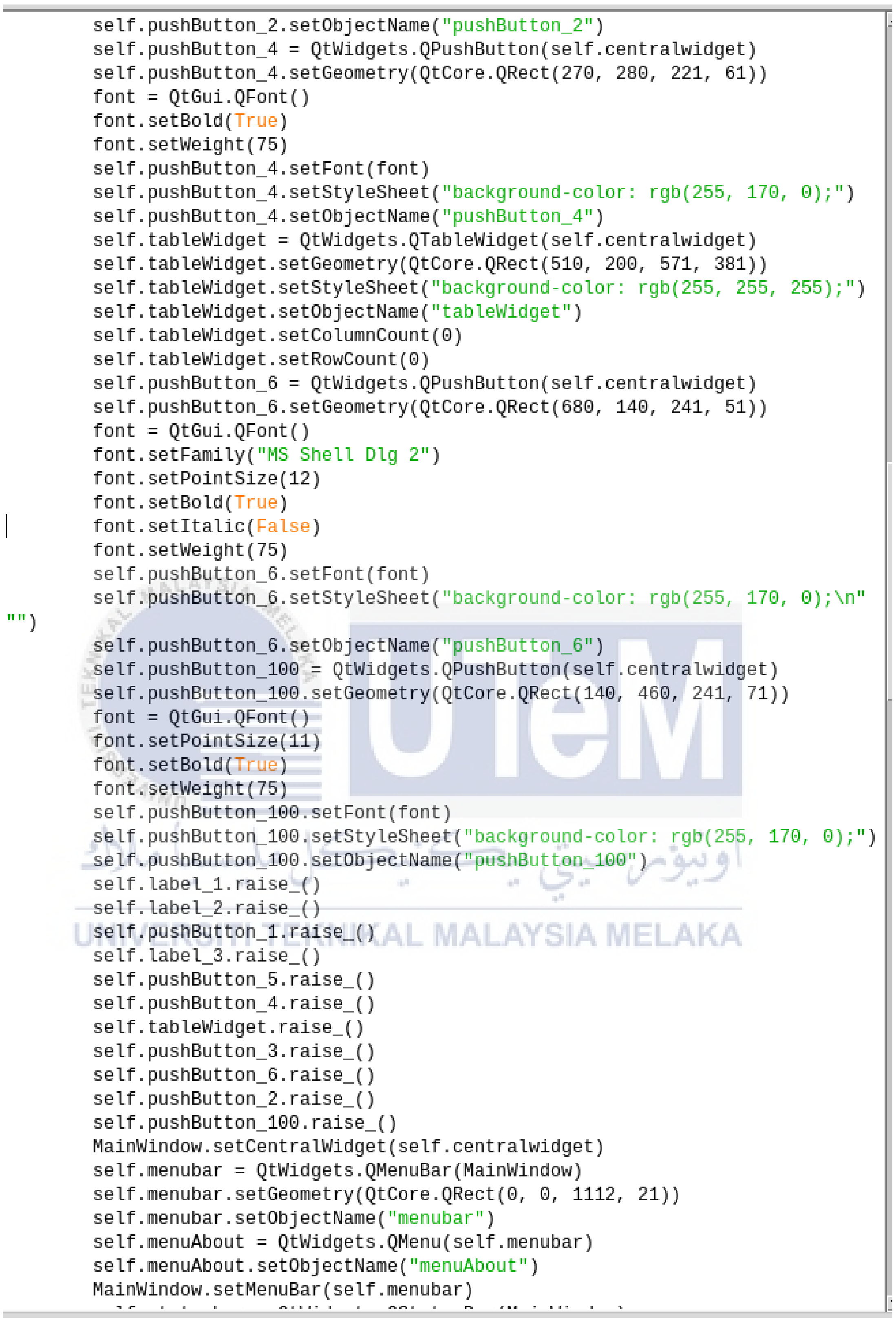
[24] “OpenCV: cv::face::FaceRecognizer Class Reference.” https://docs.opencv.org/master/dd/d65/classcv_1_1face_1_1FaceRecognizer.html#ab0d593e53ebd9a0f350c989fcac7f251 (accessed Jun. 13, 2021).

APPENDICES

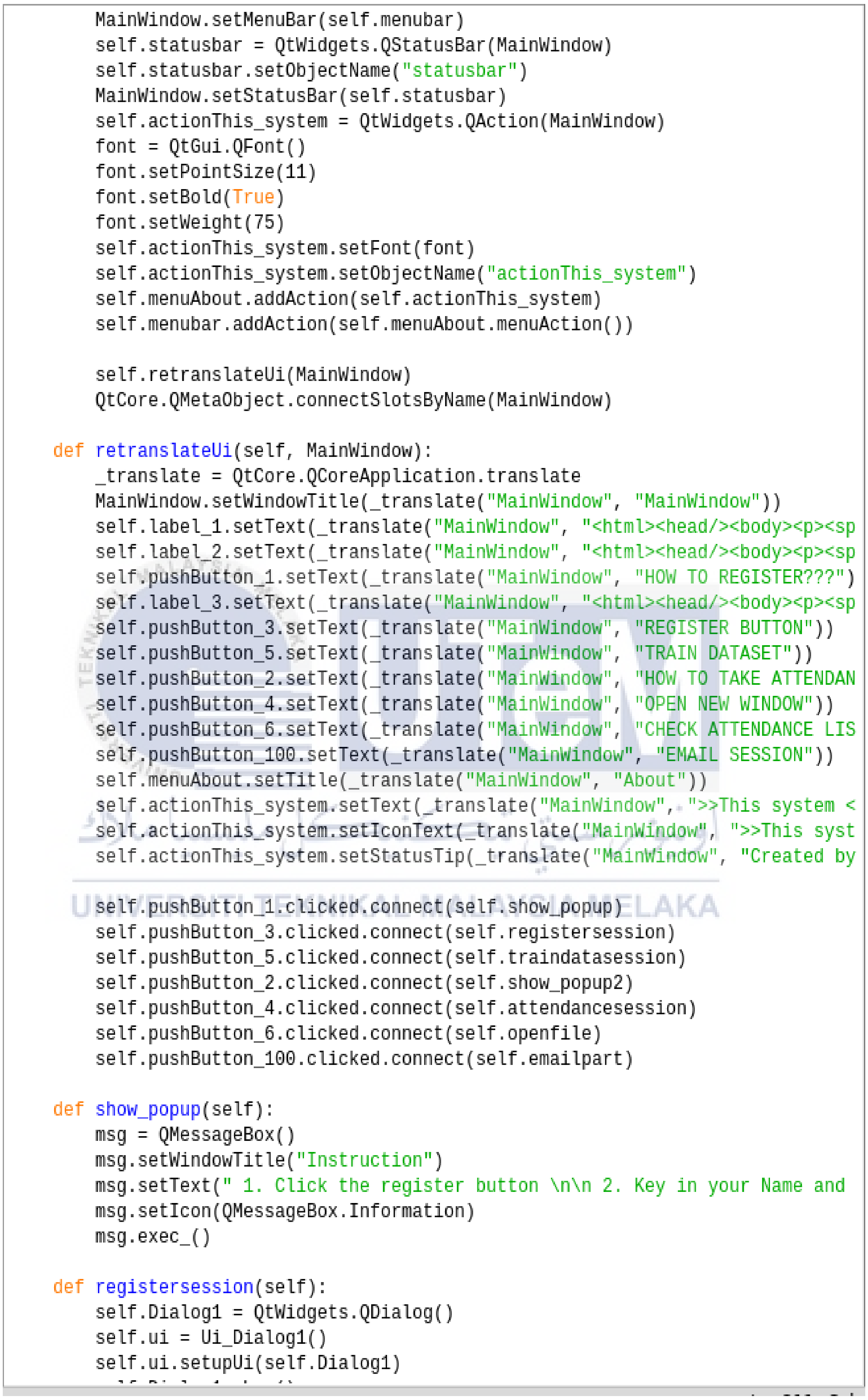
Appendix A : Mainwindow.py



```
#! /usr/bin/env python3
from PyQt5 import QtCore, QtGui, QtWidgets , uic
from PyQt5.QtWidgets import QMessageBox, QDialog,QApplication
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QFileDialog ,QTableWidgetItem, QTableWidget
import sys
import cv2
import csv
import numpy as np
import pandas as pd
import os
from PIL import Image
import time
from pathlib import Path
from Register1 import Ui_Dialog1
from Takeattendance1 import Ui_Dialog2
from Emailsession1 import Ui_Dialog4


class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1112, 626)
        MainWindow.setStyleSheet("background-color: rgb(35, 130, 255);")
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.label_1 = QtWidgets.QLabel(self.centralwidget)
        self.label_1.setGeometry(QtCore.QRect(20, 140, 231, 81))
        font = QtGui.QFont()
        font.setFamily("Orbitron")
        font.setPointSize(9)
        font.setBold(True)
        font.setItalic(False)
        font.setWeight(75)
        self.label_1.setFont(font)
        self.label_1.setStyleSheet("background-color: rgb(255, 85, 0);")
        self.label_1.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.label_1.setFrameShadow(QtWidgets.QFrame.Sunken)
        self.label_1.setLineWidth(17)
        self.label_1.setAlignment(QtCore.Qt.AlignCenter)
        self.label_1.setObjectName("label_1")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(190, 20, 801, 81))
        font = QtGui.QFont()
        font.setFamily("Orbitron")
        font.setPointSize(20)
        font.setBold(True)
        font.setWeight(75)
        self.label_2.setFont(font)
        self.label_2.setAutoFillBackground(False)
        self.label_2.setStyleSheet("color: rgb(0, 0, 0);")
        self.label_2.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.label_2.setFrameShadow(QtWidgets.QFrame.Plain)
        self.label_2.setAlignment(QtCore.Qt.AlignCenter)
        self.label_2.setAlignment(QtCore.Qt.AlignCenter)
        self.label_2.setObjectName("label_2")
        self.pushButton_1 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_1.setGeometry(QtCore.QRect(20, 220, 231, 61))
        font = QtGui.QFont()
        font.setFamily("MS Shell Dlg 2")
        font.setPointSize(12)
        font.setBold(True)
        font.setItalic(False)
        font.setWeight(75)
        self.pushButton_1.setFont(font)
        self.pushButton_1.setStyleSheet("background-color: rgb(255, 170, 0);\n"
"")
        self.pushButton_1.setObjectName("pushButton_1")
        self.label_3 = QtWidgets.QLabel(self.centralwidget)
        self.label_3.setGeometry(QtCore.QRect(270, 140, 221, 81))
        font = QtGui.QFont()
        font.setFamily("Orbitron")
        font.setPointSize(9)
        font.setBold(True)
        font.setItalic(False)
        font.setWeight(75)
        self.label_3.setFont(font)
        self.label_3.setStyleSheet("background-color: rgb(255, 85, 0);")
        self.label_3.setFrameShape(QtWidgets.QFrame.NoFrame)
        self.label_3.setFrameShadow(QtWidgets.QFrame.Sunken)
        self.label_3.setLineWidth(17)
        self.label_3.setAlignment(QtCore.Qt.AlignCenter)
        self.label_3.setObjectName("label_3")
        self.pushButton_3 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_3.setGeometry(QtCore.QRect(20, 280, 231, 61))
        font = QtGui.QFont()
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_3.setFont(font)
        self.pushButton_3.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_3.setObjectName("pushButton_3")
        self.pushButton_5 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_5.setGeometry(QtCore.QRect(20, 340, 231, 61))
        font = QtGui.QFont()
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_5.setFont(font)
        self.pushButton_5.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_5.setObjectName("pushButton_5")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(270, 220, 221, 61))
        font = QtGui.QFont()
        font.setPointSize(11)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_2.setObjectName("pushButton_2")
```

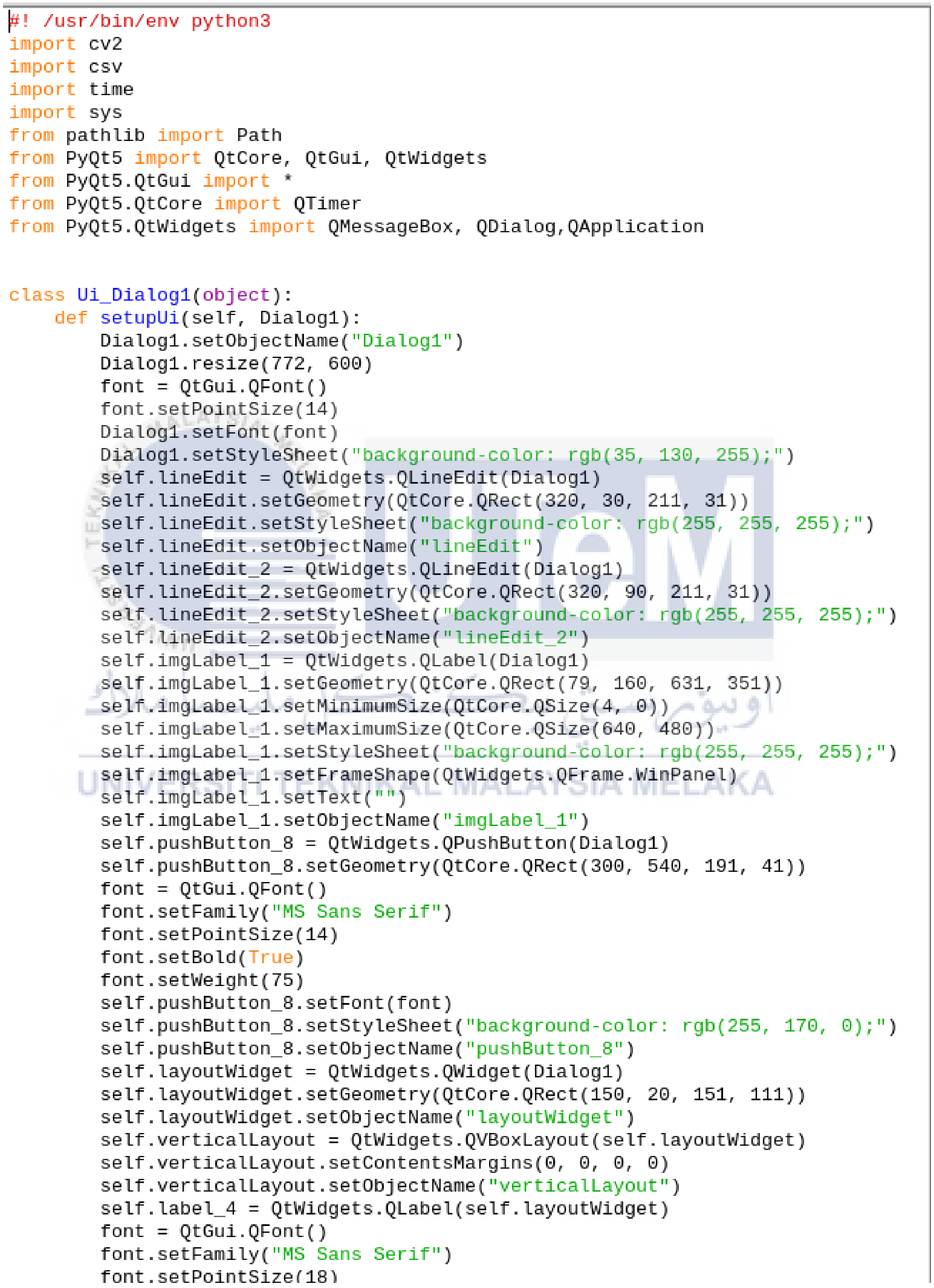


```
        self.pushButton_2.setObjectName("pushButton_2")
        self.pushButton_4 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_4.setGeometry(QtCore.QRect(270, 280, 221, 61))
        font = QtGui.QFont()
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_4.setFont(font)
        self.pushButton_4.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_4.setObjectName("pushButton_4")
        self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)
        self.tableWidget.setGeometry(QtCore.QRect(510, 200, 571, 381))
        self.tableWidget.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.tableWidget.setObjectName("tableWidget")
        self.tableWidget.setColumnCount(0)
        self.tableWidget.setRowCount(0)
        self.pushButton_6 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_6.setGeometry(QtCore.QRect(680, 140, 241, 51))
        font = QtGui.QFont()
        font.setFamily("MS Shell Dlg 2")
        font.setPointSize(12)
        font.setBold(True)
        font.setItalic(False)
        font.setWeight(75)
        self.pushButton_6.setFont(font)
        self.pushButton_6.setStyleSheet("background-color: rgb(255, 170, 0);\n"
"")
        self.pushButton_6.setObjectName("pushButton_6")
        self.pushButton_100 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_100.setGeometry(QtCore.QRect(140, 460, 241, 71))
        font = QtGui.QFont()
        font.setPointSize(11)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_100.setFont(font)
        self.pushButton_100.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_100.setObjectName("pushButton_100")
        self.label_1.raise_()
        self.label_2.raise_()
        self.pushButton_1.raise_()
        self.label_3.raise_()
        self.pushButton_5.raise_()
        self.pushButton_4.raise_()
        self.tableWidget.raise_()
        self.pushButton_3.raise_()
        self.pushButton_6.raise_()
        self.pushButton_2.raise_()
        self.pushButton_100.raise_()
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 1112, 21))
        self.menubar.setObjectName("menubar")
        self.menuAbout = QtWidgets.QMenu(self.menubar)
        self.menuAbout.setObjectName("menuAbout")
        MainWindow.setMenuBar(self.menubar)
```



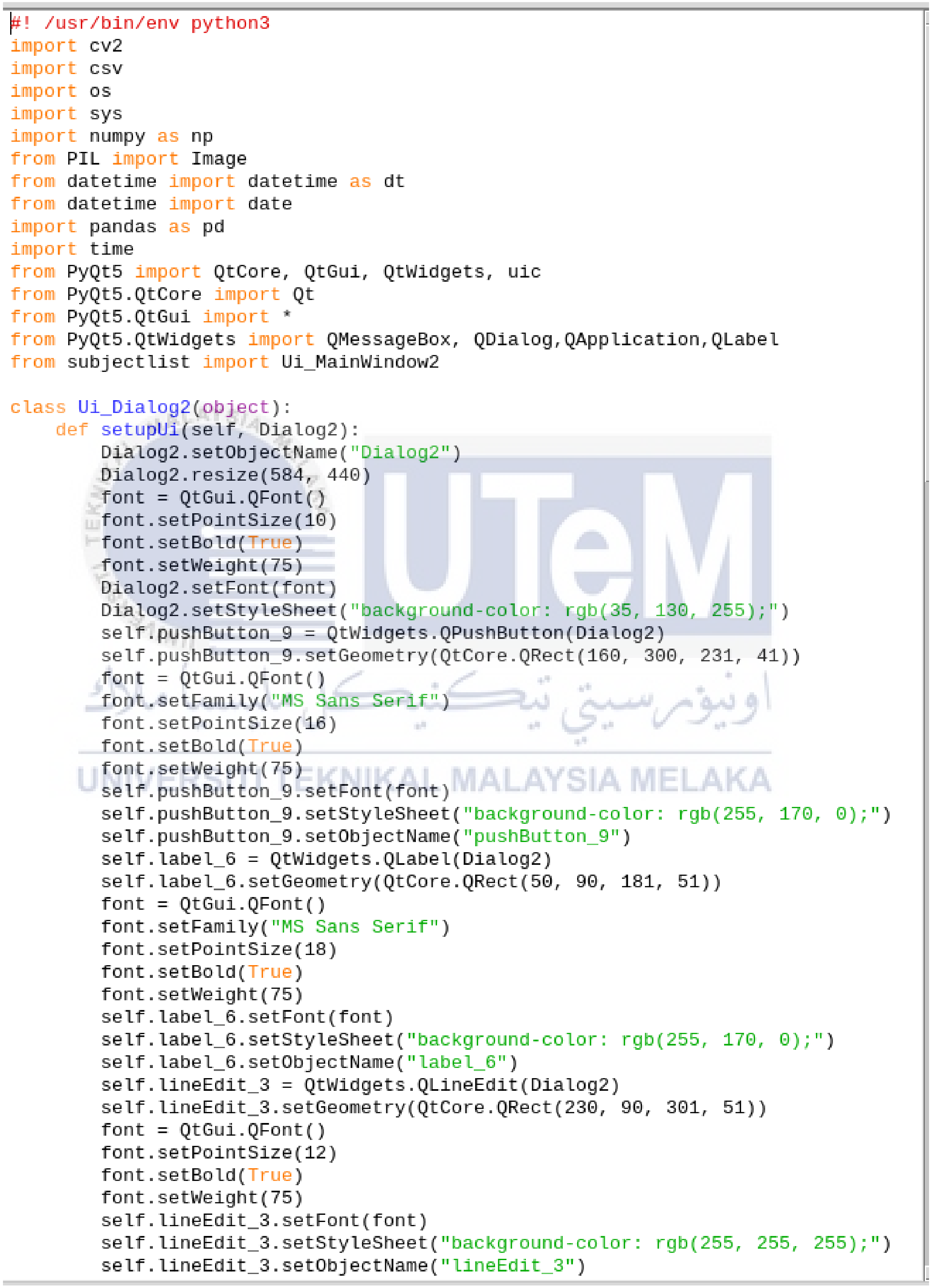
```
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)
        self.actionThis_system = QtWidgets.QAction(MainWindow)
        font = QtGui.QFont()
        font.setPointSize(11)
        font.setBold(True)
        font.setWeight(75)
        self.actionThis_system.setFont(font)
        self.actionThis_system.setObjectName("actionThis_system")
        self.menuAbout.addAction(self.actionThis_system)
        self.menubar.addAction(self.menuAbout.menuAction())

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.label_1.setText(_translate("MainWindow", "<html><head/><body><p><sp
        self.label_2.setText(_translate("MainWindow", "<html><head/><body><p><sp
        self.pushButton_1.setText(_translate("MainWindow", "HOW TO REGISTER???")
        self.label_3.setText(_translate("MainWindow", "<html><head/><body><p><sp
        self.pushButton_3.setText(_translate("MainWindow", "REGISTER BUTTON"))
        self.pushButton_5.setText(_translate("MainWindow", "TRAIN DATASET"))
        self.pushButton_2.setText(_translate("MainWindow", "HOW TO TAKE ATTENDAN
        self.pushButton_4.setText(_translate("MainWindow", "OPEN NEW WINDOW"))
        self.pushButton_6.setText(_translate("MainWindow", "CHECK ATTENDANCE LIS
        self.pushButton_100.setText(_translate("MainWindow", "EMAIL SESSION"))
        self.menuAbout.setTitle(_translate("MainWindow", "About"))
        self.actionThis_system.setText(_translate("MainWindow", ">>This system <
        self.actionThis_system.setIconText(_translate("MainWindow", ">>This syst
        self.actionThis_system.setStatusTip(_translate("MainWindow", "Created by

        self.pushButton_1.clicked.connect(self.show_popup)
        self.pushButton_3.clicked.connect(self.registersession)
        self.pushButton_5.clicked.connect(self.traindatasession)
        self.pushButton_2.clicked.connect(self.show_popup2)
        self.pushButton_4.clicked.connect(self.attendancesession)
        self.pushButton_6.clicked.connect(self.openfile)
        self.pushButton_100.clicked.connect(self.emailpart)

    def show_popup(self):
        msg = QMessageBox()
        msg.setWindowTitle("Instruction")
        msg.setText(" 1. Click the register button \n\n 2. Key in your Name and
        msg.setIcon(QMessageBox.Information)
        msg.exec_()

    def registersession(self):
        self.Dialog1 = QtWidgets.QDialog()
        self.ui = Ui_Dialog1()
        self.ui.setupUi(self.Dialog1)
        self.ui.setupUi(self.Dialog1)
        self.Dialog1.show()

    def traindatasession(self):
        recognizer = cv2.face.LBPHFaceRecognizer_create()

        names = []    # create empty names list
        paths = []

        for users in os.listdir("/home/pi/Class_attendance_system/dataset"):
            names.append(users)

        for name in names:
            for image in os.listdir("/home/pi/Class_attendance_system/dataset/{}
                path_string = os.path.join("/home/pi/Class_attendance_system/dat
                paths.append(path_string)     # dataset/user/id_1.png

        faces = []
        ids = []

        for img_path in paths:
            image = Image.open(img_path).convert("L")   # open image in img_path

            imgNp = np.array(image, "uint8")  #numpy array these image

            faces.append(imgNp)  # append to faces list

            id = int(img_path.split("/")[6].split("_")[0])

            ids.append(id)  # All id in dataset append to id list

        ids = np.array(ids)
        recognizer.train(faces,ids)
        recognizer.write("training.yml")
        print( "[INFO] Data successfully trained.")
        msg = QMessageBox()
        msg.setWindowTitle("INFO")
        msg.setText("Traning process finish")
        msg.setIcon(QMessageBox.Information)
        msg.exec_()



    def show_popup2(self):
        msg = QMessageBox()
        msg.setWindowTitle("Instruction")
        msg.setText(" 1. Click the Open new window button at below \n\n 2. Key i
        msg.setIcon(QMessageBox.Information)
        msg.exec_()

    def attendancesession(self):
        self.Dialog2 = QtWidgets.QDialog()
        self.ui = Ui_Dialog2()
        self.ui.setupUi(self.Dialog2)
        msg.setIcon(QMessageBox.Information)
        msg.exec_()



    def show_popup2(self):
        msg = QMessageBox()
        msg.setWindowTitle("Instruction")
        msg.setText(" 1. Click the Open new window button at below \n\n 2. Key i
        msg.setIcon(QMessageBox.Information)
        msg.exec_()

    def attendancesession(self):
        self.Dialog2 = QtWidgets.QDialog()
        self.ui = Ui_Dialog2()
        self.ui.setupUi(self.Dialog2)
        self.Dialog2.show()

    def openfile(self):
        file = QFileDialog.getOpenFileName(None,"Open Attendance list","/home/pi
        self.data = pd.read_csv(file[0])
        numRows = len(self.data.index)
        self.tableWidget.setColumnCount(len(self.data.columns))
        self.tableWidget.setRowCount(numRows)
        self.tableWidget.setHorizontalHeaderLabels(self.data.columns)

        for i in range(numRows):
            for j in range(len(self.data.columns)):
                self.tableWidget.setItem(i,j, QTableWidgetItem(str(self.data.iat

        self.tableWidget.resizeColumnsToContents()
        self.tableWidget.resizeRowsToContents()
        self.tableWidget.setEditTriggers(QtWidgets.QTableWidget.NoEditTriggers)

    def emailpart(self):
        self.Dialog4 = QtWidgets.QDialog()
        self.ui = Ui_Dialog4()
        self.ui.setupUi(self.Dialog4)
        self.Dialog4.show()




if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle('Windows')
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

Appendix B : Register.py

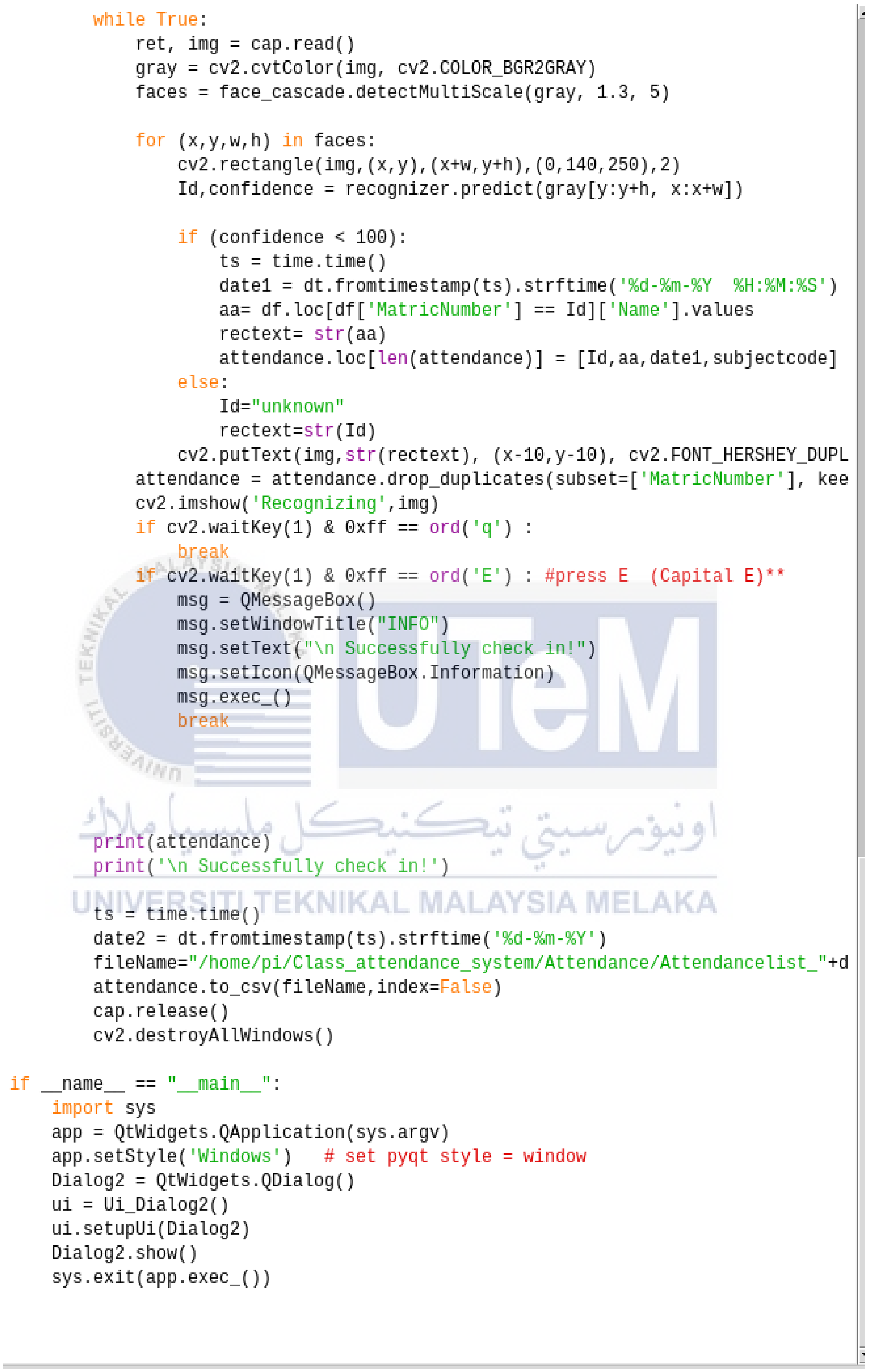


```
#! /usr/bin/env python3
import cv2
import csv
import time
import sys
from pathlib import Path
from PyQt5 import QtCore, QtGui, QtWidgets
from PyQt5.QtGui import *
from PyQt5.QtCore import QTimer
from PyQt5.QtWidgets import QMessageBox, QDialog,QApplication


class Ui_Dialog1(object):
    def setupUi(self, Dialog1):
        Dialog1.setObjectName("Dialog1")
        Dialog1.resize(772, 600)
        font = QtGui.QFont()
        font.setPointSize(14)
        Dialog1.setFont(font)
        Dialog1.setStyleSheet("background-color: rgb(35, 130, 255);")
        self.lineEdit = QtWidgets.QLineEdit(Dialog1)
        self.lineEdit.setGeometry(QtCore.QRect(320, 30, 211, 31))
        self.lineEdit.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit.setObjectName("lineEdit")
        self.lineEdit_2 = QtWidgets.QLineEdit(Dialog1)
        self.lineEdit_2.setGeometry(QtCore.QRect(320, 90, 211, 31))
        self.lineEdit_2.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.imgLabel_1 = QtWidgets.QLabel(Dialog1)
        self.imgLabel_1.setGeometry(QtCore.QRect(79, 160, 631, 351))
        self.imgLabel_1.setMinimumSize(QtCore.QSize(4, 0))
        self.imgLabel_1.setMaximumSize(QtCore.QSize(640, 480))
        self.imgLabel_1.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.imgLabel_1.setFrameShape(QtWidgets.QFrame.WinPanel)
        self.imgLabel_1.setText("")
        self.imgLabel_1.setObjectName("imgLabel_1")
        self.pushButton_8 = QtWidgets.QPushButton(Dialog1)
        self.pushButton_8.setGeometry(QtCore.QRect(300, 540, 191, 41))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_8.setFont(font)
        self.pushButton_8.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_8.setObjectName("pushButton_8")
        self.layoutWidget = QtWidgets.QWidget(Dialog1)
        self.layoutWidget.setGeometry(QtCore.QRect(150, 20, 151, 111))
        self.layoutWidget.setObjectName("layoutWidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.layoutWidget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.label_4 = QtWidgets.QLabel(self.layoutWidget)
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(18)
        font.setPointSize(18)
        font.setBold(True)
        font.setWeight(75)
        self.label_4.setFont(font)
        self.label_4.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_4.setObjectName("label_4")
        self.verticalLayout.addWidget(self.label_4)
        self.label_5 = QtWidgets.QLabel(self.layoutWidget)
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(18)
        font.setBold(True)
        font.setWeight(75)
        self.label_5.setFont(font)
        self.label_5.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_5.setObjectName("label_5")
        self.verticalLayout.addWidget(self.label_5)
        self.lineEdit.raise_()
        self.lineEdit_2.raise_()
        self.imgLabel_1.raise_()
        self.layoutWidget.raise_()
        self.pushButton_8.raise_()

        self.retranslateUi(Dialog1)
        QtCore.QMetaObject.connectSlotsByName(Dialog1)

        self.pushButton_8.clicked.connect(self.capturedata)


    def retranslateUi(self, Dialog1):
        _translate = QtCore.QCoreApplication.translate
        Dialog1.setWindowTitle(_translate("Dialog1", "Registration"))
        self.lineEdit_2.setPlaceholderText(_translate("Dialog1", "B0*********"))
        self.pushButton_8.setText(_translate("Dialog1", "Capture image"))
        self.label_4.setText(_translate("Dialog1", "<html><head/><body><p align=
        self.label_5.setText(_translate("Dialog1", "<html><head/><body><p align=


    def displayImage(self,img,window =1):
        qformat = QImage.Format_Indexed8

        if len(img.shape) == 3:
           if (img.shape[2]) == 4:
               qformat = QImage.Format_RGBA888
           else:
                qformat = QImage.Format_RGB888
        showImage = QImage(img, img.shape[1],img.shape[0], qformat)
        self.imgLabel_1.setPixmap(QPixmap.fromImage(showImage))


    def capturedata(self):

        face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xm

        cap = cv2.VideoCapture(0)
        cap = cv2.VideoCapture(0)

        userId = self.lineEdit_2.text()
        userName = self.lineEdit.text()
        img_counter = 0

        def saveImage(img, userName, userId, imgId):
            Path("/home/pi/Class_attendance_system/dataset/{}".format(userName))
            cv2.imwrite("/home/pi/Class_attendance_system/dataset/{}/{}_{}.png".

        while (cap.isOpened()):

            ret, img = cap.read()
            self.displayImage(img,1)
            cv2.waitKey()

            originalImg = img.copy()

            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5 )

            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,140,250),2)
                saveImage(gray[y:y+h,x:x+w],userName,userId,img_counter)

            self.displayImage(img,1)
            cv2.waitKey()

            if img_counter <= 69:
                print("{}.png taken!".format("dataset/" + str(userName) + '-' +
                img_counter += 1        # increment by 1
            if img_counter > 69:
                print("Process finish")
                msg = QMessageBox()
                msg.setWindowTitle("INFO")
                msg.setText("Capturing process finish")
                msg.setIcon(QMessageBox.Information)
                msg.exec_()
                break
                self.cap.isOpened = False

        data = [userId , userName]
        with open ('/home/pi/Class_attendance_system/UserDetails/UserDetails.csv
            writer = csv.writer(csvFile)
            writer.writerow(data)
            csvFile.close()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle('Windows')
    Dialog1 = QtWidgets.QDialog()
    ui = Ui_Dialog1()
    ui.setupUi(Dialog1)
    Dialog1.show()
```

Appendix C : TakeAttendance.py



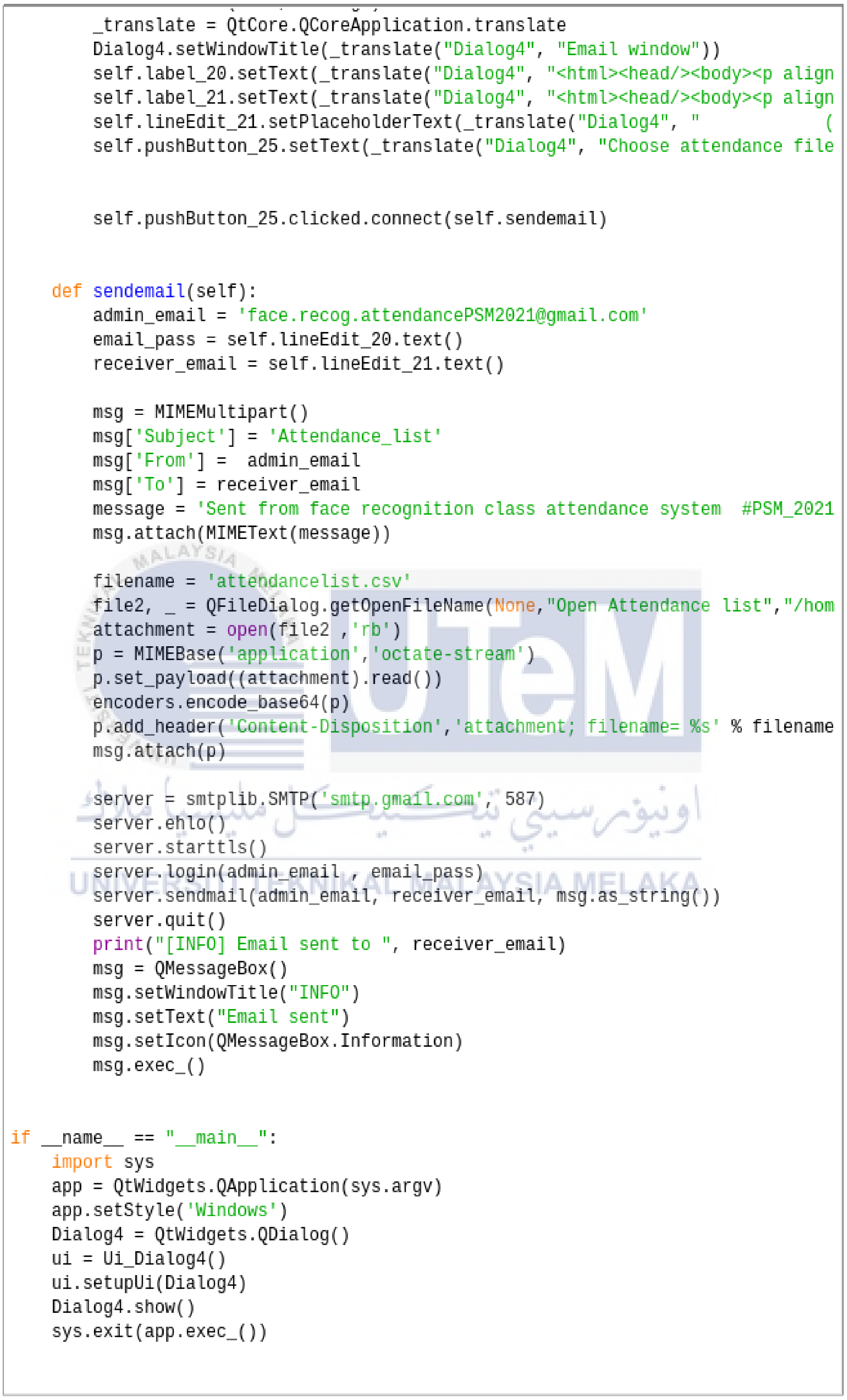
```
#! /usr/bin/env python3
import cv2
import csv
import os
import sys
import numpy as np
from PIL import Image
from datetime import datetime as dt
from datetime import date
import pandas as pd
import time
from PyQt5 import QtCore, QtGui, QtWidgets, uic
from PyQt5.QtCore import Qt
from PyQt5.QtGui import *
from PyQt5.QtWidgets import QMessageBox, QDialog,QApplication,QLabel
from subjectlist import Ui_MainWindow2

class Ui_Dialog2(object):
    def setupUi(self, Dialog2):
        Dialog2.setObjectName("Dialog2")
        Dialog2.resize(584, 440)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        Dialog2.setFont(font)
        Dialog2.setStyleSheet("background-color: rgb(35, 130, 255);")
        self.pushButton_9 = QtWidgets.QPushButton(Dialog2)
        self.pushButton_9.setGeometry(QtCore.QRect(160, 300, 231, 41))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_9.setFont(font)
        self.pushButton_9.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_9.setObjectName("pushButton_9")
        self.label_6 = QtWidgets.QLabel(Dialog2)
        self.label_6.setGeometry(QtCore.QRect(50, 90, 181, 51))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(18)
        font.setBold(True)
        font.setWeight(75)
        self.label_6.setFont(font)
        self.label_6.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_6.setObjectName("label_6")
        self.lineEdit_3 = QtWidgets.QLineEdit(Dialog2)
        self.lineEdit_3.setGeometry(QtCore.QRect(230, 90, 301, 51))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.lineEdit_3.setFont(font)
        self.lineEdit_3.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_3.setObjectName("lineEdit_3")
        self.lineEdit_3.setObjectName("lineEdit_3")
        self.pushButton_10 = QtWidgets.QPushButton(Dialog2)
        self.pushButton_10.setGeometry(QtCore.QRect(160, 200, 231, 51))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_10.setFont(font)
        self.pushButton_10.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_10.setObjectName("pushButton_10")

        self.retranslateUi(Dialog2)
        QtCore.QMetaObject.connectSlotsByName(Dialog2)

        self.pushButton_9.clicked.connect(self.attendancesession2)
        self.pushButton_10.clicked.connect(self.subjectlist)

    def retranslateUi(self, Dialog2):
        _translate = QtCore.QCoreApplication.translate
        Dialog2.setWindowTitle(_translate("Dialog2", "Take Attendance"))
        self.pushButton_9.setText(_translate("Dialog2", "Take Attendance"))
        self.label_6.setText(_translate("Dialog2", "<html><head/><body><p align=
        self.lineEdit_3.setPlaceholderText(_translate("Dialog2", "  BXXX_XXXX (I
        self.pushButton_10.setText(_translate("Dialog2", "Subject Code List"))

    def subjectlist(self):
        self.MainWindow2 = QtWidgets.QMainWindow()
        self.ui = Ui_MainWindow2()
        self.ui.setupUi(self.MainWindow2)
        self.MainWindow2.show()

    def attendancesession2(self):
        face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xm
        recognizer = cv2.face.LBPHFaceRecognizer_create()
        recognizer.read("/home/pi/Class_attendance_system/training.yml")

        subjectcode = self.lineEdit_3.text()

        df = pd.read_csv("/home/pi/Class_attendance_system/UserDetails/UserDetai
        col_names = ['MatricNumber','Name','Date/Time','Subjectcode']
        attendance = pd.DataFrame(columns = col_names)
        cap = cv2.VideoCapture(0)

        while True:
            ret, img = cap.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)

            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,140,250),2)
                Id,confidence = recognizer.predict(gray[y:y+h, x:x+w])

                if (confidence < 100):
                    ts = time.time()
                    date1 = dt.fromtimestamp(ts).strftime('%d-%m-%Y  %H:%M:%S')
```



```
        while True:
            ret, img = cap.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = face_cascade.detectMultiScale(gray, 1.3, 5)

            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(0,140,250),2)
                Id,confidence = recognizer.predict(gray[y:y+h, x:x+w])

                if (confidence < 100):
                    ts = time.time()
                    date1 = dt.fromtimestamp(ts).strftime('%d-%m-%Y  %H:%M:%S')
                    aa= df.loc[df['MatricNumber'] == Id]['Name'].values
                    rectext= str(aa)
                    attendance.loc[len(attendance)] = [Id,aa,date1,subjectcode]
                else:
                    Id="unknown"
                    rectext=str(Id)
                cv2.putText(img,str(rectext), (x-10,y-10), cv2.FONT_HERSHEY_DUPL
            attendance = attendance.drop_duplicates(subset=['MatricNumber'], kee
            cv2.imshow('Recognizing',img)
            if cv2.waitKey(1) & 0xff == ord('q') :
                break
            if cv2.waitKey(1) & 0xff == ord('E') : #press E  (Capital E)**
                msg = QMessageBox()
                msg.setWindowTitle("INFO")
                msg.setText("\n Successfully check in!")
                msg.setIcon(QMessageBox.Information)
                msg.exec_()
                break


        print(attendance)
        print('\n Successfully check in!')

        ts = time.time()
        date2 = dt.fromtimestamp(ts).strftime('%d-%m-%Y')
        fileName="/home/pi/Class_attendance_system/Attendance/Attendancelist_"+d
        attendance.to_csv(fileName,index=False)
        cap.release()
        cv2.destroyAllWindows()

if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle('Windows')   # set pyqt style = window
    Dialog2 = QtWidgets.QDialog()
    ui = Ui_Dialog2()
    ui.setupUi(Dialog2)
    Dialog2.show()
    sys.exit(app.exec_())
```

Appendix D : Email.py

```
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import smtplib
from PyQt5 import QtCore, QtGui, QtWidgets , uic
from PyQt5.QtWidgets import QMessageBox, QDialog,QApplication
from PyQt5.uic import loadUi
from PyQt5.QtWidgets import QFileDialog ,QTableWidgetItem, QTableWidget
import sys
import cv2
import csv
import numpy as np
import pandas as pd
import os
from PIL import Image
import time
from pathlib import Path
from PyQt5 import QtCore, QtGui, QtWidgets , uic
from PyQt5.QtWidgets import QFileDialog
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_Dialog4(object):
    def setupUi(self, Dialog4):
        Dialog4.setObjectName("Dialog4")
        Dialog4.resize(584, 440)
        font = QtGui.QFont()
        font.setPointSize(10)
        font.setBold(True)
        font.setWeight(75)
        Dialog4.setFont(font)
        Dialog4.setStyleSheet("background-color: rgb(35, 130, 255);")
        self.label_20 = QtWidgets.QLabel(Dialog4)
        self.label_20.setGeometry(QtCore.QRect(50, 60, 211, 51))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(18)
        font.setBold(True)
        font.setWeight(75)
        self.label_20.setFont(font)
        self.label_20.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_20.setObjectName("label_20")
        self.lineEdit_20 = QtWidgets.QLineEdit(Dialog4)
        self.lineEdit_20.setGeometry(QtCore.QRect(260, 60, 271, 51))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.lineEdit_20.setFont(font)
        self.lineEdit_20.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_20.setPlaceholderText("")
        self.lineEdit_20.setObjectName("lineEdit_20")
        self.label_21 = QtWidgets.QLabel(Dialog4)
        self.label_21.setGeometry(QtCore.QRect(50, 140, 211, 51))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setFamily("MS Sans Serif")
        font.setPointSize(18)
        font.setBold(True)
        font.setWeight(75)
        self.label_21.setFont(font)
        self.label_21.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.label_21.setObjectName("label_21")
        self.lineEdit_21 = QtWidgets.QLineEdit(Dialog4)
        self.lineEdit_21.setGeometry(QtCore.QRect(260, 140, 271, 51))
        font = QtGui.QFont()
        font.setPointSize(12)
        font.setBold(True)
        font.setWeight(75)
        self.lineEdit_21.setFont(font)
        self.lineEdit_21.setStyleSheet("background-color: rgb(255, 255, 255);")
        self.lineEdit_21.setObjectName("lineEdit_21")
        self.pushButton_25 = QtWidgets.QPushButton(Dialog4)
        self.pushButton_25.setGeometry(QtCore.QRect(80, 250, 411, 61))
        font = QtGui.QFont()
        font.setFamily("MS Sans Serif")
        font.setPointSize(16)
        font.setBold(True)
        font.setWeight(75)
        self.pushButton_25.setFont(font)
        self.pushButton_25.setStyleSheet("background-color: rgb(255, 170, 0);")
        self.pushButton_25.setObjectName("pushButton_25")

        self.retranslateUi(Dialog4)
        QtCore.QMetaObject.connectSlotsByName(Dialog4)

    def retranslateUi(self, Dialog4):
        _translate = QtCore.QCoreApplication.translate
        Dialog4.setWindowTitle(_translate("Dialog4", "Email window"))
        self.label_20.setText(_translate("Dialog4", "<html><head/><body><p align
        self.label_21.setText(_translate("Dialog4", "<html><head/><body><p align
        self.lineEdit_21.setPlaceholderText(_translate("Dialog4", "            (
        self.pushButton_25.setText(_translate("Dialog4", "Choose attendance file

        self.pushButton_25.clicked.connect(self.sendemail)


    def sendemail(self):
        admin_email = 'face.recog.attendancePSM2021@gmail.com'
        email_pass = self.lineEdit_20.text()
        receiver_email = self.lineEdit_21.text()

        msg = MIMEMultipart()
        msg['Subject'] = 'Attendance_list'
        msg['From'] =  admin_email
        msg['To'] = receiver_email
        message = 'Sent from face recognition class attendance system  #PSM_2021
        msg.attach(MIMEText(message))

        filename = 'attendancelist.csv'
        file2, _ = QFileDialog.getOpenFileName(None,"Open Attendance list","/hom
        attachment = open(file2 ,'rb')
```



```
        _translate = QtCore.QCoreApplication.translate
        Dialog4.setWindowTitle(_translate("Dialog4", "Email window"))
        self.label_20.setText(_translate("Dialog4", "<html><head/><body><p align
        self.label_21.setText(_translate("Dialog4", "<html><head/><body><p align
        self.lineEdit_21.setPlaceholderText(_translate("Dialog4", "             (
        self.pushButton_25.setText(_translate("Dialog4", "Choose attendance file


        self.pushButton_25.clicked.connect(self.sendemail)


    def sendemail(self):
        admin_email = 'face.recog.attendancePSM2021@gmail.com'
        email_pass = self.lineEdit_20.text()
        receiver_email = self.lineEdit_21.text()

        msg = MIMEMultipart()
        msg['Subject'] = 'Attendance_list'
        msg['From'] =  admin_email
        msg['To'] = receiver_email
        message = 'Sent from face recognition class attendance system  #PSM_2021
        msg.attach(MIMEText(message))

        filename = 'attendancelist.csv'
        file2, _ = QFileDialog.getOpenFileName(None,"Open Attendance list","/hom
        attachment = open(file2 ,'rb')
        p = MIMEBase('application','octate-stream')
        p.set_payload((attachment).read())
        encoders.encode_base64(p)
        p.add_header('Content-Disposition','attachment; filename= %s' % filename
        msg.attach(p)

        server = smtplib.SMTP('smtp.gmail.com', 587)
        server.ehlo()
        server.starttls()
        server.login(admin_email , email_pass)
        server.sendmail(admin_email, receiver_email, msg.as_string())
        server.quit()
        print("[INFO] Email sent to ", receiver_email)
        msg = QMessageBox()
        msg.setWindowTitle("INFO")
        msg.setText("Email sent")
        msg.setIcon(QMessageBox.Information)
        msg.exec_()


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    app.setStyle('Windows')
    Dialog4 = QtWidgets.QDialog()
    ui = Ui_Dialog4()
    ui.setupUi(Dialog4)
    Dialog4.show()
    sys.exit(app.exec_())
```