FACE RECOGNITION USING DEEP LEARNING

OOI ZI XEN

A project report submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering (Hons) Electronic Engineering

Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

January 2019

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ___________________________

Name : OOI ZI XEN

ID No. : 15AGB00038

Date : ___________________________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"FACE RECOGNITION USING DEEP LEARNING"** was prepared by **OOI ZI XEN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ____________________________

Supervisor : DR. TEH PEH CHIONG

Date : ____________________________

Specially dedicated to

my beloved mother and father

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Teh Peh Chiong for his invaluable advice, guidance and his enormous patience throughout the development of this project.

Next, I would like to thank all my friends who had contributed to the successful completion of this project, especially to those who are willing to participate in my faces data collecting and faces verifying experiments.

Finally, I must say thanks to my parents for their caring, unconditional support and encouragement throughout my study.

Thanks again to all who helped me.

FACE RECOGNITION USING DEEP LEARNING

ABSTRACT

Face recognition system is a technology accomplished at verifying or identifying a person from a video frame from a video source or a digital image. Multiple processing layers have been applied by deep learning to learn representations of data with multiple levels of feature extraction, which have achieved high accuracy to the real-world variations. Although the face recognition system has come a long way and its usage is crucial in several applications, it has remained a variety of challenges in face detection and recognition technologies, which including the pose variations, occlusions, facial expression changes, ageing of the face, illumination, etc.

In this project, the real-time face recognition system is implemented using pre-trained deep learning models with CCTV (Closed-Circuit Television) camera. The traditional CCTV is only good at recording and it is limited for campus safety and security nowadays. The face recognition system with CCTV camera can be used for controlling user access to physical locations of campus. The face recognition system does not require any kind of physical contact between the users and the device, which provide quick and convenient access to the authorized users.

The developed face recognition system exploits the pre-trained Multi-task Cascaded Convolutional Network (MTCNN) model for face detection and the standard techniques with FaceNet embeddings as feature vectors for face recognition. The developed face recognition system was tested with numerous experiments to analyze its performance. Empirical results show the face recognition in uncontrolled environments is much more challenging than in controlled conditions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

CCTV	Closed-Circuit Television
CNNs	Convolutional Neural Networks
FACES	Fast Airport Clearance Experience System
ID	Identity Document
IP	Internet Protocol
MTCNN	Multi-Task Cascaded Convolutional Network
NMS	Non-Maximum Suppression
O-Net	Output Network
PCA	Principal Component Analysis
P-Net	Proposal Network
RAM	Random-Access Memory
R-Net	Refinement Network
SVM	Support Vector Machine
GUI	Graphical User Interface
IDE	Integrated Development Environment
URL	Uniform Resource Locator
RTSP	Real Rime Streaming Protocol
LFW	Labeled Faces in the Wild
TTS	Text to Speech
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

A face recognition system is used to detect and recognize any human faces presence on images, non-real-time videos or real-time videos. In the view of computer, faces are complex objects. As such, detecting and recognising faces become a challenging task for any hardware system. For example, the pose, facial-wear, lighting or facial expression can degrade the performance of the face recognition system.

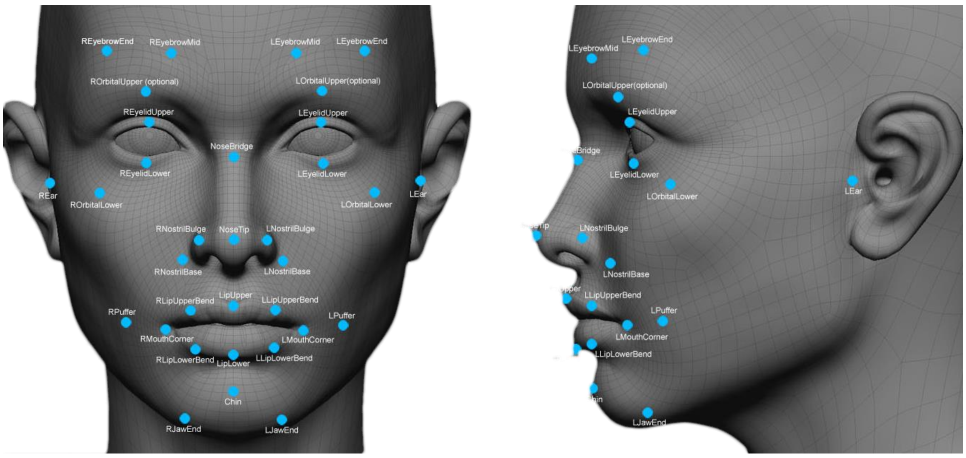


Figure 1.1: Face Features (Simon, 2015).

Face detection is basically an essential initial step in any automated face recognition system, which is to localize and extract the face area from the background. Face recognition works by performing verification and identification. The probe face image is extracted during face detection stage and it is compared with the database of previously enrolled known faces. The closest matching images is searched and carried out for verifying the most likely matched face.

According to Opalyn (2019), Malaysia's first facial recognition closed-circuit television (CCTV) surveillance has been installed on the Penang island at the beginning of this year. Penang Chief Minister Chow Kon Yeow hoped this new system will help reduce crime rates in the state. In the same year, Grab Malaysia was implementing the facial recognition technology in its mobile app in order to enhance the safety for its drivers (Hanis, 2019). Sharmila (2019) also mentioned that, AirAsia Group has disclosed the first airport face recognition system in Malaysia with self-boarding gate, which is its Fast Airport Clearance Experience System (FACES). This system uses the biometric facial recognition technology to identify the enrolled guests as they approach the automated boarding gate, allowing them to board their flight without having to show any travel documents.

In the other words, the facial recognition techniques have been introduced and are now becoming parts of our everyday lives.

1.2 Problem Statements

Campus safety and security is traditionally defined as one of the most pervasive problems that a campus or university must address, evolving to cope with the threats of terrorism, fraud, theft and unauthorised entry. Therefore, the CCTV systems have become part of most campuses, which supports the safety of students, staff and visitors.

The traditional CCTV is only good at recording without process on the footage. The security system with the traditional CCTV setup is limited for campus safety and security nowadays. As accessing to campus or campus library, it is often managed through showing the student or staff ID (Identity Document) cards, with security guards tasked with watching over the process on the CCTV monitor and stepping in when something suspicious is occurred. However, the traditional CCTV is impossible to catch everything in a such system. This leaves the opportunities for unauthorised access to the campus. Security checks can be stepped up, but it slows down the entry and exit flow, which annoying for the authorised users.

Thus, the face recognition system can be used with CCTV camera. It would be the perfect solution for controlling user access to physical locations such as building, offices, campus, library, computer lab, etc. The face detection and recognition technology does not require any kind of physical contact between the users and the device, which provide a quick and convenient access to the authorized users. In addition to improve security on the access control, the face recognition system with CCTV camera can be used for monitoring students around campus or recording the student attendance in class.

1.3 Aims and Objectives

The mainly objective of this project is to develop a real-time face recognition system using deep learning. In order to implement this system, the following objectives should be achieved:

- To detect human faces in a real-time video.
- To identify or verify the human faces in a real-time video.
- To perform the face recognition system with a CCTV camera.

1.4 Report Organization

This paper is organized in five (5) chapters. It contains introduction, literature review, methodology, results and discussion, and conclusion and recommendations.

Chapter 1 is about the introduction, in which the background information, problem statements and objectives are stated to provide an overview of this project.

Chapter 2 is the literature review, which consists the reviews regarding on the previous similar systems and techniques such as face detection algorithms, face recognition algorithms, challenging of face recognition system, etc.

Chapter 3 is the methodology. The details of the design, methods and tools used in the project will be presented in this chapter.

Chapter 4 is the results and discussions. All the testing results of the project are described and analysed.

Chapter 5 is the conclusion and recommendations. This chapter includes the summary of the project as well as the suggestions and further development that can be made.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Face Recognition System

Figure below illustrates the overview of face recognition process.

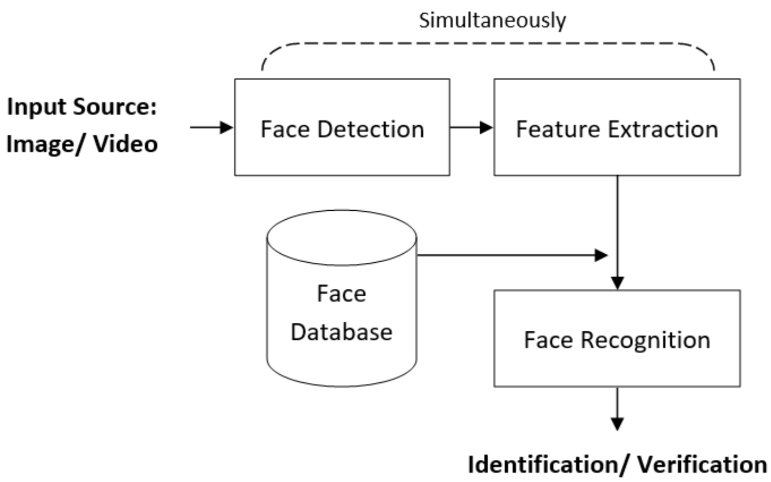


Figure 2.1: A general view of face recognition system (Rewagad *et al.*, 2013)

A video stream or an image is always as an input of face recognition system, and the output is a verification or identification of face(s) that appear in the image or video. A general face recognition process normally consists of various essential stages, in which mainly are face detection and face recognition.

Even though the accuracy of facial recognition system as a biometric technology is lower than fingerprint recognition and iris recognition, its contactless and non-invasive process makes it widely implemented (Jian *et al.*, 2014).

2.2 Face Detection

Face detection is one of the computer vision techniques with the concept of tracking or determining the face area in an image or a video (Sharifara *et al.*, 2014). Face detection is the first and essential step for any face processing systems, including face recognition system, detection of driver drowsiness in vehicles, access control, criminal identification and so-on.

There are many methods and algorithms have been proposed to detect face(s) in the given real-time video or image with different accuracy and false detection rates.

2.2.1 Face Detection by Viola-Jones Algorithm

Viola-Jones algorithm was founded by Paul Viola and Michael Jones in 2001 (Dang and Sharma, 2017). It came up with an effective algorithm to detect the human faces in real-time. Viola-Jones face detection method has four (4) main stages, which are Haar-like features selection, integral image creation, training of AdaBoost and cascading classifiers.

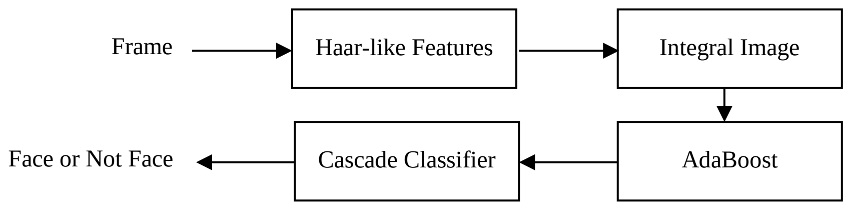
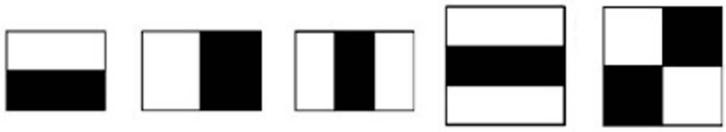


Figure 2.2: Schematic of Viola-Jones face detection method.

(1) Haar-like features selection. All human faces have the similar properties such as the eyes region is darker than the nose bridge region and so on. The Haar-like feature is used to compare these properties. There are three kind of Haar-like features: (i) two-rectangle feature, (ii) three-rectangular feature and (iii) four-rectangular feature.



(a) (b) (c) (d) (e)

Figure 2.3: Five Haar-like Patterns. (a) and (b) Shows the Two-rectangular Feature, (c) and (d) Shows the Three-rectangular Feature and (e) Shows the Four-rectangular Feature (Datta, Datta and Banerjee, 2015).

(2) Integral image creation. An integral image is generated to allow for very rapid feature evaluation with Haar-like features.

(3) Training of AdaBoost. It a simple and effective classifier, which choose a minor number of significant features from a massive library of potential features using an algorithm.

(4) Cascading classifiers. It combines the successively complex classifiers in a cascade structure that intensely increases the detector speed by focusing attention on promising regions of the image.

Dang and Sharma (2017) compared and analysed the precision and recall of four basic algorithms which are used for face detection: (1) Viola-Jones, (2) Support Vector Machines-Based (3) Neural Network-Based Face Detection and (4) SMQT Features and SNOW Classifier, face detection. They concluded that the Viola-Jones is the best among all these algorithms.

Datta, Datta and Banerjee (2015) also mentioned that the Viola-Jones face detector is able to process face image rapidly with high true detection rates is a real-time framework. Rajeshwari and Prof. Anala (2015) agreed that the Viola-Jones method gives better results, but it has greater time consumption than skin colour-based detection method and background subtraction method.

From the works mentioned above, the Viola-Jones algorithm can be considered as a popular method for face detection. However, the Viola-Jones algorithm cannot detect faces in a diverse position or angle (Enriquez, 2018). Low accuracy of face detection has resulted when the face is not presented in a front-facing position with proper lighting. In other words, the Viola-Jones face detection method could not handle non-frontal faces efficiently.

2.2.2 Face Detection by Multi-Task Cascaded Convolutional Network (MTCNN)

The Viola-Jones face detector while being prevalent in face detection tasks for a decade. As mentioned before, Viola-Jones face detector degrades expressively with greater visual variations of faces that usually occurs in real-world applications.

Inspired by the achievement obtained in computer vision tasks through the use of deep convolutional neural networks (CNNs), numerous studies were motivated to use this architecture for face detection. In this respect, Zhang *et al.* (2016) proposed a Multi-task Cascaded Convolutional Networks (MTCNN) based framework for joint face detection and alignment, which implements three stages of designed deep CNNs in a cascaded structure that forecast the face and landmark locations.

The overall pipeline of MTCNN is shown in figure below.

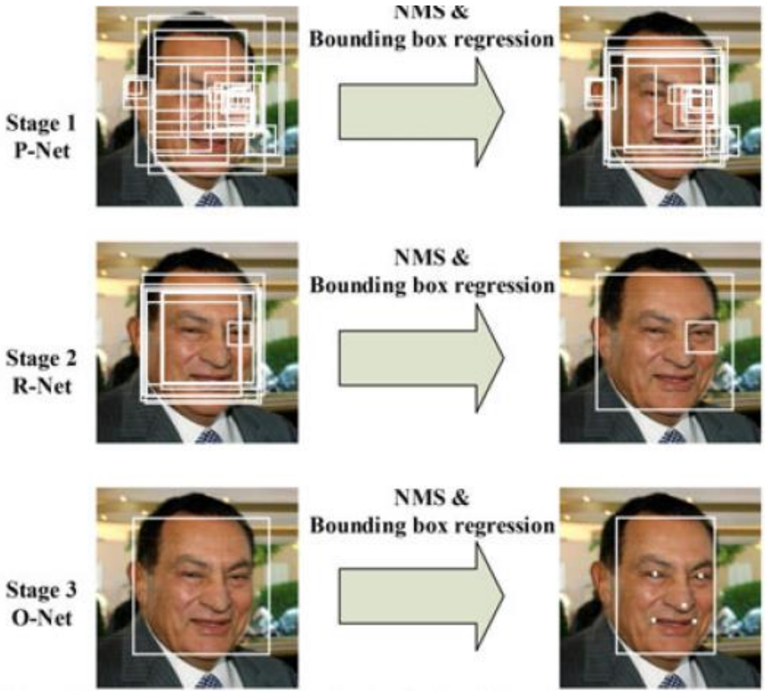


Figure 2.4: Pipeline of MTCNN framework (Zhang *et al.*, 2016).

There are three-stage cascaded framework in the MTCNN face detection method:

(1) Stage 1: Candidate facial windows and its bounding box regression vectors are produced quickly through the few layers of CNNs named Proposal Network (P-Net). These candidates are calibrated based on the estimated bounding box regression vectors. Then the non-maximum suppression (NMS) algorithm is used to merge the highly overlapped candidate windows.

(2) Stage 2: These candidates are refined in the next stage through a more complex CNNs called Refinement Network (R-Net), which further discards a large number of untrue candidates (non-faces windows).

(3) Stage 3: A more powerful CNNs, Output Network (O-Net) generates the final bounding box and facial landmarks position.

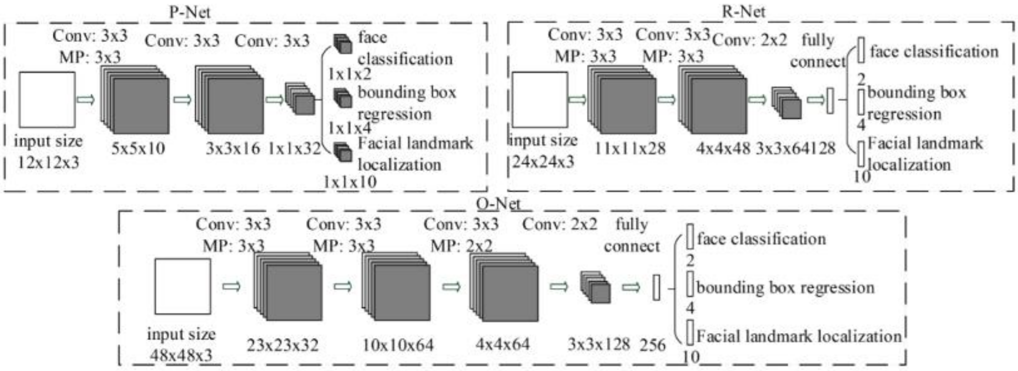


Figure 2.5: P-Net, R-Net, and O-Net architectures in MTCNN structure (Zhang et al., 2016).

Two different papers proposed by Cai *et al.* (2018) and Ma and Wang (2019) respectively, applied the MTCNN in their proposed approach to detect the faces in images. Ma and Wang (2019) mentioned that the MTCNN detector has a good performance in face detection, which works well for a large angle non-frontal face.

In contrast to the Viola-Jones algorithm, CNNs able to detect faces in various positions or angle and different lighting circumstances. As a result of it, the CNNs face detection method requires to store a larger amount of information and much more space needed than the Viola-Jones algorithm (Enriquez, 2018). Accessing too much RAM (Random-Access Memory) and requiring stronger processing unit is a constant problem to run the program of CNNs. It limited the CNNs can be implemented correctly. Therefore, while CNNs are faster and much more reliable in term of accuracy in face detection, the Viola-Jones algorithm is still widely used today.

2.3 Face Recognition

Face recognition methods have changed expressively over the years. Traditional face recognition methods relied on hand-crafted features (edges and texture descriptors) combined with machine learning techniques, such as principal component analysis (PCA) (Trigueros, Meng and Hartnett, 2018). At the same time, the traditional face recognition methods have a low accuracy to the different variations encountered in unconstrained environments.

The traditional face recognition methods have been superseded recently by deep learning methods based on CNNs, which has a great success in the computer vision community (Florian, Dmitry and James, 2015). The CNN-based face recognition methods achieve better accuracy as they are capable of learning the features that are robust to the real-world variations present in the face images used during training, for example, the FaceNet model.

2.3.1 Face Recognition by FaceNet

Researchers from Google, Florian, Dmitry and James (2015) have developed a face recognition system in 2015, called FaceNet. It is apart from other methods that proposed by Taigman *et al.* (2014) and Sun, Wang and Tang (2015), which use the CNNs bottleneck layer, or require additional post-processing such as concatenation of multiple models and PCA, as well as Support Vector Machine (SVM) classification.

The FaceNet system uses a deep convolutional network. Two different core architectures had been discussed for the FaceNet model structure, which are the Zeiler and Fergus (2014) style networks and the Inception type network that proposed by Szegedy *et al.* (2014). The core architecture is treated as a black box in the FaceNet model structure as shown in Figure 2.6.

Figure 2.6: FaceNet Overall Architecture (Florian, Dmitry and James, 2015).

Its network has a batch input layer and a deep CNN (Deep Architecture) followed by L_2 normalization, which outcomes in the face embedding. The triplet loss is followed during training. It maximises the distance between the anchor and a negative of a different identity and minimises the distance between an anchor and a positive, both of which have the same identity.

Figure 2.7: The Triplet Loss (Florian, Dmitry and James, 2015).

As mentioned before, there are two types of architectures had been used and explored. The first category is the standard convolutional layers of the Zeiler and Fergus (2014) style networks, which results in a model 22 layers deep as shown in Figure 2.8.

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Figure 2.8: The structure of Zeiler and Fergus (2014) based model (Florian, Dmitry and James, 2015).

The second category is the model based on GoogleNet style Inception type network by Szegedy *et al.* (2014).

type	**output size**	**depth**	#1×1	**#3×3 reduce**	#3×3	**#5×5 reduce**	#5×5	**pool proj (p)**	**params**	**FLOPS**
conv1 (7×7×3, 2)	112×112×64	1							9K	119M
max pool + norm	56×56×64	0						m 3×3, 2		
inception (2)	56×56×192	2		64	192				115K	360M
norm + max pool	28×28×192	0						m 3×3, 2		
inception (3a)	28×28×256	2	64	96	128	16	32	m, 32p	164K	128M
inception (3b)	28×28×320	2	64	96	128	32	64	L_2, 64p	228K	179M
inception (3c)	14×14×640	2	0	128	256,2	32	64,2	m 3×3,2	398K	108M
inception (4a)	14×14×640	2	256	96	192	32	64	L_2, 128p	545K	107M
inception (4b)	14×14×640	2	224	112	224	32	64	L_2, 128p	595K	117M
inception (4c)	14×14×640	2	192	128	256	32	64	L_2, 128p	654K	128M
inception (4d)	14×14×640	2	160	144	288	32	64	L_2, 128p	722K	142M
inception (4e)	7×7×1024	2	0	160	256,2	64	128,2	m 3×3,2	717K	56M
inception (5a)	7×7×1024	2	384	192	384	48	128	L_2, 128p	1.6M	78M
inception (5b)	7×7×1024	2	384	192	384	48	128	m, 128p	1.6M	78M
avg pool	1×1×1024	0								
fully conn	1×1×128	1							131K	0.1M
L2 normalization	1×1×128	0								
total									7.5M	1.6B

Figure 2.9: The structure of Inception type network (Florian, Dmitry and James, 2015).

Florian, Dmitry and James (2015) concluded that the final performance of both architectures perform comparably.

Once the FaceNet model having been trained with triplet loss for different classes of faces to capture its difference and its similarities, the 128-dimensional embedding returned by the FaceNet model can be used to clusters faces effectively. Figure 2.10 illustrates a 128 elements vector generated from a test image.

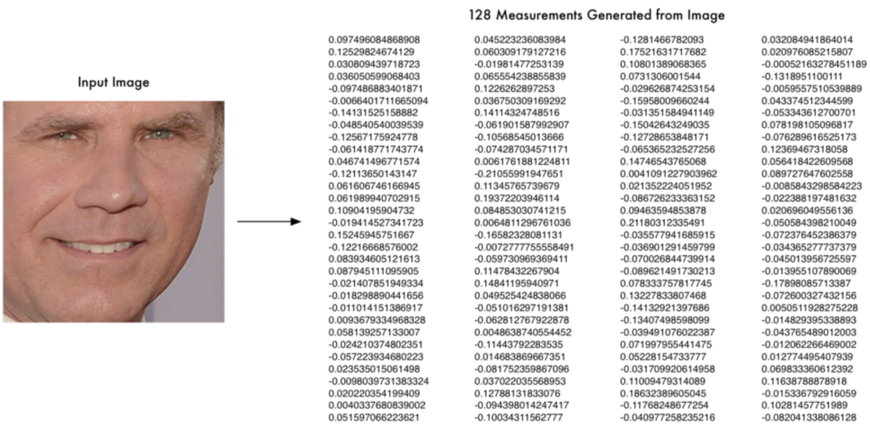


Figure 2.10: 128 Measurements Generated from A Face Image (Adam, 2016).

Florian, Dmitry and James noted “FaceNet, that directly learns a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity.” (Florian, Dmitry and James, 2015). Briefly, faces in this system are encoded and represented as 128-dimension points in space (called face vectors). Faces are considered a match when the Euclidean distance between the face vectors is smaller than its threshold value. The formula below is used to find the Euclidean distance between the face vectors.

$$\begin{aligned} \mathrm{d}(\mathrm{p},\mathrm{q}) = \mathrm{d}(\mathrm{q},\mathrm{p}) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_3)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\textstyle\sum_{i=1}^{n}(q_i - p_i)^2} \end{aligned} \tag{2.1}$$

Davis (2017) and Adam (2016) are the creator of *dlib* software library and the author of the *face_recognition* module reported that the FaceNet model has achieved an accuracy of 99.38% on the standard Labeled Faces in the Wild benchmark (which is a data set contains faces collected from the web). Brownlee (2019) also stated the FaceNet system is an effective and robust face recognition system. However, the overall accuracy of the FaceNet model is lower with Asian individuals as compared to European individuals (Adam, 2018). This is because the model was trained by the public datasets which are not evenly distributed amongst all individuals from all countries.

2.4 Challenges of Face Recognition System

Although the face recognition system has come a long way and its usage is crucial in several applications, it has remained a variety of challenges in face detection and recognition technologies. In other words, the face detection and recognition in the view of computer is not yet equalled to the human ability for detecting and recognizing face(s) despite many variations in appearance.

Olszewska (2016) stated a few challenges which all have a direct influence on the computer face recognition system, which including the pose variations, occlusions, facial expression changes, ageing of the face, illumination, digital image/ video resolution and modality, availability of face datasets, etc.

2.4.1 Pose Variations

The movement or orientation of head, which can be defined by the egocentric rotation angles could make the face recognition system across pose a difficult task. Figure below shows the head pose orientation in three degree of freedom (roll, pitch and raw).

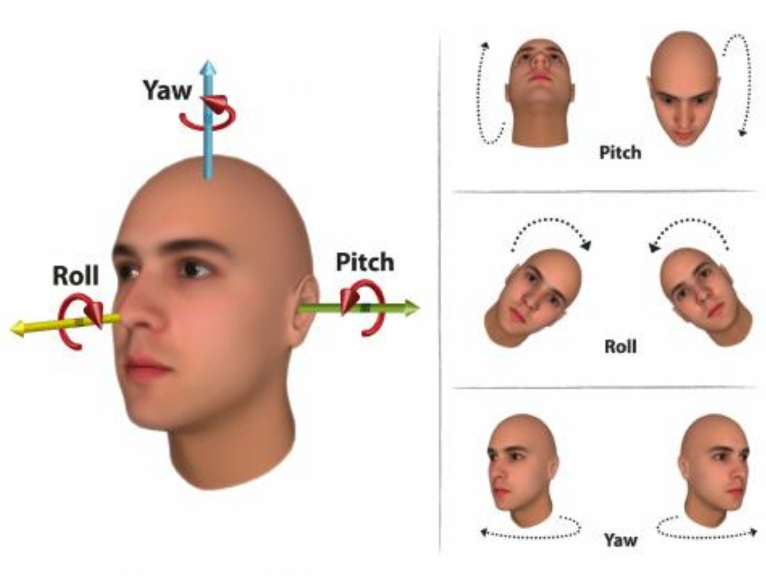


Figure 2.11: The head pose orientation in three degrees of freedom (roll, pitch and raw) (Arcoverde Neto *et al.*, 2014)

These orientations of head could make possible the substantial changes in face appearance (or shape) and cause intra-subject variations of the face. Even if the face recognition system can tolerate cases with small angles of rotation, it could make the face detection and face recognition more challenging when the rotation angle goes greater (Malikovich, Ugli and O'Ktamovna, 2017).

It would mislead the system result in faulty verification as the available image in the face database may have only the frontal view of the face which differs in the pose with the input sources (image or video).

2.4.2 Occlusions

In the face recognition system, occlusions refer to the case that the whole face is not available in the image or video. Occlusions can be things, illumination, spectacles, distance, glasses, beard or moustache, which may encumber to the face recognition system as well as a person talking on the phone, having their face covered with hands or wearing glasses, scarves, caps, masks, etc. (Malikovich, Ugli and O'Ktamovna, 2017). Such a situation can severely affect the face recognition process.

Figure 2.12: Example of a person with occlusions (Sharifara *et al.*, 2014)

2.4.3 Illumination

Illumination is also known as the lighting condition. Face detection and face recognition are more difficult to perform in a low level of lighting condition as the shadow may appear on the face or facial pattern can be (partially) indiscernible

(Olszewska, 2016). On the other hands, the high levels of lights can lead to over-exposure of the face and (partially) indiscernible facial patterns.

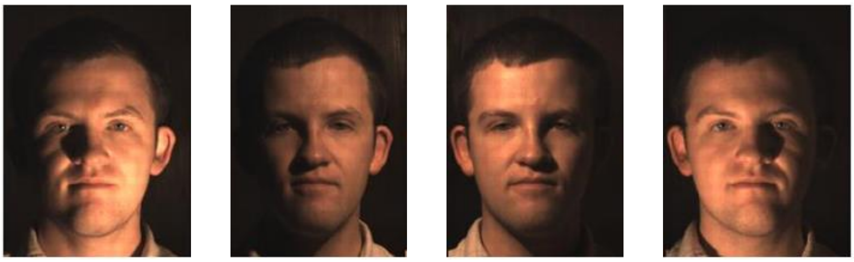


Figure 2.13: Example of a person with different light condition/ illumination (Ali, 2010)

2.4.4 Ageing of Faces

Everything changes with time, as well as the human face(s). With the increasing of age, the appearance of a person – face shape or line, hairstyle, eyebrows shape, etc. will also change. Figure 2.10 compares the ageing faces across 60 years. The appearance changing of the persons can directly affect the face recognition rate.

(a) (b)

Figure 2.14: Human ageing process aver 60 years. Same person has been pictured respectively at: (a) a younger age and (b) an older age (Katie, 2013)

2.4.5 Facial Expression Changes

Changes in facial expression due to the varying emotional states can cause the variability in face appearance (facial-feature shape change).

2.4.6 Similar Faces

Different persons may have a similar appearance (e.g. twins), which is sometimes impossible to identify them in a computer vision. In order to differentiate them, a second biometric factor such as fingerprint or iris-based authentication is needed.

2.4.7 Image resolution

Another typical factor influencing the efficiency of the face recognition system is the quality (resolution) of the input image or video (Olszewska, 2016). The low resolution of input sources (image or video) has restricted information as most detail is lost. Such a low-resolution face image cannot provide enough information for face detection and recognition, leading to the failure in face detection and recognition.

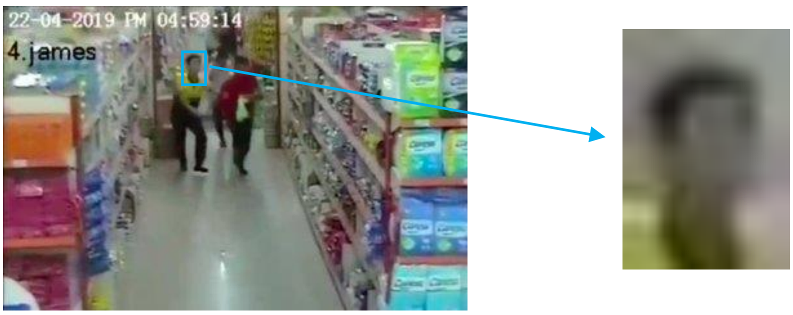


Figure 2.15: Low resolution image

2.5 Review on Types of CCTV

Face recognition system is increasingly used with CCTV cameras. It is important to make the right decision regarding the types of CCTV camera, locations of the CCTV camera to be installed (e.g. indoor or outdoor), the signal transmission speed of the CCTV camera, the CCTV camera resolution, the budget cost and so-on. Table below shows the different types of CCTV cameras.

Table 2.1: Comments on Different Types of CCTV Cameras (Vinay, 2019).

Types of CCTV cameras	Descriptions	Remarks
Dome Camera  **Figure 2.16: Demo Camera.**	Dome Cameras are classically used for indoor security systems. These cameras are in dome shape, which allows them to be inconspicuous as the direction of the cameras is difficult to tell, but it still visible to the eyes.	- Indoor used - Difficult use for face recognition system
Bullet Camera  **Figure 2.17: Bullet Camera**	Bullet Cameras can be used for outdoors and indoors. It is a long and tapered cylinder camera. They are typically weatherproof and installed inside protective casings.	- Indoor or outdoor used - Can be used in face recognition system
IP Camera  **Figure 2.18: IP Camera.**	The Internet Protocol (IP) camera is capable of transmitting recordings over the internet. It may or may not require a wire connection. Recording can be sent over a far distance without requiring any power boost using a cable.	- Indoor or outdoor used - Can be used in face recognition system - The strength of Wifi connection would affect the face recognition performance

2.6 Review on Python Programming Language

Figure below illustrates the fast growing of Python in popularity.

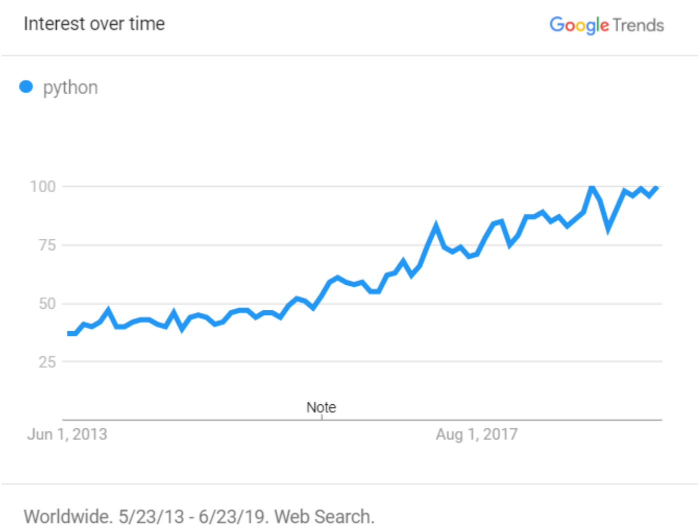


Figure 2.19: Python Tendency in Popularity for the Past 6 Years (Google Trends, 2019)

Python programming language has been widely used in recent time. Srinath (2017) listed the corporations that have used and been using Python for different function. For example, Google included Python in its web search system, YouTube video sharing service makes extensive use of Python, etc.

Python is a high-level and dynamic programming language which supports general-purpose. As a dynamically typed language, Python is considered slow due to its extreme flexibility. However, Python is very easy to understand and use as compared to other programming languages. Thus, it has gained popularity for being a beginner-friendly language. Python also has a large standard library which providing tools suited to many tasks.

In addition, Python scripts can be used on various operating systems such as Linux, Windows, Mac OS, UNIX, etc. In the other words, Python programs can be moved from one platform to another and run it without any changes.

CHAPTER 3

METHODOLOGY

3.1 System Overview

The objective of this project is to implement a real-time face recognition system with CCTV camera using deep learning. The developed system can be categorised into three phases: (1) Training, (2) Main System and (3) Data Recording, which as shown as Figure 3.1.

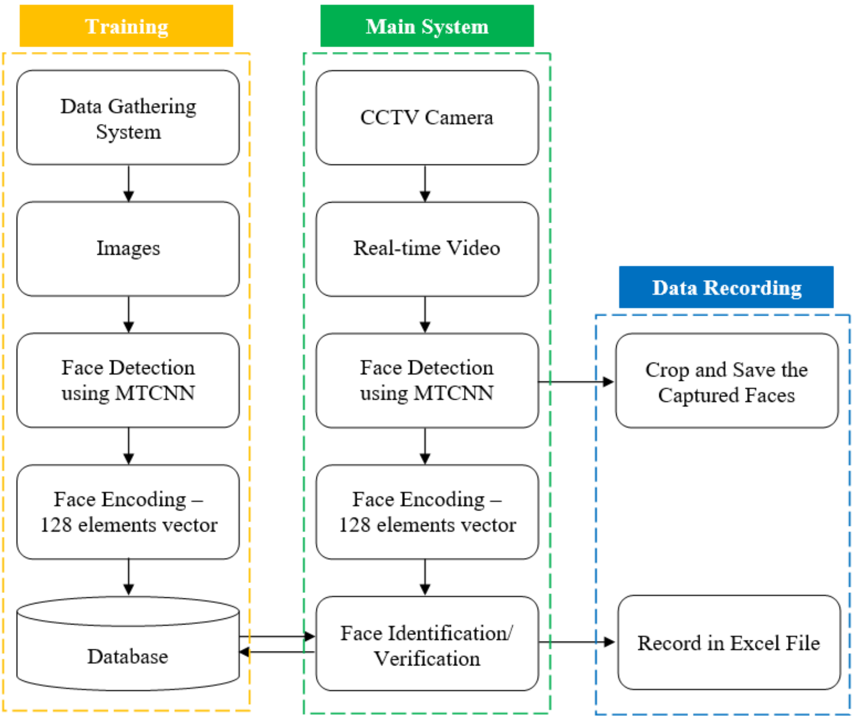


Figure 3.1: Overview of the Developed System

The training phase, consisting of the data gathering system, is required to gather the face data of the persons to be identified. The data includes the faces encodings and the detail information of the faces. These data are then stored as a database.

The main phase is the real-time face recognition system. It begins by acquiring the real-time video input from a CCTV camera. The video frames will be grabbed and resized to a smaller size for faster face detection processing. The MTCNN face detector is introduced to detect the face region from the background and produce the bounding box for the detected face. Next, the Dlib face recognition tool is applied to extract high-quality features from the detected face. This tool maps the features of the detected face to a 128 elements vector, which is known as face encoding. Then, the face recognition/verification is performed by comparing the predicted face encoding with the database of previously enrolled known faces encodings and checking if their Euclidean distance is small enough.

Data recording is designed for the next phase. Its purpose is to save and record all the detected and recognized faces that appeared in the frame of the real-time video.

3.2 Hardware and Software Requirement of System

This section provides an outline for the hardware and software requirements of the developed system.

3.2.1 Hardware Requirement of System

- Internet Protocol (IP) Day Night CCTV Camera with 1080p resolution
- CPU + GPU NVIDIA GeForce MX150 Laptop

3.2.2 Software Requirement of System

- Window 10
- Python (Anaconda)
- Python Library Packages:
    - OpenCV Library
    - Tensorflow Library
    - MTCNN Library
    - Face_recognition Library
    - Dlib Library
    - NumPy Library
    - Threading Library
    - OS Library
    - Time Library
    - Pyttsx3 Library
    - Openpyxl Library
    - Tkinter – Python GUI (Graphic User Interface)

3.3 System Implementation

3.3.1 Step 1: Install the Required Software and Library Packages

The purposed face recognition system is developed using Anaconda with the Python programming language.

Install Anaconda

Anaconda is essentially a nicely packaged Python IDE (Integrated Development Environment) that is shipped with tons of useful library package, such as NumPy, Time, Matplotlib and so-on. Anaconda also uses the concept of creating environments to isolate different libraries and versions.

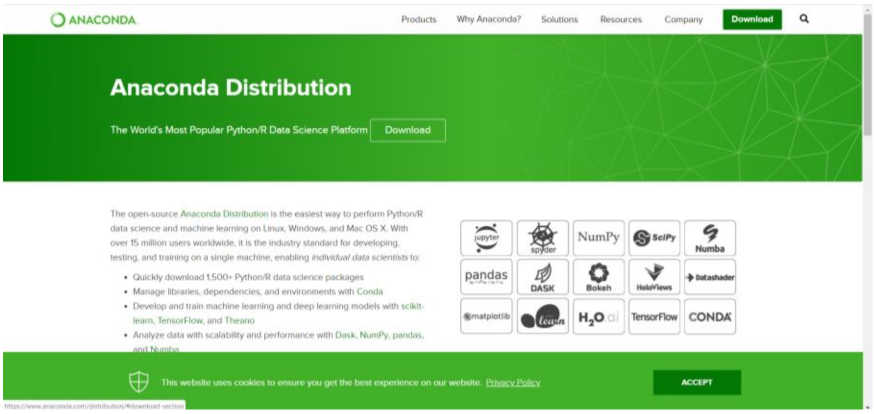


Figure 3.2: Anaconda Download Webpage.

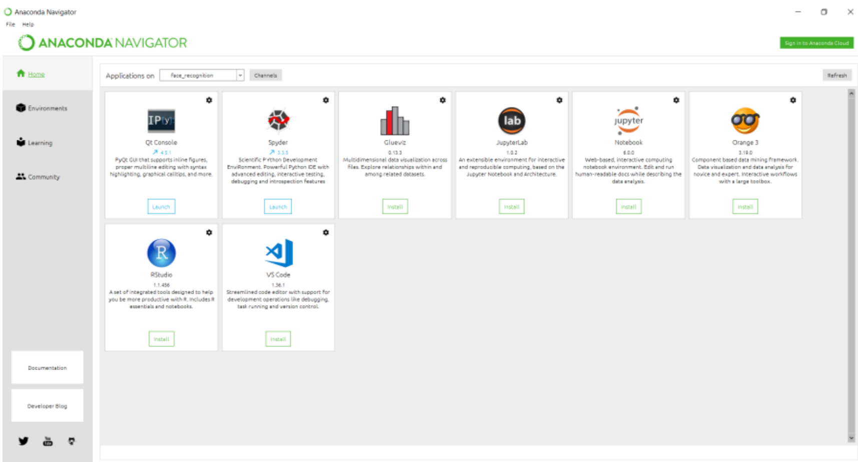


Figure 3.3: Anaconda Navigator Interface.

Install Library Packages

Some Python library packages are required to be installed for the developed system, which includes the OpenCV, Tensorflow, MTCNN, Face_recognition, Dlib, NumPy, Threading, OS, Time, Pyttsx3, Openpyxl and Tkinter Library. These library packages can be simply installed by entering the relevant command in the Python terminal as shown in table below.

Table 3.1: The commands correspond to the library packages.

Library Packages	Command
OpenCV	`pip install opencv-python`
Tensorflow	`pip install tensorflow-gpu`
MTCNN	`pip install mtcnn`
Dlib	`pip install dlib`
Face_recognition	`pip install face_recognition`
NumPy	`pip install numpy`
Threading	`pip install threaded`
OS	`pip install os-win`
Time	`pip install times`
Pyttsx3	`pip install pyttsx3`
Openpyxl	`pip install openpyxl`
Tkinter	-

Tkinter library is included in Python built-in package distribution. Thus, there is no need to worry about the Tkinter library's installation as it comes with Python default.

3.3.2 Step 2: Setup the Camera

An IP Day Night CCTV camera is used for the developed system, which is connected to the laptop via a router (without Internet) as shown in Figure 3.4. The power adapters of the camera and router are plugged-in and charged. The camera is connected to the laptop via router using ethernet cables. Now the camera is connecting to the laptop basically and physically. The CCTV camera can be mounted to anywhere that has a controlled environment.

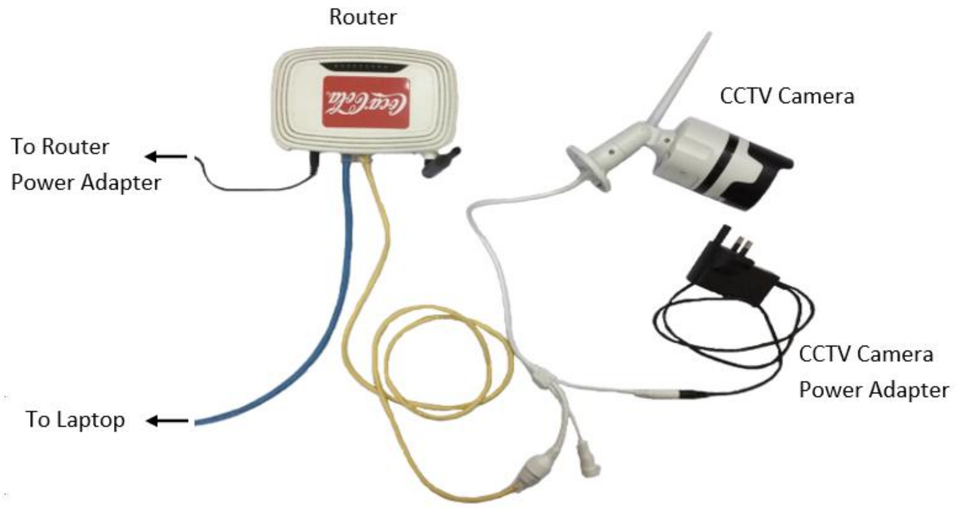


Figure 3.4: Camera Setting in the Developed System.

Then, the camera's IP address is obtained. By accessing to the IP address of the camera, the camera setting (resolution, frame rate, etc.) and display setting (brightness, contrast, etc.) can be configured.

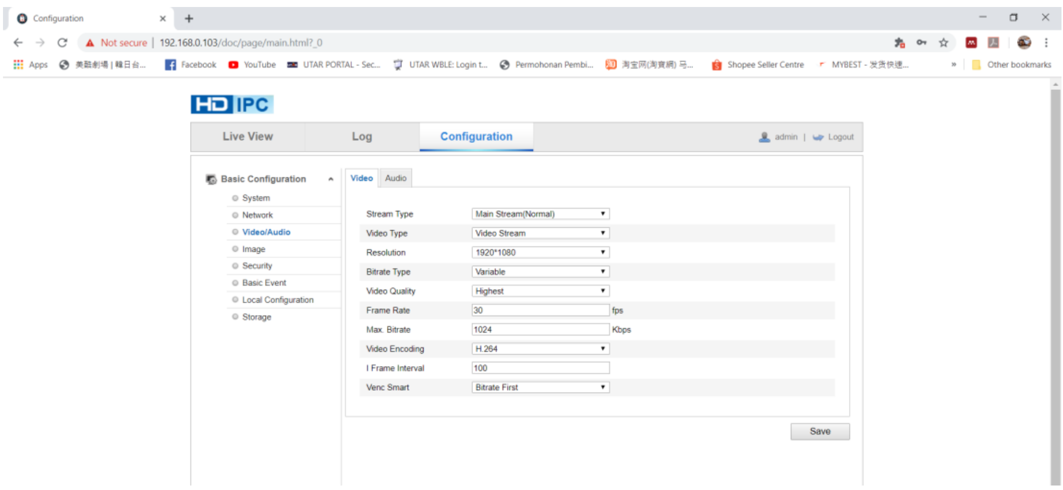


Figure 3.5: Camera Setting for the Developed System.

The display setting should be configured appropriately so that the frames can be captured in the advanced exposure modes.

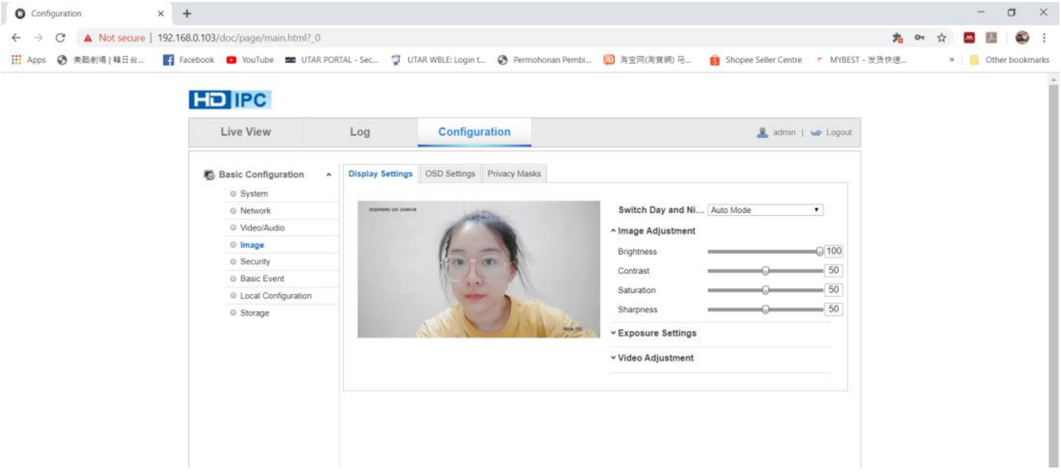


Figure 3.6: Camera Display Setting for the Developed System.

3.3.3 Step 3: Capture Video from Camera

Once the camera has been set up and the required Python library packages have been installed, the live video stream can be captured with the camera using Python. OpenCV library package provides a simple interface to this.

The IP CCTV camera is accessed in OpenCV by providing the streaming URL (Uniform Resource Locator) of the camera. RTSP (Real Time Streaming Protocol) protocol is used by the camera to steam video. The video stream is captured in the result of 1920 × 1080 pixels.

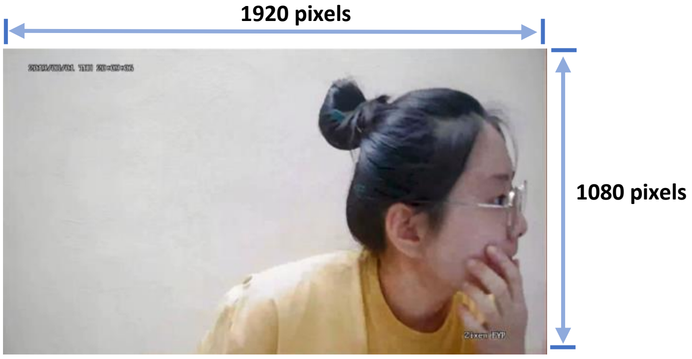


Figure 3.7: Video Acquisition.

3.3.4 Step 4: Face Detection

Face detection is the first and essential step for the face recognition system. A face must be captured in order to recognize it. The face detection technique in the developed system is using the pre-trained MTCNN face detector model. The working principle of MTCNN is mentioned in Chapter 2. It has a good face detection result, which works well for a large angle non-frontal face. Figure below shows the result of detecting facial regions.

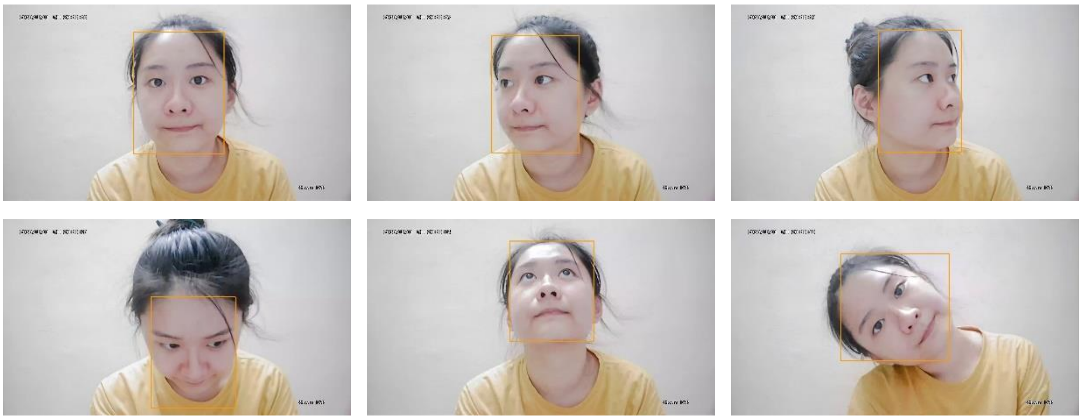


Figure 3.8: Face Detection in Different Angles of Face.

The locations and outlines of each person's eyes, nose, mouth and chin can also be obtained using the MTCNN face detector. There are total 68 coordinates on the face. However, this face landmark is not a must in the face recognition system.

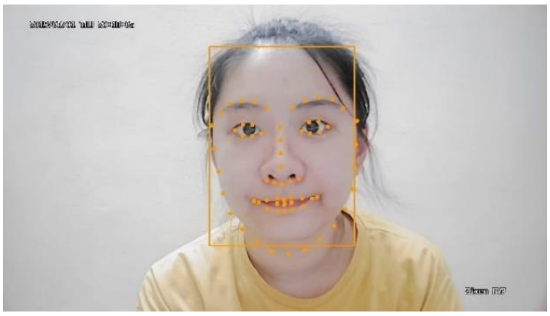


Figure 3.9: 68 Points Face Landmarks.

3.3.5 Step 5: Data Gathering

The main purpose of the data gathering system is to gather the face data of the persons to be identified. It is designed and developed using Python Tkinter Graphical User Interface (GUI), which allows the user can interact with the program easily. Figure below shows the user interface of the designed data gathering system

Figure 3.10: Python Tkinter GUI of The Designed Data Gathering System

Before capturing the faces, user is required to key in his/her details, which includes the full name, position, student or staff ID (identity document) number and the faculty.

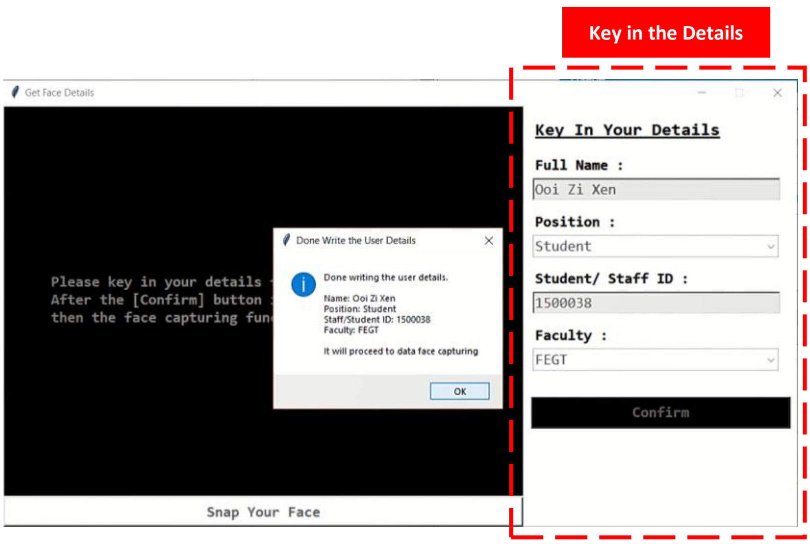
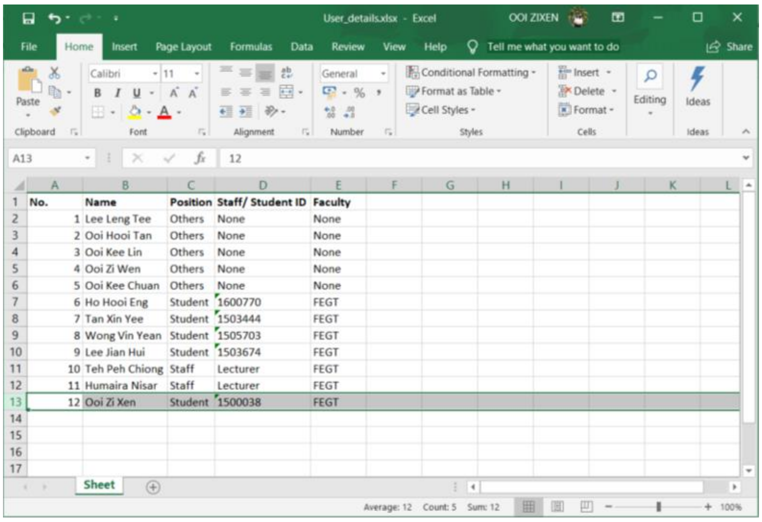


Figure 3.11: Key in the User's Details.

When the 'Confirm' button is pressed, the user's details will be saved to a excel file and the webcam will be turned on.



No.	Name	Position	Staff/ Student ID	Faculty
1	Lee Leng Tee	Others	None	None
2	Ooi Hooi Tan	Others	None	None
3	Ooi Kee Lin	Others	None	None
4	Ooi Zi Wen	Others	None	None
5	Ooi Kee Chuan	Others	None	None
6	Ho Hooi Eng	Student	1600770	FEGT
7	Tan Xin Yee	Student	1503444	FEGT
8	Wong Vin Yean	Student	1505703	FEGT
9	Lee Jian Hui	Student	1503674	FEGT
10	Teh Peh Chiong	Staff	Lecturer	FEGT
11	Humaira Nisar	Staff	Lecturer	FEGT
12	Ooi Zi Xen	Student	1500038	FEGT

Figure 3.12: Save User's Details in Excel File.

Once the webcam is turned on, it will proceed to the process of face capturing. User is required to click the button to capture the face image. In the face capturing process, two (2) images for each head pose (front, left side, right side, bottom and top view of face) should be captured, which are total ten (10) images. The instruction will be displayed on the frame.

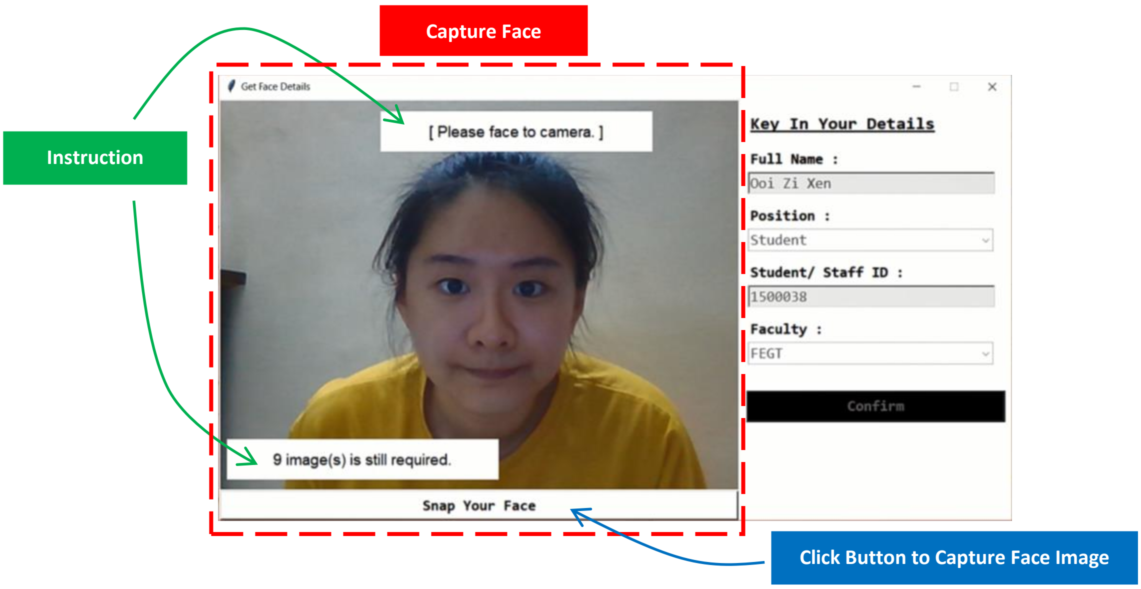


Figure 3.13: Process of Face Capturing.

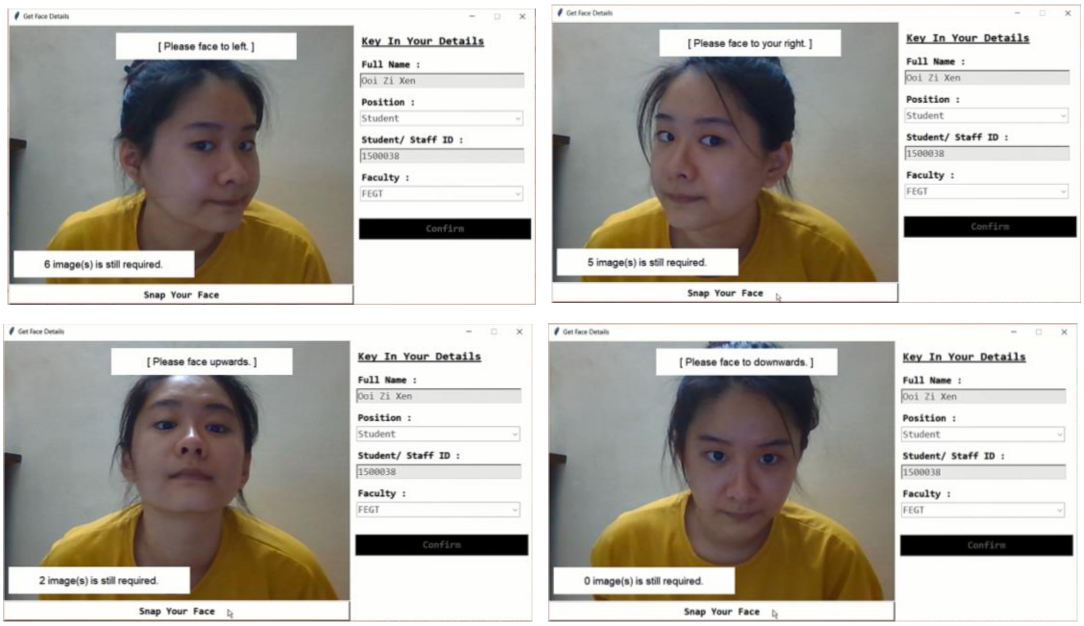


Figure 3.14: Capture Face in Different Side Views.

After that, the captured face images will be saved to the corresponding folder.

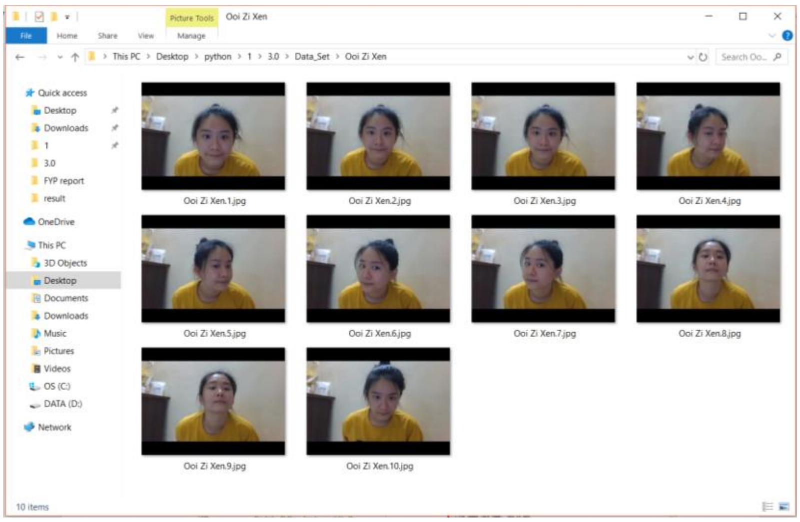


Figure 3.15: Save Images to Corresponding Folder.

Face appears on each image is assumed to be one (1) face. The face is then detected using MTCNN face detector. If more than one (1) face appear on the image, only the first detected face will be proceeded to obtain its face encodings. Face_recognition library is applied to extract high-quality features from the detected face vector and these features is mapped to a 128 elements vector (known as face encodings). The face_recognition library wraps around Dlib's facial recognition functionality that contains the FaceNet, which is an open face deep learning facial recognition model.

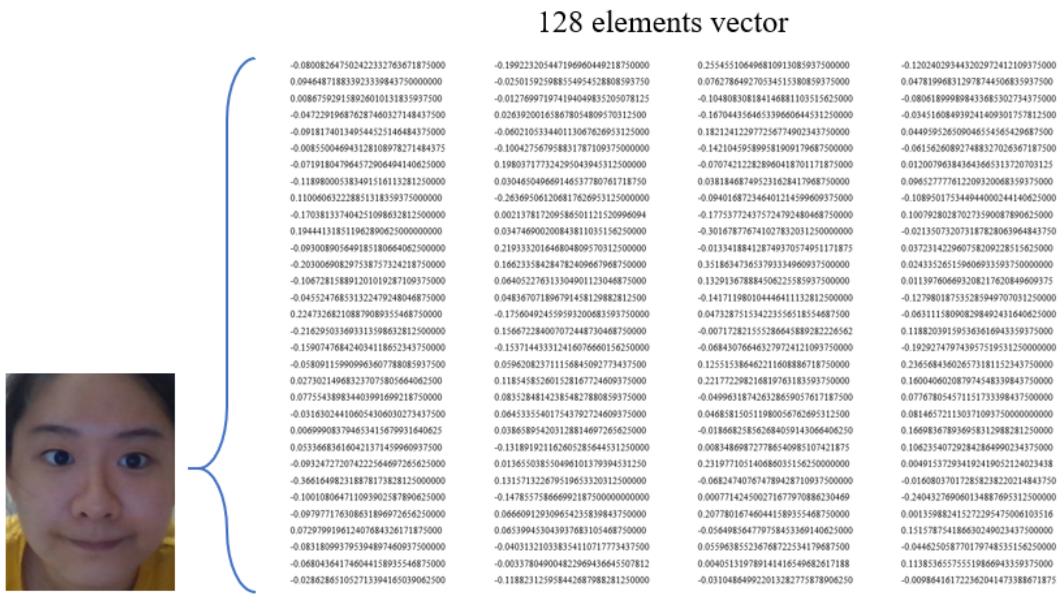


Figure 3.16: Obtain Face Encodings.

Next, these face encodings with the corresponding label for that faces are saved in a text file, which acts as the database. When new faces are added using the data gathering system, the above steps are repeated, and the new face encodings will be appended to the end of the text file. Other than our own dataset, Labeled Faces in the Wild (LFW) dataset also have been used for the database.

3.3.6 Step 6: Face Verification

This step is in fact the easiest step in the whole process. It is to find the person in the database of known people who has the closest measurement to the real-time detected face and return the relevant label.

The system starts by capturing the real-time video input from a CCTV camera. Each face that appears on the video frame is detected and the face features of the detected face will be mapped to a 128 elements vector using the face_recognition library. The face recognition/verification is then performed by comparing the predicted face encoding with the database of previously enrolled known faces encodings and checking if their Euclidean distance is small enough. The result of the system is the name of the person with his/her detail information.

Subsequently, a process, called text to speech (TTS) is performed. It is implemented by Pyttsx3 library with Python to make the computer speaks the written words in the English language. Once a known face is verified, the speaking sound is generated to notice user. For example, the computer will speak “OK” when the face is verified.

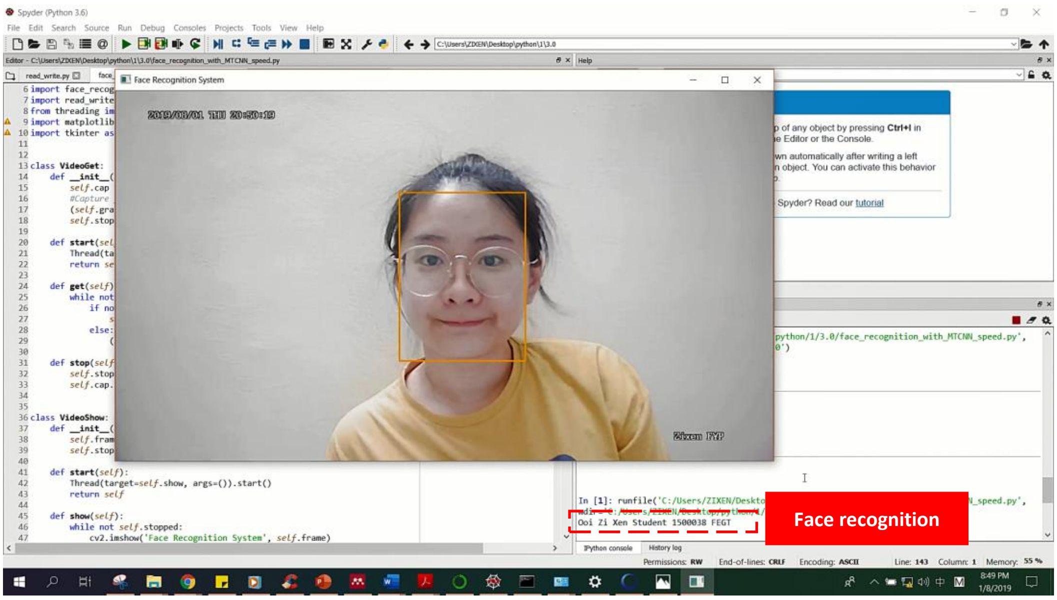


Figure 3.17: Face Verification.

3.3.7 Step 7: Data Recording

All the detected faces will be captured and saved to a folder in this step.

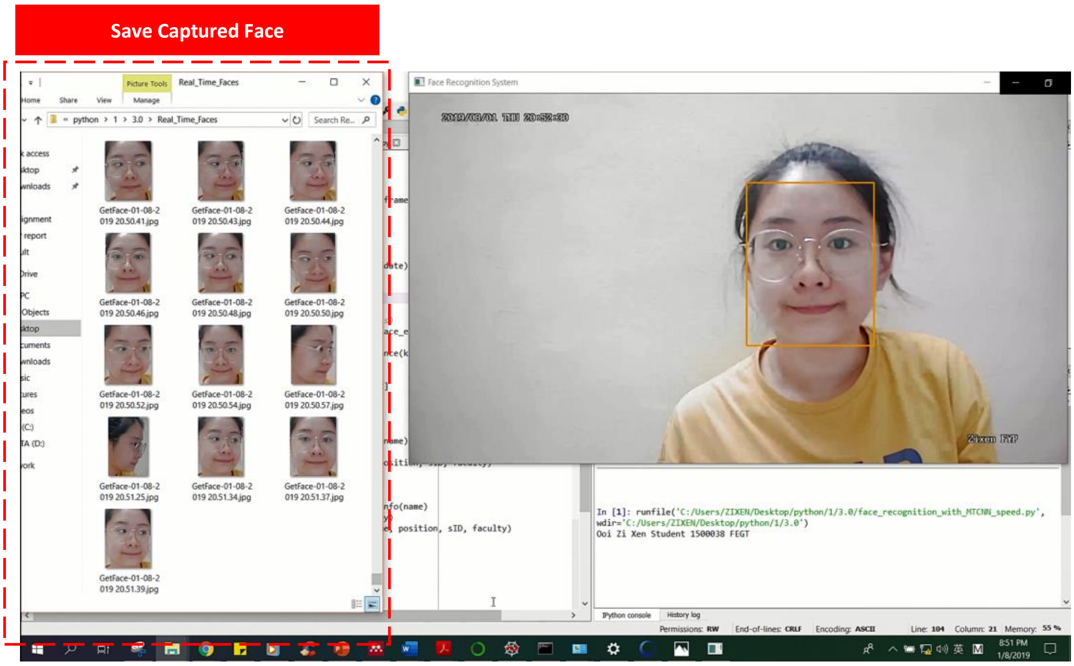


Figure 3.18: Save the Captured Face.

Lastly, the name and the details of the person with current date and time will be recorded in an excel file when the detected face is verified as a known face.

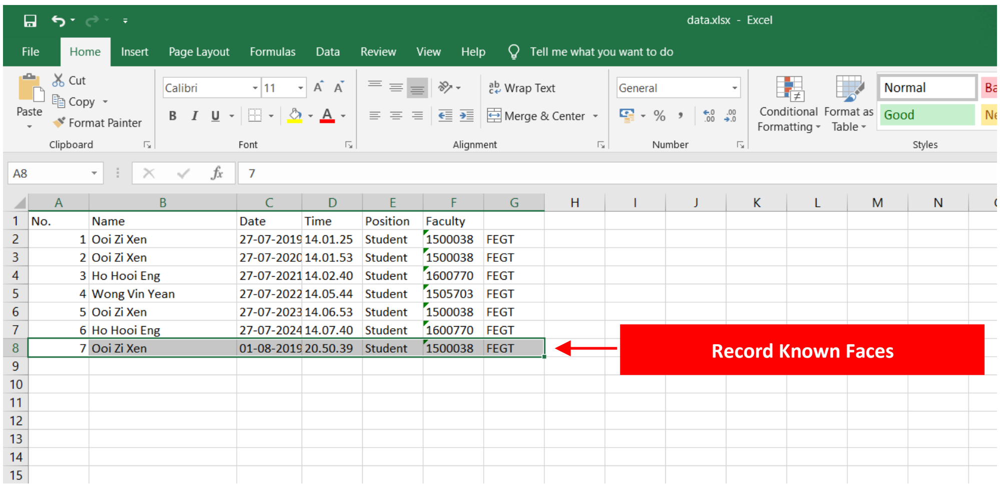


Figure 3.19: Record the Name and Details of the Known Face.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Results

Numerous experiments were conducted to test the performance of the developed face recognition system, which the face recognition system has tested with different lighting conditions, with different head pose variations, with occlusions, with different facial expressions, with different face sizes, with different motion, with spoofing face, with different faces and with multiple faces.

The descriptions of testing result in the developed system are simply discussed in the following content.

		True Condition	
	Total Population	Faces Detected	Faces Not Detected
Predicted Condition	Predicted Faces Detected	True Positive (TP)	False Positive (FP) *Type I Error*
	Predicted Non-Faces Detected	False Negative (FN) *Type II Error*	True Negative (TN)

Figure 4.1: Result Descriptions of Face Detection.

Figure 4.1 shows the result description of face detection, where

- True positive (TP) is the condition when face is correctly detected as faces.
- True negative (TN) is the condition when face is falsely detected as non-faces.
- False positive (FP) is the condition when non-face is falsely detected as faces.
- False negative (FN) is the condition when non-face is correctly detected as non-faces.

The face should be detected, then the face recognition could be conducted.

		True Condition	
	Total Population	Faces Recognized Correctly	Faces Recognized Wrongly
Predicted Condition	Predicted Faces Recognized	True Positive (TP)	False Positive (FP) *Type I Error*
	Predicted No Faces Recognized	False Negative (FN) *Type II Error*	True Negative (TN)

Figure 4.2: Result Descriptions of Face Recognition.

Figure 4.2 shows the result description of face recognition, in which

- True positive (TP) is the condition when face does match a person's face in a database, and that match is correct.
- True negative (TN) is the condition when face fails to match a person's face that is not contained in database.
- False positive (FP) is the condition when face does match a person's face in database, but that match is incorrect. Or the condition when face fails to match a person's face that is, in fact, contained in a database.
- False negative (FN) is the condition when face does match a person's face in database, which is actually not in a database.

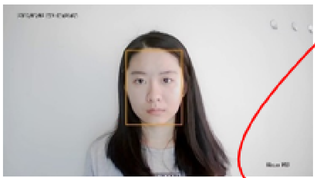
In the other words, TP, TN, FP and FN are used to describe the testing result for the developed face recognition system in this paper.

By referring to the testing results, the performance of the developed face recognition system can be evaluated by various measures, for example, accuracy. Accuracy is defined as "the fraction of quantity of correct classification over the entire number of samples."

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4.1)$$

In order to measure the accuracy of the developed system in different conditions, a real-time video is taken for each different condition and the total number of frames of that video is counted. Next, the frame numbers for TP, TN, FP and FN conditions is obtained for face detection and face recognition. Also, the average time taken to detect and recognize face is measured. Figure below simply explains the structure of the result in table-form.

Face was detected in total 137 frames.

Head Pose	Face Detection				Face Recognition			
	TP	TN	FP	FN	TP	TN	FP	FN
Frontal 	$\frac{137}{190}$	$\frac{53}{190}$	$\frac{0}{190}$	$\frac{0}{190}$	$\frac{127}{137}$	$\frac{0}{137}$	$\frac{10}{137}$	$\frac{0}{137}$
	Accuracy = 1.0000				**Accuracy** = 0.9270			
	Average Time Taken to Detect Face = 0.1555 s				**Average Time Taken to Recognize Face** = 2.1407 s			

The real-time video was taken with 190 frames.

Since face was detected in total 137 frames, face recognition was only performed on these 137 frames.

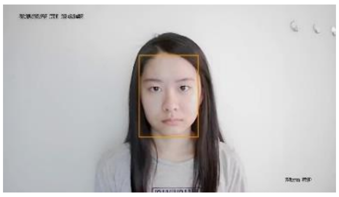
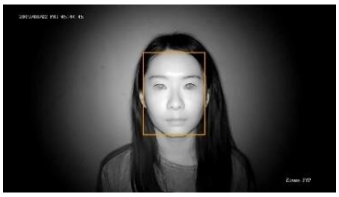
Figure 4.3: Example of the Result in Table-form with Explanation.

The accuracy is calculated using equation 4.1.

4.1.1 With Different Lighting Conditions

The developed face recognition system was tested with different lighting conditions, which were high, normal and low light conditions.

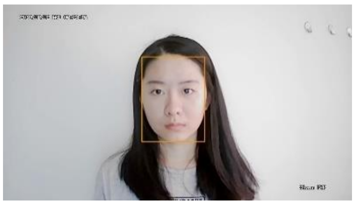
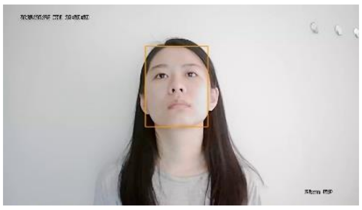
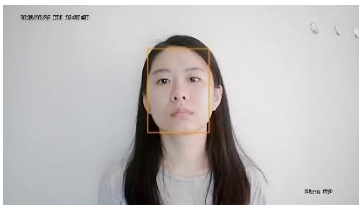
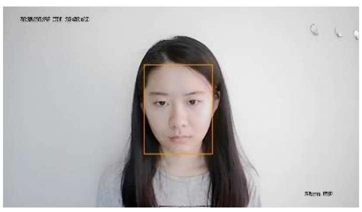
Table 4.1: Testing Results of The Developed Face Recognition System with Different Light Conditions.

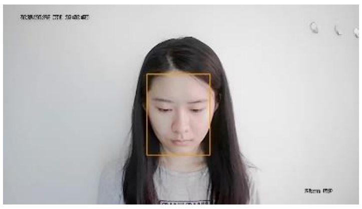
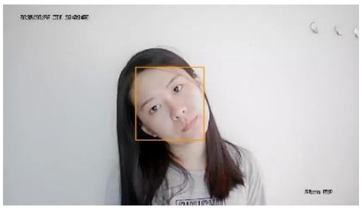
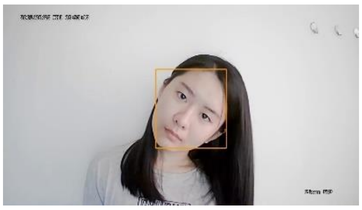
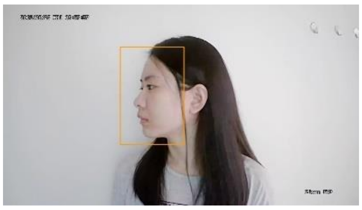
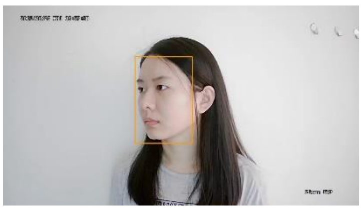
Light Conditions	Face Detection				Face Recognition			
	TP	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
(1) High 	$\frac{217}{374}$	$\frac{153}{374}$	$\frac{4}{374}$	$\frac{0}{374}$	$\frac{89}{217}$	$\frac{0}{217}$	$\frac{128}{217}$	$\frac{0}{217}$
	Accuracy = 0.9893				**Accuracy** = 0.4101			
	Average time taken to detect face = 0.1679 s				**Average time taken to recognize face** = 2.3626 s			
(2) Normal 	$\frac{347}{388}$	$\frac{41}{388}$	$\frac{0}{388}$	$\frac{0}{388}$	$\frac{347}{347}$	$\frac{0}{347}$	$\frac{0}{347}$	$\frac{0}{347}$
	Accuracy = 1.0000				**Accuracy** = 1.0000			
	Average time taken to detect face = 0.1541 s				**Average time taken to recognize face** = 2.2114 s			
(3) Low 	$\frac{315}{413}$	$\frac{92}{413}$	$\frac{5}{413}$	$\frac{1}{413}$	$\frac{300}{316}$	$\frac{0}{316}$	$\frac{16}{316}$	$\frac{0}{316}$
	Accuracy = 0.9855				**Accuracy** = 0.9494			
	Average time taken to detect face = 0.1427 s				**Average time taken to recognize face** = 1.8358 s			

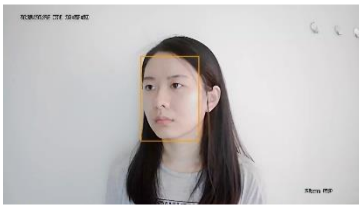
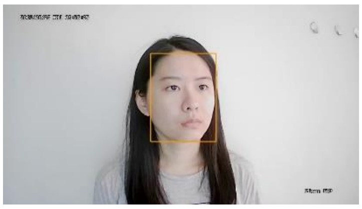
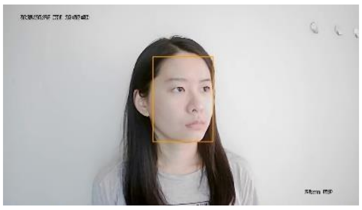
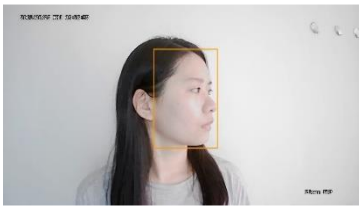
4.1.2 With Different Head Pose Variations

The developed face recognition system was tested with different head pose variations.

Table 4.2: Testing Results of The Developed Face Recognition System with Different Head Pose Variations.

<table>
<tr><th rowspan="2">Head Pose</th><th colspan="4">Face Detection</th><th colspan="4">Face Recognition</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th></tr>
<tr><td rowspan="3">(1) Frontal
</td><td>$\frac{137}{190}$</td><td>$\frac{53}{190}$</td><td>$\frac{0}{190}$</td><td>$\frac{0}{190}$</td><td>$\frac{127}{137}$</td><td>$\frac{0}{137}$</td><td>$\frac{10}{137}$</td><td>$\frac{0}{137}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9270</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1555 s</td><td colspan="4">Average time taken to recognize face = 2.1407 s</td></tr>
<tr><td rowspan="3">(2) Pitch (− 90°)
</td><td>$\frac{0}{89}$</td><td>$\frac{0}{89}$</td><td>$\frac{89}{89}$</td><td>$\frac{0}{89}$</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="4">Accuracy = 0.0000</td><td colspan="4">Accuracy = 0.0000</td></tr>
<tr><td colspan="4">Average time taken to detect face = /</td><td colspan="4">Average time taken to recognize face = /</td></tr>
<tr><td rowspan="3">(3) Pitch (−45°)
</td><td>$\frac{27}{85}$</td><td>$\frac{58}{85}$</td><td>$\frac{0}{85}$</td><td>$\frac{0}{85}$</td><td>$\frac{10}{27}$</td><td>$\frac{0}{27}$</td><td>$\frac{17}{27}$</td><td>$\frac{0}{27}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.3704</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1462 s</td><td colspan="4">Average time taken to recognize face = 2.7585 s</td></tr>
<tr><td rowspan="3">(4) Pitch (− 15°)
</td><td>$\frac{57}{97}$</td><td>$\frac{40}{97}$</td><td>$\frac{0}{97}$</td><td>$\frac{0}{97}$</td><td>$\frac{54}{57}$</td><td>$\frac{0}{57}$</td><td>$\frac{3}{57}$</td><td>$\frac{0}{57}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9474</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1492 s</td><td colspan="4">Average time taken to recognize face = 2.8024 s</td></tr>
<tr><td rowspan="3">(5) Pitch (+15°)
</td><td>$\frac{47}{104}$</td><td>$\frac{57}{104}$</td><td>$\frac{0}{104}$</td><td>$\frac{0}{104}$</td><td>$\frac{45}{47}$</td><td>$\frac{0}{47}$</td><td>$\frac{2}{47}$</td><td>$\frac{0}{47}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9574</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1513 s</td><td colspan="4">Average time taken to recognize face = 2.6909 s</td></tr>
</table>

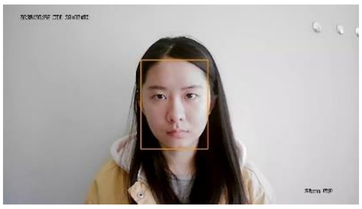
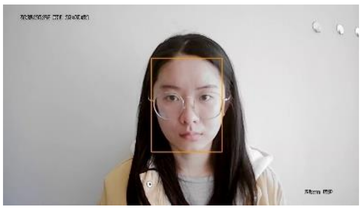
<table>
<tr><td rowspan="3">(6) Pitch (+45°)
</td><td>$\frac{40}{83}$</td><td>$\frac{43}{83}$</td><td>$\frac{0}{83}$</td><td>$\frac{0}{83}$</td><td>$\frac{33}{40}$</td><td>$\frac{0}{40}$</td><td>$\frac{7}{40}$</td><td>$\frac{0}{40}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.8250</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1535 s</td><td colspan="4">Average time taken to recognize face = 2.6497 s</td></tr>
<tr><td rowspan="3">(7) Pitch (+90°)
</td><td>$\frac{0}{136}$</td><td>$\frac{0}{136}$</td><td>$\frac{136}{136}$</td><td>$\frac{0}{136}$</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td colspan="4">Accuracy = 0.0000</td><td colspan="4">Accuracy = 0.0000</td></tr>
<tr><td colspan="4">Average time taken to detect face = /</td><td colspan="4">Average time taken to recognize face = /</td></tr>
<tr><td rowspan="3">(8) Roll (− 45°)
</td><td>$\frac{152}{213}$</td><td>$\frac{60}{213}$</td><td>$\frac{1}{213}$</td><td>$\frac{0}{213}$</td><td>$\frac{151}{152}$</td><td>$\frac{0}{152}$</td><td>$\frac{1}{152}$</td><td>$\frac{0}{152}$</td></tr>
<tr><td colspan="4">Accuracy = 0.9953</td><td colspan="4">Accuracy = 0.9934</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1456 s</td><td colspan="4">Average time taken to recognize face = 3.0463 s</td></tr>
<tr><td rowspan="3">(9) Roll (+45°)
</td><td>$\frac{147}{152}$</td><td>$\frac{0}{152}$</td><td>$\frac{5}{152}$</td><td>$\frac{0}{152}$</td><td>$\frac{145}{147}$</td><td>$\frac{0}{147}$</td><td>$\frac{2}{147}$</td><td>$\frac{0}{147}$</td></tr>
<tr><td colspan="4">Accuracy = 0.9671</td><td colspan="4">Accuracy = 0.9864</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1468 s</td><td colspan="4">Average time taken to recognize face = 2.9472 s</td></tr>
<tr><td rowspan="3">(10) Yaw (− 90°)
</td><td>$\frac{13}{210}$</td><td>$\frac{0}{210}$</td><td>$\frac{197}{210}$</td><td>$\frac{0}{210}$</td><td>$\frac{0}{13}$</td><td>$\frac{0}{13}$</td><td>$\frac{13}{13}$</td><td>$\frac{0}{13}$</td></tr>
<tr><td colspan="4">Accuracy = 0.0619</td><td colspan="4">Accuracy = 0.0000</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1746 s</td><td colspan="4">Average time taken to recognize face = 2.8669 s</td></tr>
<tr><td rowspan="3">(11) Yaw (− 45°)
</td><td>$\frac{39}{134}$</td><td>$\frac{95}{134}$</td><td>$\frac{0}{134}$</td><td>$\frac{0}{134}$</td><td>$\frac{38}{39}$</td><td>$\frac{0}{39}$</td><td>$\frac{1}{39}$</td><td>$\frac{0}{39}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy =0.9744</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1474 s</td><td colspan="4">Average time taken to recognize face = 2.4006 s</td></tr>
</table>

<table>
<tr><td rowspan="3">(12) Yaw $(-15°)$
</td><td>$\frac{67}{67}$</td><td>$\frac{0}{67}$</td><td>$\frac{0}{67}$</td><td>$\frac{0}{67}$</td><td>$\frac{66}{67}$</td><td>$\frac{0}{67}$</td><td>$\frac{1}{67}$</td><td>$\frac{0}{67}$</td></tr>
<tr><td colspan="4">**Accuracy** = 1.0000</td><td colspan="4">**Accuracy** = 0.9851</td></tr>
<tr><td colspan="4">**Average time taken to detect face** = 0.1500 s</td><td colspan="4">**Average time taken to recognize face** = 3.0739 s</td></tr>
<tr><td rowspan="3">(13) Yaw $(+15°)$
</td><td>$\frac{63}{106}$</td><td>$\frac{43}{106}$</td><td>$\frac{0}{106}$</td><td>$\frac{0}{106}$</td><td>$\frac{61}{63}$</td><td>$\frac{0}{63}$</td><td>$\frac{2}{63}$</td><td>$\frac{0}{63}$</td></tr>
<tr><td colspan="4">**Accuracy** = 1.0000</td><td colspan="4">**Accuracy** = 0.9683</td></tr>
<tr><td colspan="4">**Average time taken to detect face** = 0.1541 s</td><td colspan="4">**Average time taken to recognize face** = 2.2552 s</td></tr>
<tr><td rowspan="3">(14) Yaw $(+45°)$
</td><td>$\frac{29}{97}$</td><td>$\frac{68}{97}$</td><td>$\frac{0}{97}$</td><td>$\frac{0}{97}$</td><td>$\frac{27}{29}$</td><td>$\frac{0}{29}$</td><td>$\frac{2}{29}$</td><td>$\frac{0}{29}$</td></tr>
<tr><td colspan="4">**Accuracy** = 1.0000</td><td colspan="4">**Accuracy** = 0.9310</td></tr>
<tr><td colspan="4">**Average time taken to detect face** = 0.1494 s</td><td colspan="4">**Average time taken to recognize face** = 2.1604 s</td></tr>
<tr><td rowspan="3">(15) Yaw $(+90°)$
</td><td>$\frac{1}{77}$</td><td>$\frac{0}{77}$</td><td>$\frac{76}{77}$</td><td>$\frac{0}{77}$</td><td>$\frac{0}{1}$</td><td>$\frac{0}{1}$</td><td>$\frac{1}{1}$</td><td>$\frac{0}{1}$</td></tr>
<tr><td colspan="4">**Accuracy** = 0.0130</td><td colspan="4">**Accuracy** = 0.0000</td></tr>
<tr><td colspan="4">**Average time taken to detect face** = 0.1557 s</td><td colspan="4">**Average time taken to recognize face** = 3.5685 s</td></tr>
</table>

4.1.3 With Occlusions

The developed face recognition system was tested with different occlusions. Occlusions refer to case that the whole face is not available in the image or video.

Table 4.3: Testing Results of The Developed Face Recognition System with Different Occlusions.

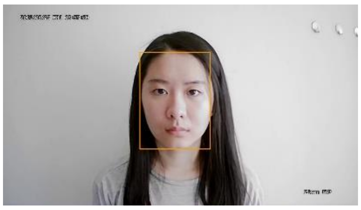
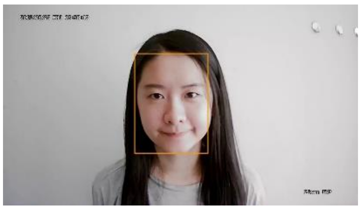
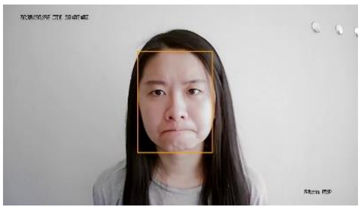
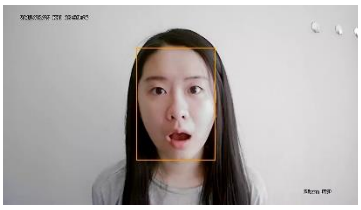
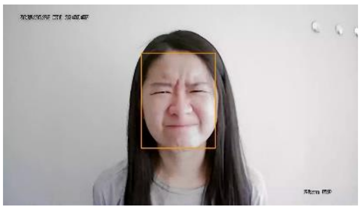
<table>
<tr><th rowspan="2">Occlusions</th><th colspan="4">Face Detection</th><th colspan="4">Face Recognition</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th></tr>
<tr><td rowspan="3">(1) No Occlusions
</td><td>$\frac{204}{263}$</td><td>$\frac{59}{263}$</td><td>$\frac{0}{263}$</td><td>$\frac{0}{263}$</td><td>$\frac{196}{204}$</td><td>$\frac{0}{204}$</td><td>$\frac{8}{204}$</td><td>$\frac{0}{204}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9608</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1538 s</td><td colspan="4">Average time taken to recognize face = 1.8509 s</td></tr>
<tr><td rowspan="3">(2) With Spectacles
</td><td>$\frac{116}{142}$</td><td>$\frac{26}{142}$</td><td>$\frac{0}{142}$</td><td>$\frac{0}{142}$</td><td>$\frac{108}{116}$</td><td>$\frac{0}{116}$</td><td>$\frac{8}{116}$</td><td>$\frac{0}{116}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9310</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1468 s</td><td colspan="4">Average time taken to recognize face = 2.5273 s</td></tr>
<tr><td rowspan="3">(3) With Hood
</td><td>$\frac{193}{198}$</td><td>$\frac{5}{198}$</td><td>$\frac{0}{198}$</td><td>$\frac{0}{198}$</td><td>$\frac{185}{193}$</td><td>$\frac{0}{193}$</td><td>$\frac{8}{193}$</td><td>$\frac{0}{193}$</td></tr>
<tr><td colspan="4">Accuracy = 0.1475 s</td><td colspan="4">Accuracy = 0.9585</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1475 s</td><td colspan="4">Average time taken to recognize face = 2.0588 s</td></tr>
<tr><td rowspan="3">(4) With Phone
</td><td>$\frac{134}{162}$</td><td>$\frac{28}{162}$</td><td>$\frac{0}{162}$</td><td>$\frac{0}{162}$</td><td>$\frac{131}{134}$</td><td>$\frac{0}{134}$</td><td>$\frac{3}{134}$</td><td>$\frac{0}{134}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9776</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1447 s</td><td colspan="4">Average time taken to recognize face =2.7087 s</td></tr>
</table>

<table>
<tr><td rowspan="3">(5) Hands Cover Mouth
</td><td>$\frac{136}{150}$</td><td>$\frac{14}{150}$</td><td>$\frac{0}{150}$</td><td>$\frac{0}{150}$</td><td>$\frac{96}{136}$</td><td>$\frac{0}{136}$</td><td>$\frac{40}{136}$</td><td>$\frac{0}{136}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.7059</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1356 s</td><td colspan="4">Average time taken to recognize face = 2.5639 s</td></tr>
<tr><td rowspan="3">(6) Hands Cover Eyes
</td><td>$\frac{44}{195}$</td><td>$\frac{15}{195}$</td><td>$\frac{136}{195}$</td><td>$\frac{0}{195}$</td><td>$\frac{4}{44}$</td><td>$\frac{0}{44}$</td><td>$\frac{40}{44}$</td><td>$\frac{0}{44}$</td></tr>
<tr><td colspan="4">Accuracy = 0.2256</td><td colspan="4">Accuracy = 0.0909</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1448 s</td><td colspan="4">Average time taken to recognize face =3.7967 s</td></tr>
</table>

4.1.4 With Different Facial Expressions

The developed face recognition system was tested with different facial expressions.

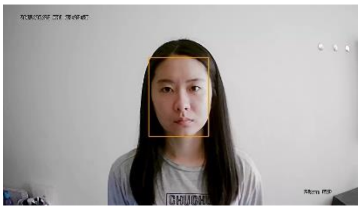
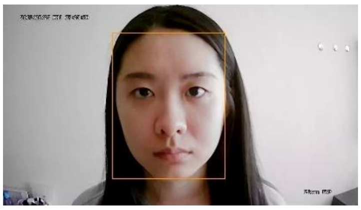
Table 4.4: Testing Results of The Developed Face Recognition System with Different Facial Expressions.

Facial Expressions	Face Detection				Face Recognition			
	TP	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
(1) Neutral 	$\frac{158}{201}$	$\frac{43}{158}$	$\frac{0}{201}$	$\frac{0}{201}$	$\frac{156}{158}$	$\frac{0}{158}$	$\frac{2}{158}$	$\frac{0}{158}$
	Accuracy = 1.0000				**Accuracy** = 0.9873			
	Average time taken to detect face = 0.1595 s				**Average time taken to recognize face** = 2.1025 s			
(2) Smile 	$\frac{89}{113}$	$\frac{24}{113}$	$\frac{0}{113}$	$\frac{0}{113}$	$\frac{87}{89}$	$\frac{0}{89}$	$\frac{2}{89}$	$\frac{0}{89}$
	Accuracy = 1.0000				**Accuracy** = 0.9775			
	Average time taken to detect face = 0.1597 s				**Average time taken to recognize face** = 2.3194 s			
(3) Sadness 	$\frac{70}{102}$	$\frac{32}{102}$	$\frac{0}{102}$	$\frac{0}{102}$	$\frac{66}{70}$	$\frac{0}{70}$	$\frac{4}{70}$	$\frac{0}{70}$
	Accuracy = 1.0000				**Accuracy** = 0.9429			
	Average time taken to detect face = 0.1550 s				**Average time taken to recognize face** = 2.3846 s			
(4) Surprise 	$\frac{168}{175}$	$\frac{7}{175}$	$\frac{0}{175}$	$\frac{0}{175}$	$\frac{152}{168}$	$\frac{0}{168}$	$\frac{16}{168}$	$\frac{0}{168}$
	Accuracy = 1.0000				**Accuracy** = 0.9048			
	Average time taken to detect face = 0.1482 s				**Average time taken to recognize face** = 2.4850 s			
(5) Disgust 	$\frac{162}{201}$	$\frac{39}{201}$	$\frac{0}{201}$	$\frac{0}{201}$	$\frac{57}{162}$	$\frac{0}{162}$	$\frac{105}{162}$	$\frac{0}{162}$
	Accuracy = 1.0000				**Accuracy** = 0.3519			
	Average time taken to detect face = 0.1478 s				**Average time taken to recognize face** = 2.6018 s			

4.1.5 With Different Face Sizes

The developed face recognition system was tested with different face size. The distance from the camera to the person will cause the varying of face size. When the distance from the camera to the person is about 100.0 cm, the face of the person is considered as a small face. When the distance from the camera to the person is about 60.0 cm, the face of the person is considered as a medium face. When the distance between the camera and the person is about 30.0 cm, the face of the person is considered as a large face.

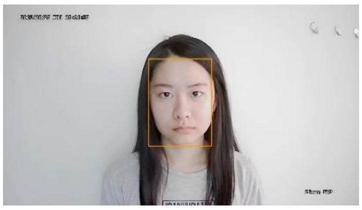
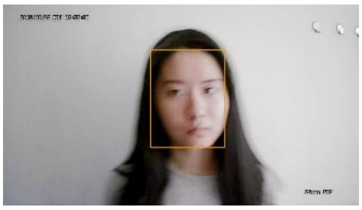
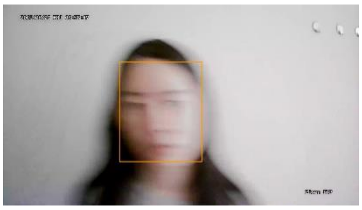
Table 4.5: Testing Results of The Developed Face Recognition System with Different Face Sizes.

<table>
<tr><th rowspan="2">Face Size</th><th colspan="4">Face Detection</th><th colspan="4">Face Recognition</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th></tr>
<tr><td rowspan="3">(1) Small Face
</td><td>$\frac{254}{280}$</td><td>$\frac{24}{280}$</td><td>$\frac{2}{280}$</td><td>$\frac{0}{280}$</td><td>$\frac{184}{254}$</td><td>$\frac{0}{254}$</td><td>$\frac{70}{254}$</td><td>$\frac{0}{254}$</td></tr>
<tr><td colspan="4">Accuracy = 0.9929</td><td colspan="4">Accuracy = 0.7244</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1322 s</td><td colspan="4">Average time taken to recognize face = 1.5609 s</td></tr>
<tr><td rowspan="3">(2) Medium Face
</td><td>$\frac{131}{148}$</td><td>$\frac{17}{148}$</td><td>$\frac{0}{148}$</td><td>$\frac{0}{148}$</td><td>$\frac{116}{131}$</td><td>$\frac{0}{131}$</td><td>$\frac{15}{131}$</td><td>$\frac{0}{131}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.8855</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1439 s</td><td colspan="4">Average time taken to recognize face = 2.0341 s</td></tr>
<tr><td rowspan="3">(3) Large Face
</td><td>$\frac{110}{158}$</td><td>$\frac{48}{158}$</td><td>$\frac{0}{158}$</td><td>$\frac{0}{158}$</td><td>$\frac{110}{110}$</td><td>$\frac{0}{110}$</td><td>$\frac{0}{110}$</td><td>$\frac{0}{110}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 1.0000</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1437 s</td><td colspan="4">Average time taken to recognize face = 2.4576 s</td></tr>
</table>

4.1.6 With Different Motions

The developed face recognition system was tested with different motions.

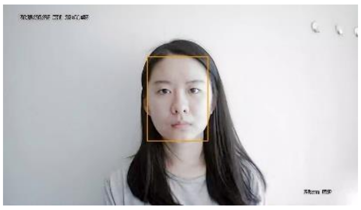
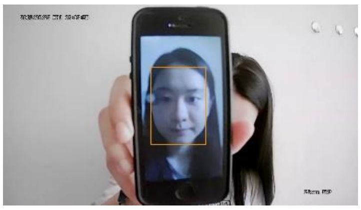
Table 4.6: Testing Results of The Developed Face Recognition System with Different Motions.

<table>
<tr><th rowspan="2">Motions</th><th colspan="4">Face Detection</th><th colspan="4">Face Recognition</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th></tr>
<tr><td rowspan="3">(1) No Motion
</td><td>$\frac{347}{388}$</td><td>$\frac{41}{388}$</td><td>$\frac{0}{388}$</td><td>$\frac{0}{388}$</td><td>$\frac{342}{347}$</td><td>$\frac{0}{347}$</td><td>$\frac{5}{347}$</td><td>$\frac{0}{347}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9856</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1541 s</td><td colspan="4">Average time taken to recognize face = 2.2114 s</td></tr>
<tr><td rowspan="3">(2) Slow
</td><td>$\frac{74}{151}$</td><td>$\frac{75}{151}$</td><td>$\frac{2}{151}$</td><td>$\frac{0}{151}$</td><td>$\frac{42}{74}$</td><td>$\frac{0}{74}$</td><td>$\frac{32}{74}$</td><td>$\frac{0}{74}$</td></tr>
<tr><td colspan="4">Accuracy = 0.9868</td><td colspan="4">Accuracy = 0.5676</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1529 s</td><td colspan="4">Average time taken to recognize face = 2.7323 s</td></tr>
<tr><td rowspan="3">(3) Fast
</td><td>$\frac{59}{119}$</td><td>$\frac{28}{119}$</td><td>$\frac{32}{119}$</td><td>$\frac{0}{119}$</td><td>$\frac{2}{59}$</td><td>$\frac{0}{59}$</td><td>$\frac{57}{59}$</td><td>$\frac{0}{59}$</td></tr>
<tr><td colspan="4">Accuracy = 0.7311</td><td colspan="4">Accuracy = 0.0339</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1491 s</td><td colspan="4">Average time taken to recognize face = 2.7400 s</td></tr>
</table>

4.1.7 With Spoofing Face

The developed face recognition system was tested with spoofing face. A spoofing face can be a photograph or a video recording that shows the facial image of a person to the camera in order to be falsely authenticated by the face recognition system.

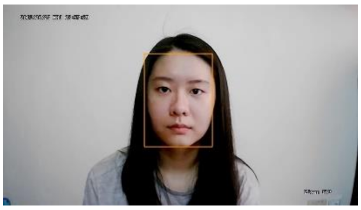
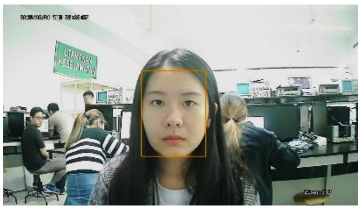
Table 4.7: Testing Results of The Developed Face Recognition System with Spoofing Face.

<table>
<tr><th rowspan="2">Spoofing Faces</th><th colspan="4">Face Detection</th><th colspan="4">Face Recognition</th></tr>
<tr><th>TP</th><th>TN</th><th>FP</th><th>FN</th><th>TP</th><th>TN</th><th>FP</th><th>FN</th></tr>
<tr><td rowspan="3">(1) Real Face
</td><td>$\frac{235}{257}$</td><td>$\frac{22}{257}$</td><td>$\frac{0}{257}$</td><td>$\frac{0}{257}$</td><td>$\frac{231}{235}$</td><td>$\frac{0}{235}$</td><td>$\frac{4}{235}$</td><td>$\frac{0}{235}$</td></tr>
<tr><td colspan="4">Accuracy = 1.0000</td><td colspan="4">Accuracy = 0.9830</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1466 s</td><td colspan="4">Average time taken to recognize face = 1.7950 s</td></tr>
<tr><td rowspan="3">(2) Spoofing Face
</td><td>$\frac{161}{186}$</td><td>$\frac{25}{186}$</td><td>$\frac{1}{186}$</td><td>$\frac{0}{186}$</td><td>$\frac{146}{161}$</td><td>$\frac{0}{161}$</td><td>$\frac{15}{161}$</td><td>$\frac{0}{161}$</td></tr>
<tr><td colspan="4">Accuracy = 0.9946</td><td colspan="4">Accuracy = 0.9068</td></tr>
<tr><td colspan="4">Average time taken to detect face = 0.1770 s</td><td colspan="4">Average time taken to recognize face = 2.1495 s</td></tr>
</table>

4.1.8 With Different Backgrounds

The developed face recognition system was tested with different backgrounds.

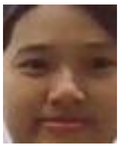
Table 4.8: Testing Results of The Developed Face Recognition System with Different Backgrounds.

Backgrounds	**Face Detection**				**Face Recognition**			
	TP	**TN**	**FP**	**FN**	**TP**	**TN**	**FP**	**FN**
(1) Clean Background 	$\frac{100}{100}$	$\frac{0}{100}$	$\frac{0}{100}$	$\frac{0}{100}$	$\frac{93}{100}$	$\frac{0}{100}$	$\frac{7}{100}$	$\frac{0}{100}$
	Accuracy = 1.0000				**Accuracy** = 0.9300			
	Average time taken to detect face = 0.0937 s				**Average time taken to recognize face** = 1.5343 s			
(2) Complex Background 	$\frac{93}{100}$	$\frac{0}{100}$	$\frac{1}{100}$	$\frac{6}{100}$	$\frac{86}{93}$	$\frac{0}{93}$	$\frac{7}{93}$	$\frac{0}{93}$
	Accuracy = 0.9300				**Accuracy** = 0.9247			
	Average time taken to detect face = 0.0963 s				**Average time taken to recognize face** = 2.2271 s			

4.1.9 With Different Faces

The developed face recognition system was tested with different faces. Total six (6) persons had involved in this experiment. Each of them has 5 to 10 images in the dataset.

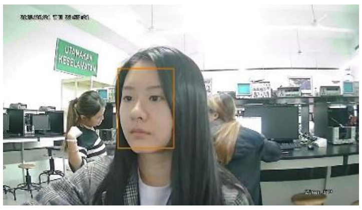
Table 4.9: Testing Results of The Developed Face Recognition System with Different Faces.

Candidate	Face Detection	Face Recognition
(1) 	Detected	Recognized Correctly
(2) 	Detected	Recognized Correctly
(3) 	Detected	Recognized Correctly
(4) 	Detected	Recognized Correctly
(5) 	Detected	Recognized Correctly
(6) 	Detected	Recognized Correctly

4.1.10 With Multiple Faces

The developed face recognition system was tested with multiple faces.

Table 4.10: Testing Results of The Developed Face Recognition System with Multiple Faces.

Number of Faces	Face Detection	Face Recognition
(1) Single Face 	Detected	Recognized Correctly
(2) Multiple Faces 	All Detected	All Recognized Correctly

4.2 Discussion

In the developed face recognition system, the face recognition/ verification is performed by comparing the face encoding of the detected face with the database of previously enrolled known faces encodings. The training image of each individual with latest appearance is required to be considered as a significant factor for the face recognition system.

The data gathering system is developed to gather the face images and create our own face recognition dataset. Each individual in our own dataset consists of 1 to 10 images. Other than our own dataset, Labeled Faces in the Wild (LFW) dataset also have been used for the database. LFW is a face image database which is designed for studying the problem of unconstrained face recognition. This database was created and maintained by researchers at the University of Massachusetts, Amherst, which holds about 12,900 faces images that collected from the web. Each face has been labelled with the name of the person pictured. Approximately 5000 people are pictured in the dataset with two or more distinct photos.

Once the database has been readily trained and prepared, the face recognition system can be performed. Compared with controlled environments, images or video frames from uncontrolled environments hold more variation in pose, lighting, expression, occlusion, background, scale and so-on. Therefore, face recognition in uncontrolled environments is much more challenging than in controlled conditions. Therefore, several experiments were conducted to test the performance of the developed face recognition system in different conditions. For example, the face recognition system had been tested with different lighting conditions, with different head pose variations, with occlusions, with different facial expressions, with different face sizes, with different motion, with spoofing face, with different faces and with multiple faces.

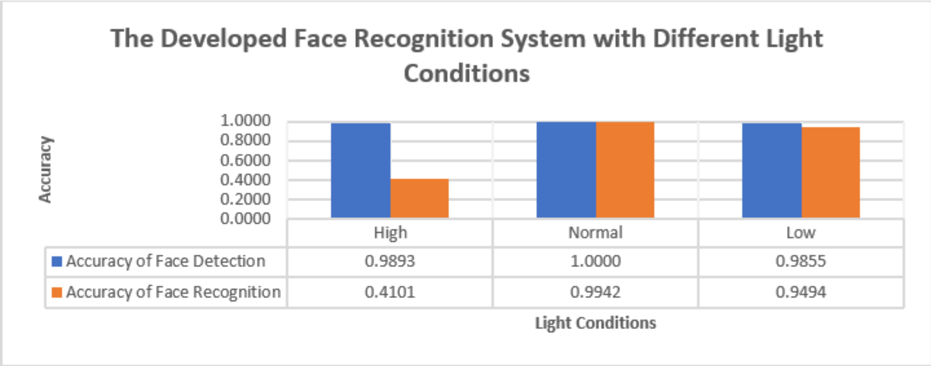
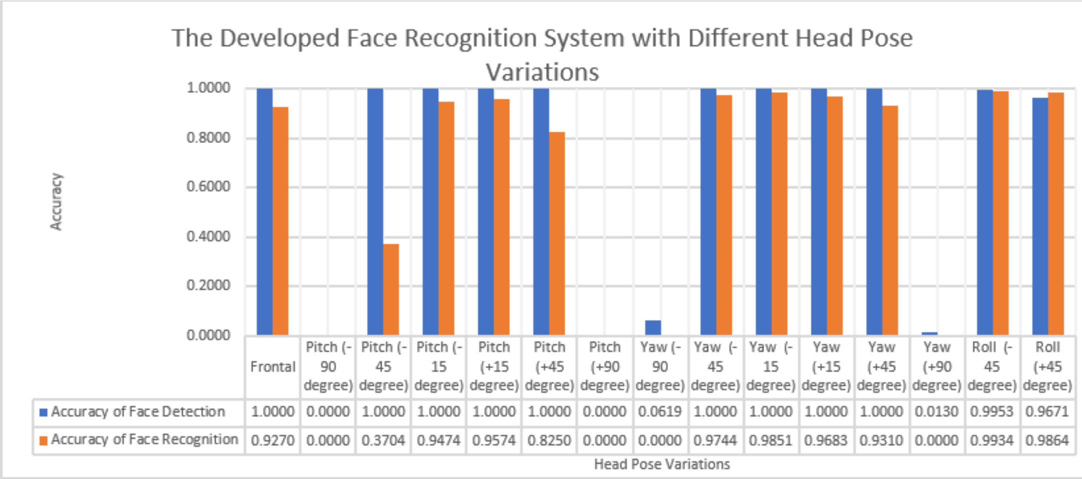


Figure 4.4: Test Results of The Developed Face Recognition System with Different Light Conditions.

The data above were collected and generated from the results obtained from the developed face recognition system with different light conditions (high, normal and low light conditions).

The face detection has average accuracy of 99.16 % in different light conditions. Since the CCTV camera used in the system is a day and night camera, the face detection and recognition can be performed well in a low light condition. However, the high lighting condition has weakened certain face features and causes the decreasing of the accuracy in face recognition.



	Frontal	Pitch (-90 degree)	Pitch (-45 degree)	Pitch (-15 degree)	Pitch (+15 degree)	Pitch (+45 degree)	Pitch (+90 degree)	Yaw (-90 degree)	Yaw (-45 degree)	Yaw (-15 degree)	Yaw (+15 degree)	Yaw (+45 degree)	Yaw (+90 degree)	Roll (-45 degree)	Roll (+45 degree)
Accuracy of Face Detection	1.0000	0.0000	1.0000	1.0000	1.0000	1.0000	0.0000	0.0619	1.0000	1.0000	1.0000	1.0000	0.0130	0.9953	0.9671
Accuracy of Face Recognition	0.9270	0.0000	0.3704	0.9474	0.9574	0.8250	0.0000	0.0000	0.9744	0.9851	0.9683	0.9310	0.0000	0.9934	0.9864

Figure 4.5: Test Results of The Developed Face Recognition System with Different Head Pose Variations.

The data above were collected and generated from the results obtained from the developed face recognition system with different head pose variations. The head was oriented in term of pitch, roll and yaw movements in the experiment.

The empirical result shows the effect of the head pose variations on the performance of the face recognition system. 99.66 % of average accuracy of face detection could be achieved in different pose variations with maximum 45 degree. Since the participant involved in this experiment has the images with different face angles in the database, the face recognition could be performed better in small angle non-frontal faces.

On the other hand, the face detection had performed worse (approximately 0.00 %) when the head was in $-90°$ pitch movement, $+90°$ pitch movement, $-90°$ yaw movement and $+90°$ yaw movement. As mentioned early, face detection is the first and essential step before face recognition. Therefore, face recognition was not performed for these head orientation movement as the face was not detected.

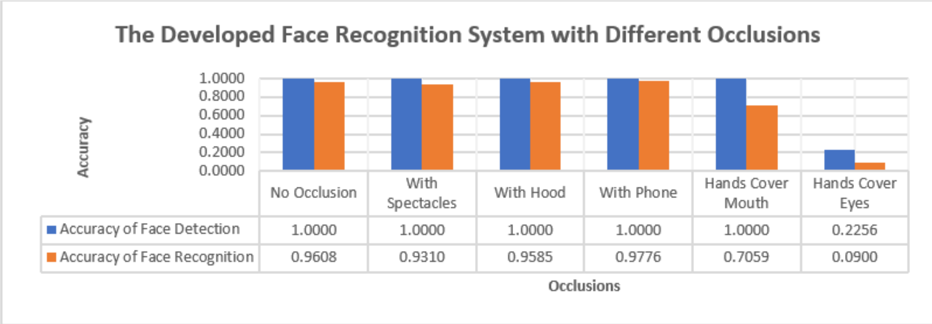


Figure 4.6: Test Results of The Developed Face Recognition System with Different Occlusions.

The data shown in Figure 4.6 were collected and generated from the results obtained from the developed face recognition system with different occlusions on face. Occlusion is one critical issue that affects the performance of face recognition.

The performance of face detection and face recognition on small occluded face is high. These small occlusions include a person talking on the phone or wearing glasses, caps or hood.

The face partially covered with hand still can be detected, but low performance on the face recognition.

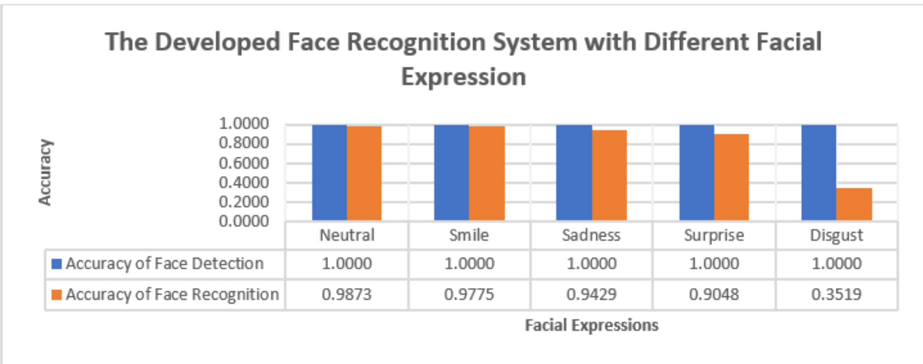


Figure 4.7: Test Results of The Developed Face Recognition System with Different Facial Expressions.

The data shown in Figure 4.7 were collected and generated from the results obtained from the developed face recognition system with different facial expressions such as smile, sadness, surprise and disgust.

Face with different facial expressions can be fully detected. Face recognition have also performed well on the facial expressions of smile, sadness and surprise. However, the facial expression of disgust changes the face features severely and leads to the low performance of face recognition.

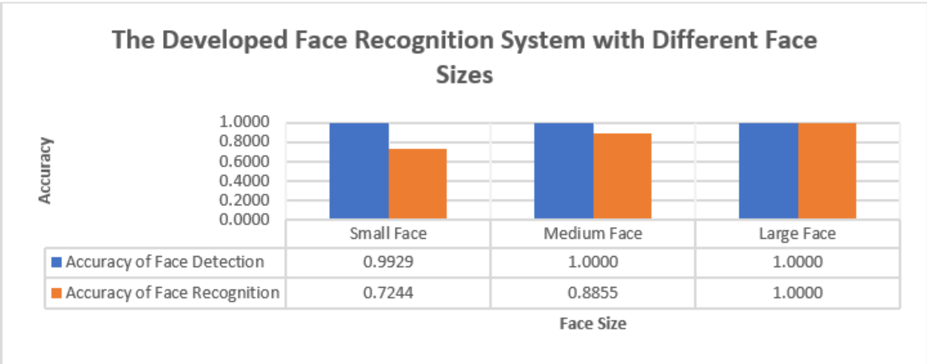


Figure 4.8: Test Results of The Developed Face Recognition System with Different Face Sizes.

The data shown in Figure 4.8 were collected and generated from the results obtained from the developed face recognition system with different face sizes. The varying of the face size depends on the distance from camera to the person. When the distance from the camera to the person becomes smaller, the face size will become larger.

The face detection method used in the system can detect the smaller faces. From the data above, it can be concluded that the larger faces provide a more accurate set of face recognition results.

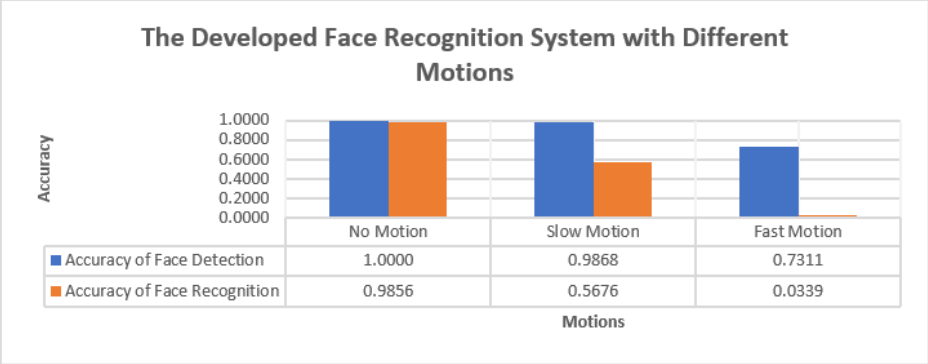


Figure 4.9: Test Results of The Developed Face Recognition System with Different Motions.

The data shown in Figure 4.9 were collected and generated from the results obtained from the developed face recognition system with motions. The motion here means the moving speed of the face appears on the video frame. A normal CCTV camera usually is not able to capture the face when a person is walking quickly.

Low performance of face recognition is achieved for the fast motion. This is because the fast motion causes the captured faces become blur. In the other words, the facial features cannot be extracted perfectly on the blur face.

Face detection and face recognition can be obtained in high accuracy when the person stay motionless in front of the camera.

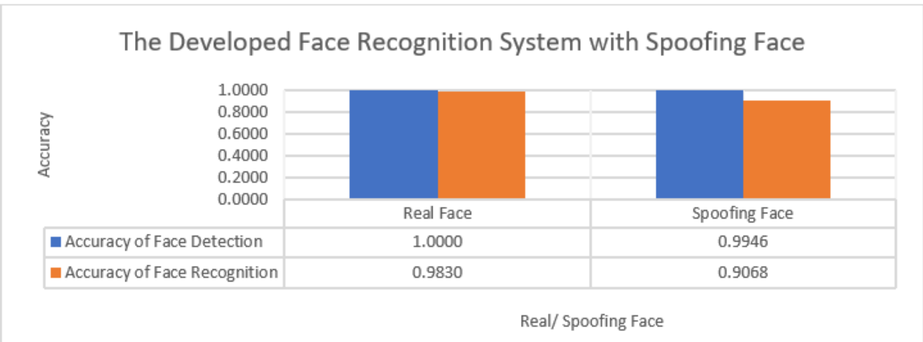


Figure 4.10: Test Results of The Developed Face Recognition System with Spoofing Face.

The data shown in Figure 4.10 were collected and generated from the results obtained from the developed face recognition system with spoofing face.

The results show the developed face recognition system can detect and recognize the spoofing face.

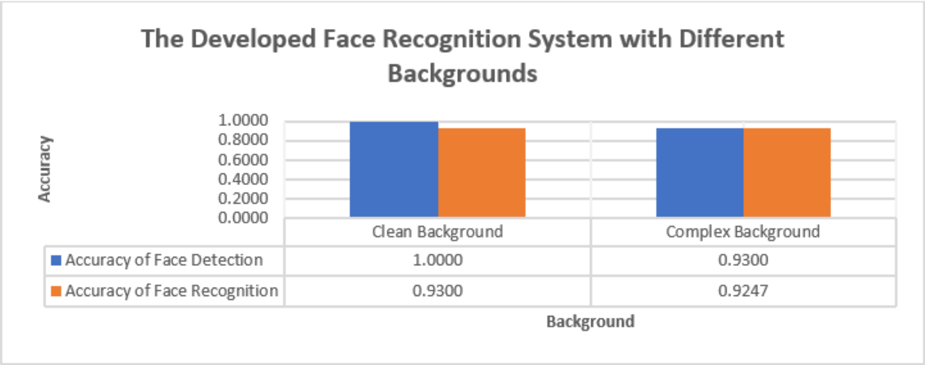


Figure 4.11: Test Results of The Developed Face Recognition System with Different Backgrounds.

The data shown in Figure 4.11 were collected and generated from the results obtained from the developed face recognition system with different backgrounds.

Some detection error was presented in the complex background. However, the complex background did not bring a huge effect on the face detection and face recognition. It still achieved a high accuracy (more than 90.0 %) for face detection and face recognition when the background is complex.

Consequently, the developed face recognition system has different performance in different conditions. Table below simply summarise the worse and the best conditions for the developed face recognition system.

Table 4.11: Summary of the Performance of the Developed System in Different Conditions.

Performance of the System	Conditions
Low	• High lighting condition • Head in $\pm 90°$ pitch movement • Head in $\pm 90°$ yaw movement • Partially occluded face • Facial expression of disgust • Smaller face size • Fast motion of face • Complex background
High	• Normal lighting condition • Frontal face • Face with no occlusion • Neutral face expression • Larger face size • No motion of face • Clean background

The developed face recognition had also tested with multiple faces and with different participants' face. In the case of multiple faces, the four faces that appeared on the frame of the real-time video had been successful detected and recognized correctly. However, the multiple faces might require more time to recognized all faces. In the case of different faces, six participants had involved in that experiment and each of them has 1 to 10 images in the database. Each participants' face had also been successful detected and recognized correctly.

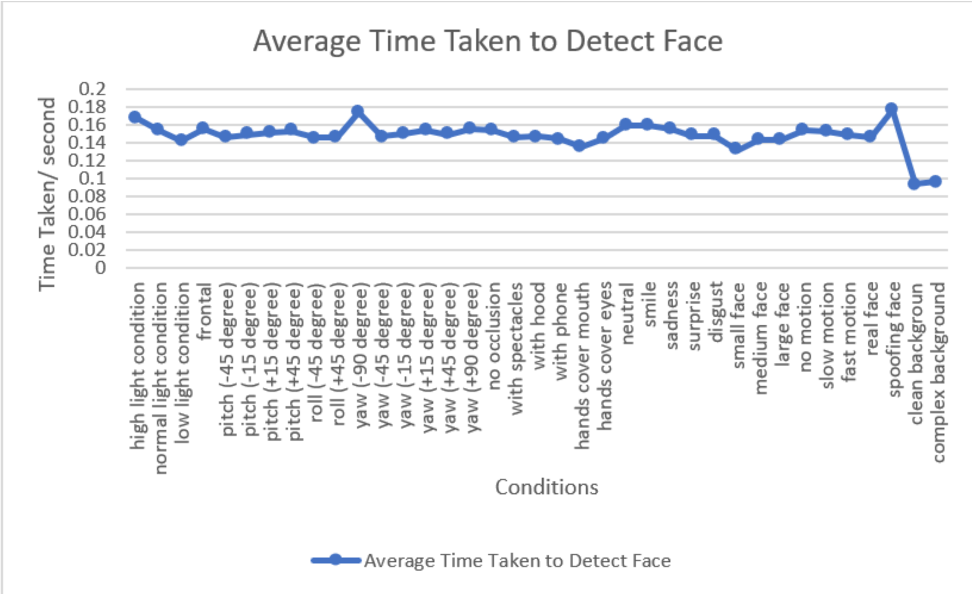


Figure 4.12: Average Time Taken to Detect a Face in Different Conditions.

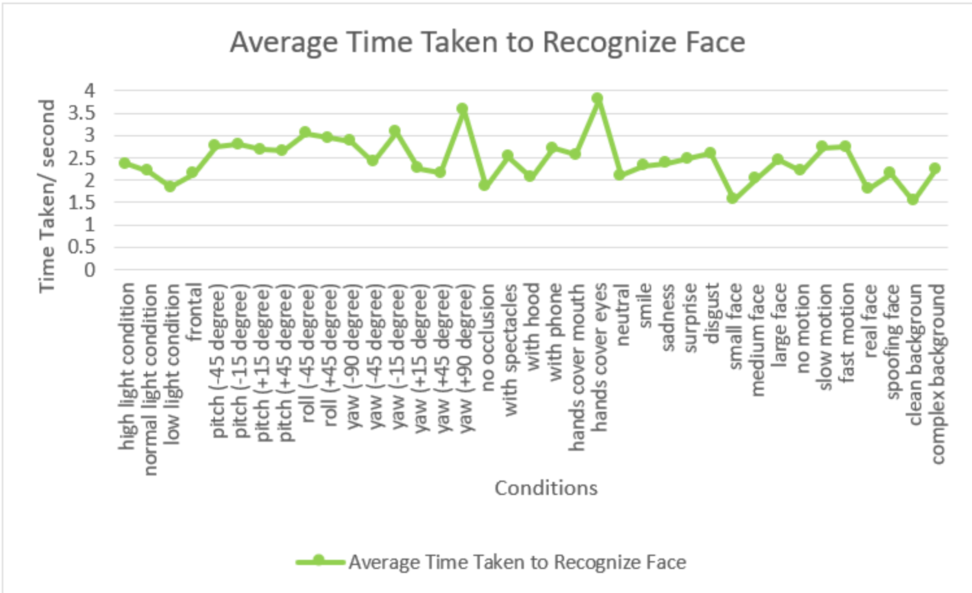


Figure 4.13: Average Time Taken to Recognize a Face in Different Conditions.

The line chart shown in Figure 4.11 and 4.12 states the average time taken to detect face and recognize face in different conditions that have mentioned and tested previously using the developed face recognition system. The overall average time taken for face detection in the developed system is about 0.15 second. And the overall average time taken for face recognition in the developed system is about 2.50 second.

4.3 Limitations

After conducting several experiments, some of the limitations in the developed system can be noticed. First, the face detection and face recognition have a low performance in a high lighting condition, condition of head orientation with 90 degrees and condition of smaller face. Besides, the CCTV camera used in the developed system could not capture a fast motion object. Blurred faces would be obtained when a person walks by the camera quickly. As such, the performance of the face recognition will become weaker.

Furthermore, the developed face recognition system can detect and recognize the spoofing face. It means that the face recognition system can falsely authenticate when someone showing any photograph or video recording that contains face to the camera.

In addition, the latency in the IP CCTV camera is one of the limitations of the developed system. Latency in the network video surveillance context is the amount of time between the instant a frame is captured from the camera source and the instant that frame is displayed. There was about 1-2 second time lag between the CCTV camera and the display in the developed system.

The frame rate of the output display is low. Although the multiple thread technique has been used in the developed system to speed up the video processing, the frame rate of the output display still maintained at an average value of 9. It may due to the weak performance of the laptop’s processor.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The objective of this project is to implement a face recognition system with CCTV camera using deep learning. The developed system should able to detect and identify face in a real-time video. With recent advancements in deep learning, high accuracy of face recognition can be achieved to the real-world variations.

The system has then tested with different conditions based on light conditions, head pose variations, occlusions, facial expressions, face sizes, motion, spoofing face, different faces and multiple faces. The performance of the system varies among these different conditions. As a conclusion for the analysis, the face recognition for a large frontal face with no occlusion and neutral facial expression in a reasonable lighting condition would achieve the highest performance of the system. In the other words, the face recognition in uncontrolled environments is much more challenging than in controlled conditions.

5.2 Recommendation

Many real-world applications require good recognition performance in uncontrolled environments. A camera specially designed for facial recognition will be a better choice as compared to a normal CCTV camera. The specially designed camera can capture the angle of the face that is suitable for face recognition. It should able to capture the high-quality face images and also able to prevent the capturing of blurred face images.

Besides, future work can focus on spoofing attack. Spoofing attack is an attempt to acquire the privileges or access rights of someone by using a photo, video or another replacement for an authorized person's face. Especially when the biometric technologies continue to grow year after year in such different environments, the safety weaknesses of users has become better known to the general public.

A strong processor and larger memory space are required in deep learning. This is because deep leaning is very computationally intensive. An expensive high-performance system may require in building a deep learning system. For example, the GPU cannot be leave out when building a system for deep learning.

Another future work can focus on the the ways to improve the face recognition system efficiency in uncontrolled environments.

REFERENCES

Adam, G. (2016) *Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning*. Available at: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78 (Accessed: 28 July 2019).

Adam, G. (2018) *Face Recognition Accuracy Problems · ageitgey/face_recognition Wiki · GitHub*. Available at: https://github.com/ageitgey/face_recognition/wiki/Face-Recognition-Accuracy-Problems (Accessed: 29 July 2019).

Ali, T. (2010) *Face Recognition: An Introduction*. Available at: https://alitarhini.wordpress.com/2010/12/05/face-recognition-an-introduction/ (Accessed: 22 June 2019).

Arcoverde Neto, E. N. *et al.* (2014) 'Enhanced real-time head pose estimation system for mobile device', *Integrated Computer-Aided Engineering*, 21(3), pp. 281–293. doi: 10.3233/ICA-140462.

Brownlee, J. (2019) *How to Develop a Face Recognition System Using FaceNet in Keras*. Available at: https://machinelearningmastery.com/how-to-develop-a-face-recognition-system-using-facenet-in-keras-and-an-svm-classifier/ (Accessed: 28 July 2019).

Cai, Z. *et al.* (2018) 'Joint Head Pose Estimation with Multi-task Cascaded Convolutional Networks for Face Alignment', *Proceedings - International Conference on Pattern Recognition*. IEEE, 2018–Augus, pp. 495–500. doi: 10.1109/ICPR.2018.8545898.

Dang, K. and Sharma, S. (2017) 'Review and comparison of face detection algorithms', *Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering*. IEEE, pp. 629–633. doi: 10.1109/CONFLUENCE.2017.7943228.

Datta, A. K., Datta, M. and Banerjee, P. K. (2015) *Face Detection and Recognition: Theory and Practice*. CRC Press, 2015.

Davis, K. (2017) *dlib C++ Library: High Quality Face Recognition with Deep Metric Learning*. Available at: http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html (Accessed: 28 July 2019).

Enriquez, K. (2018) 'Faster Face Detection using Convolutional Neural Networks & the Viola-Jones Algorithm', (May).

Florian, S., Dmitry, K. and James, P. (2015) 'FaceNet: A Unified Embedding for Face Recognition and Clustering', pp. 815–823. doi: 10.1109/CVPR.2015.7298682.

Google Trends (2019). Available at: https://trends.google.com/trends/ (Accessed: 23 June 2019).

Hanis, Z. (2019) *Grab implements passenger verification using face recognition tech - Nation | The Star Online, Star Online*. Available at: https://www.thestar.com.my/news/nation/2019/04/11/grab-implementing-passenger-verification-using-face-recognition-tech/ (Accessed: 26 July 2019).

Jian, Z. *et al.* (2014) *A Collaborative Linear Discriminative Representation Classification Method for Face Recognition*. Lancester: DEStech Publications.

Katie, H. (2013) *Photographer Compares Aging Faces Across 60 Years*. Available at: https://mymodernmet.com/ana-oliveira-identities/ (Accessed: 22 June 2019).

Ma, M. and Wang, J. (2019) 'Multi-View Face Detection and Landmark Localization Based on MTCNN', *Proceedings 2018 Chinese Automation Congress, CAC 2018*. IEEE, pp. 4200–4205. doi: 10.1109/CAC.2018.8623535.

Malikovich, K. M., Ugli, I. S. Z. and O'Ktamovna, D. L. (2017) 'Problems in face recognition systems and their solving ways', *2017 International Conference on Information Science and Communications Technologies, ICISCT 2017*, 2017–Decem, pp. 1–4. doi: 10.1109/ICISCT.2017.8188594.

Olszewska, J. I. (2016) 'Automated Face Face Recognition : Challenges and and Solutions', pp. 59–79. doi: 10.5772/66013.

Opalyn, M. (2019) *Penang launches country's first facial recognition CCTV surveillance | Malaysia | Malay Mail*, *malaymail*. Penang. Available at: https://www.malaymail.com/news/malaysia/2019/01/02/penang-launches-countrys-first-facial-recognition-cctv-surveillance/1708422 (Accessed: 3 July 2019).

Rajeshwari, D. G. and Prof. Anala, M. R. (2015) 'Implementation and Analysis of Face Detection', (7), pp. 108–113.

Rewagad, P. P. *et al.* (2013) 'Critical Review on Face Recognition Algorithms', 2013, pp. 29–32. doi: http://dx.doi.org/10.5772/66013.

Sharifara, A. *et al.* (2014) 'A General Review of Human Face Detection Including a Study of Neural Networks and Haar Feature-based Cascade Classifier in Face Detection', *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*. IEEE, pp. 73–78. doi: 10.1109/ISBAST.2014.7013097.

Sharmila, N. (2019) *AirAsia uses facial recognition for self check-in and boarding at Senai Airport*, *Star Online*. Available at: https://www.thestar.com.my/tech/tech-news/2019/04/26/register-your-face-on-airasia-app-for-a-seamless-experience-at-senai-airport/ (Accessed: 26 July 2019).

Simon, D. (2015) *MasterCard will begin trials of face recognition software to authorize payments – let the travesty begin! » The Privacy Surgeon*. Available at: http://www.privacysurgeon.org/blog/incision/mastercard-will-begin-trials-of-face-recognition-software-to-authorize-payments-let-the-travesty-begin/ (Accessed: 26 July 2019).

Srinath, K. R. (2017) 'Python-The Fastest Growing Programming Language', *International Research Journal of Engineering and Technology*. Available at: www.irjet.net (Accessed: 26 July 2019).

Sun, Y., Wang, X. and Tang, X. (2015) 'Deeply learned face representations are sparse, selective, and robust', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07–12–June, pp. 2892–2900. doi: 10.1109/CVPR.2015.7298907.

Szegedy, C. *et al.* (2014) 'Going Deeper with Convolutions'.

Taigman, Y. *et al.* (2014) 'DeepFace: Closing the gap to human-level performance in face verification', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708. doi: 10.1109/CVPR.2014.220.

Trigueros, D. S., Meng, L. and Hartnett, M. (2018) 'Face Recognition: From Traditional to Deep Learning Methods'. Available at: http://arxiv.org/abs/1811.00116.

Vinay, P. (2019) *10 Different Types of CCTV Cameras and Their Purpose*. Available at: https://www.techprevue.com/cctv-cameras-different-types-purpose/ (Accessed: 26 July 2019).

Zeiler, M. D. and Fergus, R. (2014) 'Visualizing and Understanding Convolutional Networks', pp. 818–833.

Zhang, K. *et al.* (2016) 'Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks', *IEEE Signal Processing Letters*. IEEE, 23(10), pp. 1499–1503. doi: 10.1109/LSP.2016.2603342.

APPENDICES

APPENDIX A: Computer Programme Listing

Please refer to CD.