Project No: SCE17-0596

Face Recognition Based Attendance System

Submitted by: Mok Wen Jun
Matriculation Number: U1522226J

Supervisor: Dr Althea Liang Qianhui
Examiner: Assoc Prof Deepu Rajan

School of Computer Science & Engineering

A final year project report presented to the Nanyang Technological University
in partial fulfilment of the requirements of the degree of
Bachelor of Engineering (Computer Science)

2018

Table of Contents

Abstract

Face recognition systems have gained a considerable amount of attention as they can be used in many different applications. Applications where such systems are used include those for identification purposes in airports [1], in surveillance cameras to detect potential criminals, in computers and mobile phones as a form of biometric authentication, and in other forms of technology such as robotics. Face recognition systems work by comparing selected facial features from an image with faces in a training set. [2]

The objective of this project is to develop a face recognition based attendance system for students to take their attendance. Students will take their attendance by looking into the camera for their attendance to be taken. This will help to save the teachers time and effort as teachers will not need to spend time to take the attendance of the students and the attendance of up to five students can be taken at a time.

Other sub-tasks in this project include creating the back-end portion of the system where administrators can also test the face recognition accuracy and face recognition speed of the system by adjusting the various settings. The various techniques used in face detection and face recognition such as the Viola Jones detection framework, Principal Component Analysis and Eigenfaces were also being explored. The factors which affect the face recognition speed and face recognition accuracy were also determined. Other tasks include finding out how super-resolution techniques were able to improve the resolution of the students' faces to ensure that their faces can be recognized correctly. Experiments were also conducted to determine how the various settings affect the face recognition accuracy and face recognition speed. The face recognition application was developed using the C# programming language.

Acknowledgements

I would like to thank my supervisor, Dr Althea Liang, for allowing me to work on my Final Year Project on implementing a Face Recognition Based Attendance System which enables students to take their attendance via Face Recognition. I really enjoyed working on this interesting topic. I appreciate Dr Liang's effort in supervising and guiding me throughout this project. She is always willing to lend a helping hand whenever I encounter problems with my project. She is also able to provide creative and meaningful ideas for my project.

I would also like to thank my parents to allow me to pursue my studies at Nanyang Technological University and work on this project. Lastly, I would also like to thank everybody who gave me the support throughout the project.

Mok Wen Jun
October 2018

List of Figures

List of Tables

Chapter 1 Introduction

Face recognition systems are used to identify or verify an individual's identity using a digital image or a video frame from a video source [2]. Face recognition is important in many computer vision applications such as human-computer interactions and security systems. Video-based surveillance and monitoring systems have become more easily available due to the advancement in technology and they can make use of face recognition to help to track down people who commit criminal offences.

1.1 Motivations

Attendance plays an important role in our education system. It is important to monitor the students' attendance in a classroom to make sure that they come for lessons. Traditional methods of attendance taking involves teachers passing the attendance sheet around for students to sign their attendance or calling out names of the students individually to take their attendance. [3] This may create a lot of loopholes in the system as students may sign the attendance of another student who is not present in the classroom. There is a lot of paperwork involved and teachers need to maintain huge records. [1] There is also the possibility of records getting lost, stolen or damaged. [3] This way of taking attendance takes up a significant amount of the teachers' time which could instead be better used during lessons to impact knowledge to students. Teachers may also make mistakes when taking the students' attendance.
Hence, a face recognition based attendance system is being developed for this project. The face recognition based attendance system can help to speed up the attendance taking process as teachers do not need to take up time during lessons to take the students' attendance and up to five students can take their attendance at a time. Students will also need to be physically present in order for their attendance to be taken and are not able to take the attendance of another student on his or her behalf. In addition, the administrators will also not need to take up time printing and collating attendance sheets as all information are delivered in the form of digital data.

1.2 Objective

The objective of the face recognition based attendance system is to take the attendance of students in the classroom using face recognition. This will be done by detecting and recognizing the faces of students who belong to the class.

The face recognition based attendance system will be placed at the entrance of the classroom. Students can either choose to register their attendance individually or as a group of up to five students. Students will need to look at the camera placed at the entrance of the classroom before entering the classroom to get their attendance taken. When the students' faces are recognized, their attendance will be taken and recorded in the database. The time and date of the students who entered the classroom will also be recorded. Teachers can also use the application to view or modify the attendance records of the students. Administrators can use the application to collect more training face images of students and perform experiments to determine the face detection and face recognition accuracy by adjusting the various settings.

1.3 Report Organisation

Chapter 1 introduces this project by giving a brief introduction on what face recognition is. It also gives the background of the traditional methods of attendance taking in schools and the various problems which might surface using these methods. This chapter also states the objective of this project as well as giving a brief description on how this project is able to solve the problems which might occur using the traditional methods of attendance taking.

Chapter 2 describes the various applications which have been developed using face recognition. It also explains how face recognition is implemented using PCA (Principal Component Analysis) and Eigenfaces. How face detection techniques using the Viola Jones object detection framework is also explained in this section.

Chapter 3 describes the various ways the administrators can obtain the training images of the students for face recognition to be done.

Chapter 4 describes how students, teachers and administrators are able to use the face recognition based attendance system.

Chapter 5 gives an overview of the face recognition based attendance system including the architecture diagram of the system and the relationships between the individual modules of the system.

Chapter 6 describes the experiment set up process which includes how the training and testing images were obtained as well as how the experiment was being conducted.

Chapter 7 describes the experiment results obtained from the various face recognition experiments conducted and the findings which can be derived from the experiments.

Chapter 8 describes how super-resolution techniques can help to improve the resolution of face images and hence improve the face recognition accuracy.

Chapter 9 suggests several implementations which could be done to improve this project as well as how this project could be further improved to speed up the attendance taking process and improve the face recognition accuracy.

Chapter 10 concludes this project by giving a brief description on the main benefits of this project and describes the processes this project has taken.

Chapter 2 Literature Review

Before the actual brainstorming of how a face recognition attendance system could be implemented, identification of existing face recognition applications has been done and the methods used in face recognition have been studied.

2.1 Related Works

There are many research works being done to develop applications which make use of face recognition technology in recent years. Retailers are using face recognition technology for security purposes. For example, Walmart started testing facial recognition in its stores in order to identify shoplifters. [18]

A research group at Facebook has developed a deep learning face recognition system called "DeepFace" which is able to identify faces in digital images by employing a nine-layer neural net with more than a hundred and twenty million connection weights. [2] A face recognition system which is able to automate the tagging of user-uploaded photos has also been incorporated by Facebook. There are more than hundred million tags per day added by users to the photo collection. [18]

Facedeals is an application which is used to target customers with special offers from the business which these customers normally frequent by integrating face recognition with these customers Facebook profiles. Cameras are installed at the business entrance which are able to recognize customers when they enter using face recognition. [19]

Face ID was introduced by apple as a form of biometric authentication to enable owners to access to their phones using face recognition techniques. [2] Scene-Tap, a mobile phone-based OSN uses face recognition to get the mean age and gender ratio at bars in order to help people find the right place to drink. [18]

The MasterCard Identity Check mobile application verifies online payments when people buy goods online using face recognition. Users are able to verify their payments using the in-built cameras on their smart phones to take a photo of their faces. [19]

2.2 Methods used in face recognition

During the face recognition process, the faces are identified by extracting features or landmarks from an image of the person's face. The position, size, shape of the nose, eyes, cheekbones and mouth are being analysed. These features are then used to compare with images with matching features. [3]
Popular face recognition techniques include Principal Component Analysis (PCA), Eigenfaces, FisherFaces, Local Binary Patterns Histograms (LBPH), Linear Discriminant Analysis, Convolutional Neural Networks, Elastic Bunch Graph Matching and the Hidden Markov model.

In general, face recognition consists of the following five steps. The five steps are:

1. Pre-processing
2. Face detection
3. Facial components of region of interest
4. Feature extraction
5. Classification

2.2.1 OpenCV Gray Scale

Before a detected face captured by the web camera is added to the training set for face recognition, the image needs to be converted to grayscale first. Grayscale representations are used for extracting the face to be placed into the training set instead of coloured images because grayscale makes the algorithm simpler and reduces computational requirements. In addition, coloured images may also introduce unnecessary information which will then lead to even more training images needed to achieve good performance.

2.2.2 Principal Component Analysis and Eigenfaces

Principal Component Analysis is a statistical approach that decreases the number of variables in face recognition. The images in the training set used in Principal Component Analysis are represented by eigenfaces. [12] Eigenfaces are made up of a linear combination of weighted eigenvectors which are used in the computer vision of face recognition. [8] These eigenvectors are made up of the covariance matrix of a training image set. [12]
The main steps involved in the face recognition process using Principal Component Analysis are as follows:

1. A training face database and a dataset are prepared.
2. The average face vector is computed.
3. The average face vector is subtracted from the original faces.
4. The covariance matrix and the eigenvalues and eigenvectors of the covariance matrix are computed.
5. The K eigenvectors that match the K biggest eigenvalues are kept.
6. The weights of the features of the training images are computed.
7. The testing face image is read.
8. The feature vector of the test face is computed.
9. The Euclidean distance between the test feature vector and all the training feature vectors is computed.
10. The face class which has the lowest Euclidean distance that shows similarity to the test image is determined. [12]

Eigenfaces are created from the statistical analysis of many images of faces. These faces combine to form a human face. For example, a person's face can be created by 20% from eigenface 1, 25% from eigenface 2 and 22% from eigenface 3 and so on. Eigenfaces which are formed are created by dark and light areas which are arranged in a certain pattern. This is to enable different features of the face to be singled out and evaluated. Examples of patterns used include patterns that look for symmetry, detect facial hair, the location of the hairline, how large the eyes, nose and mouth are. An example of Eigenfaces is shown in Figure 1.

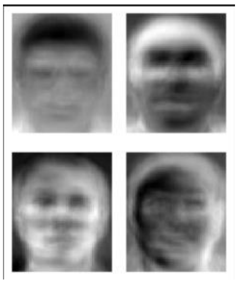


Figure 1: Figure showing an example of EigenFaces

Eigenfaces can improve the efficiency and speed of face recognition. They are able to provide an inexpensive and simple way for facial recognition and they are able to handle huge databases. Gallery images which are seen by the system are saved as collections of weights that describe how much each eigenface contributes to the image. The weights of a new face can be found by projecting the image into a collection of eigenfaces. This is so that a set of weights describing the probe face can be produced. The weights describing the probe face will be compared with all the weights in the gallery set to find the image containing the closest match. [9] The weights of each gallery image convey information which describes the image.

Every bitmap represents a data instance in a dataset of face images. Principal Component Analysis will be done on these instances. Each pixel is a number between 0 and 255 which represents the level of gray the images have. The bitmap can be unfolded into a large vector. The first row of pixels will become the first K attributes in the vector. The second row will become the second K attributes in the vector and so on. The final vector will consist of K^2 dimensions (K rows of K attributes each.) This process is done for every instance in the dataset to obtain a matrix which has K^2 rows and N columns which represents the individual instances as shown in Figure 2. [7]

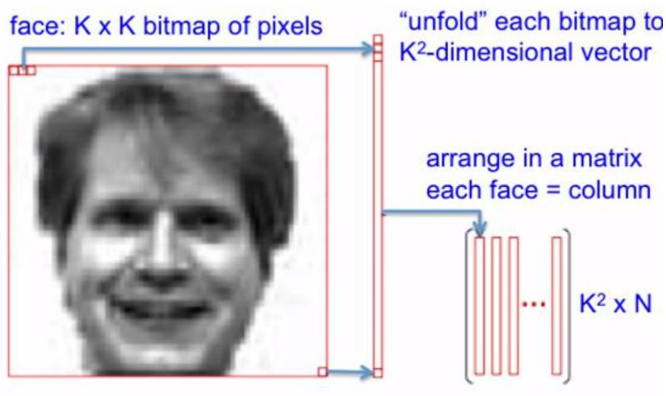


Figure 2

Principal Component Analysis will be performed on these instances. At the end of Principal Component Analysis, a certain number **m** of these new dimensions of these eigenvectors with the biggest eigen values (eigen vectors which span the most variance in this matrix.) are chosen. n columns will be obtained at the end where each column is K^2 dimensional. Each column corresponds to the original positions of pixels in the bitmap. Each of the eigen vectors has K square dimensions as many as pixels. The eigenvectors are then folded back into a bitmap. The first K attributes are transformed into the first row of pixels, the second K attributes are transformed into the second row of pixels and so on. In the end, something like the image in Figure 3 will be created. [7]

Figure 3

The image in Figure 3 does not look like the average face of any kind. It shows the most prominent deviations from the mean in the dataset where the mean is the prototypical average looking face. This is produced by averaging the faces of different people. These are the dimensions people seem to vary the most from the mean. The highest variation usually tends to be at the hair since everyone has a nose, mouth and eyes but has more variations with the hair. If we repeat this process, we get the eigenvectors. The eigenfaces in Figure 4 captures the variations in the hair, the variations in the eyes and variations in the position of the face. [7]

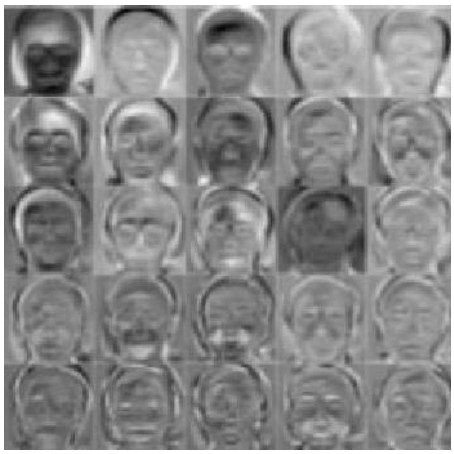


Figure 4

An image of a face is made up of whatever the mean face is and some amount of the first eigenface, some amount of the second eigenface and so on as shown in Figure 5. [7]

Figure 5

2.3 Face Detection

Face detection of an image needs to be performed first before face recognition. The camera needs to detect a face first before the training image of the face can be added into the training set. The method used for face detection is the Viola-Jones object detection framework.

The Viola-Jones algorithm consists of four stages.

The four stages are:

1. Haar Feature Selection
2. Creating an Integral Image
3. Adaboost training
4. Cascading classifiers [16]

In the first stage of the Viola-Jones algorithm, Haar feature based Cascade classification is used. Haar feature-based cascade classifiers are implemented whereby a cascade function is trained from many images of faces (positive images) and images without faces (negative images). Features then need to be extracted from the image.

All human faces contain similar properties. Haar features may be used to match these regularities.

Some of the properties that all human faces share include the following:

1. The upper cheeks are brighter than the region where the eyes are located.
2. The eye region is darker than the nose bridge region.

The other two Haar features are the location and size of the bridge of nose, mouth and eyes as well as the oriented gradients of pixel intensities.

These four features matched by the Viola-Jones algorithm are sought in the image of a face as shown in Figures 6 and 7 below. [16]

Figure 6: Haar Feature where the upper cheeks are brighter than the eye region

Figure 7: Haar feature which looks similar to the nose bridge region that is brighter than the eye region is applied to the face

Each Haar feature is computed by the formula:

Sum of pixels in the black rectangle – sum of pixels in the white rectangle.

There are three types of Haar features as shown in Figure 8. These Haar features are classified into two-rectangle, three-rectangle and four-rectangle features. Viola-Jones uses the two-rectangle feature. It is used to find the difference in brightness between the black and white rectangles over a certain area.

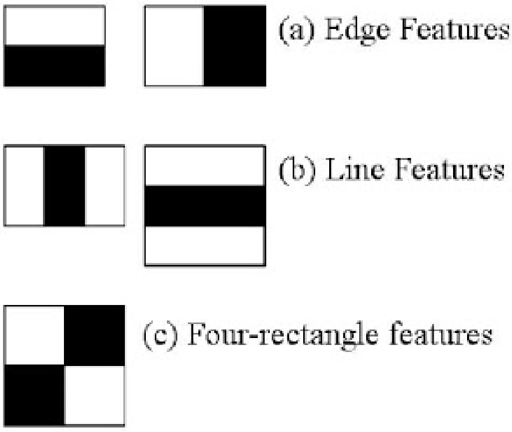


Figure 8

The integral image is then used to compute the sum of pixels under the black and white rectangles for each feature.

The best threshold is obtained for each feature that is able to classify the faces into positive ones and negative ones. The features which are able to classify the face and non-face images accurately are selected.

It may take a long time for the features to be evaluated. Hence, a variant of the learning algorithm, Adaboost is used to select the best features and train classifiers which use them. [16]

Weak classifiers refer to classifiers which are not able to classify the image alone but are able to form a strong classifier with others. [13] A "strong" classifier is constructed from the Adaboost algorithm as a linear combination of weighted "weak" classifiers. [16] The final classifier is computed from the total sum of these weak classifiers. [13]

In many images, most of the image are made up of non-face regions. Hence, an easier way can be used to check whether the window is a face region. If the window does not contain a face region, it should be discarded and not processed anymore. More time can be spent checking on regions where a face may be present. The concept of cascade of classifiers can be used. This concept involves grouping the features into different stages of classifiers and apply them one by one instead of applying thousands of features on a window. If the window passes the stage, we apply the second stage of features, otherwise the window will be discarded. This process is continued until the window passes all the stages, which indicates that a face region is detected. [13]

Chapter 3 Methods to obtain the training face images of the students

In order for the face recognition to be performed, there must be images of students' faces placed in the training set first. The training images of the students can be obtained by the following methods:

1. Obtain the training images from the students Facebook accounts.

Training images of students can be obtained from their photos in their Facebook accounts.
The face images from these photos are then cropped so that they can be placed into the training set as shown in Figure 9.

Figure 9: Figure showing face images from photos being cropped

2. The training images can also be obtained from the photos on the students' matriculation cards.

3. The training images can also be obtained from the image frames of students from CCTV cameras within the school compounds when the students are in school. Administrators can use the students' faces extracted from these CCTV cameras and place them into the training set. An example is shown in Figure 10.

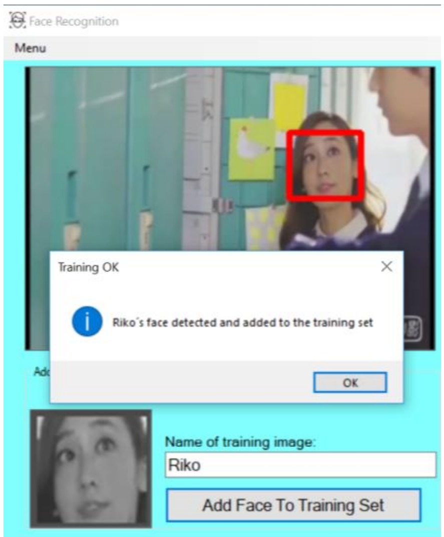


Figure 10: Example of student "Riko" face being added into the training set

4. Students can use the application to add more images of their own faces with different expressions, pose variations, with or without glasses and positions of their faces into the training set. This can be done through the following steps as shown in Figure 11.

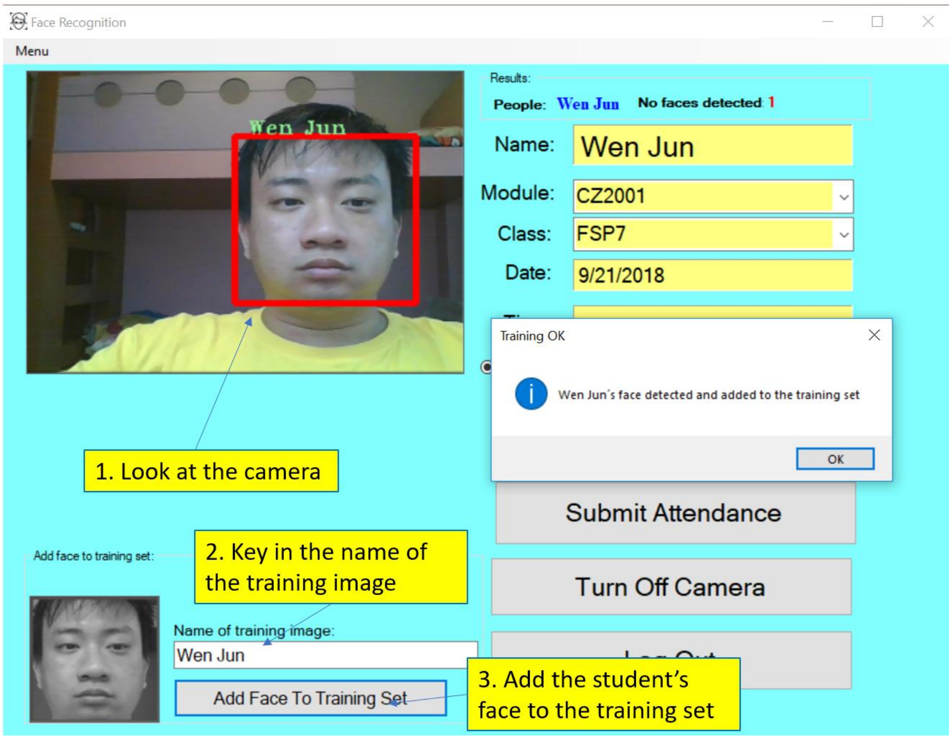


Figure 11: Figure showing how students can add training images of themselves into the training set.

These training face images will then be cropped and converted to grayscale to remove the unnecessary information which coloured images might introduce.

The training face images will then be stored in a folder (also called a training set) with names (also called labels) being assigned to every training image. This training set is found in the …\...\bin\Debug\TrainedFaces folder. When these images are stored in the folder, they are being numbered automatically. The training set will contain many grayscale images of students as shown in Figure 12 below. These are then used for comparison purposes during the face recognition phase.

Figure 12: Figure showing the training set containing training images

There will be names assigned to each training face image in the training set. These names are stored in TrainedLabels.txt as shown in Figure 13. In Figure 13, the training image names (or labels) are separated by "%". The first name will correspond to the first training image in the training set with the image name as "face1.bmp", the second name will correspond to the second training image in the training set with the image name as "face2.bmp" and so on. The first two digits indicate the number of training images in the training set.

Figure 13: Figure showing the trained labels given to each image

Chapter 4 How the application works

4.1 How students can use the application

When student(s) look at the camera to take their attendance, the process of the face recognition system is split up into many steps.

The main steps are:

1. A high definition camera device will be installed at the entrance of the classroom.
2. Students will need to log into the system to get their attendance taken by typing in their user names and passwords and selecting the user type as “Student” as shown in Figure 14.

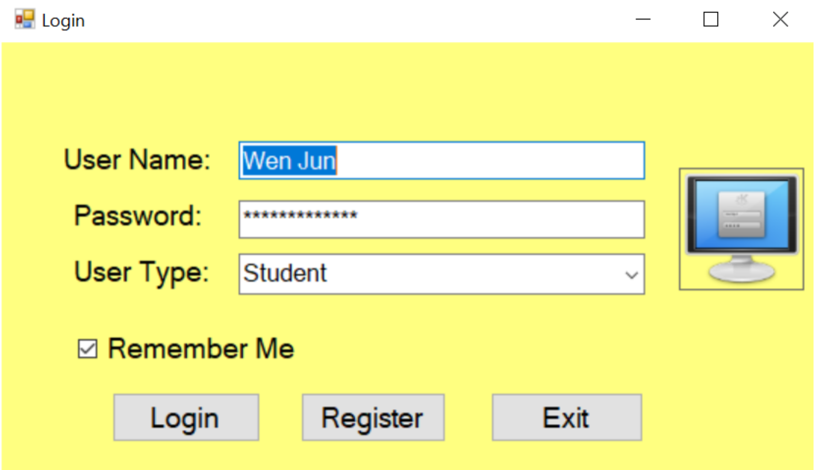


Figure 14: Figure showing a student logging into the system

3. Students can either choose to take their attendance individually or as a group of up to five students by looking at the camera for their attendance to be taken.
   The following diagram in Figure 15 below shows an example of a student taking his attendance individually.

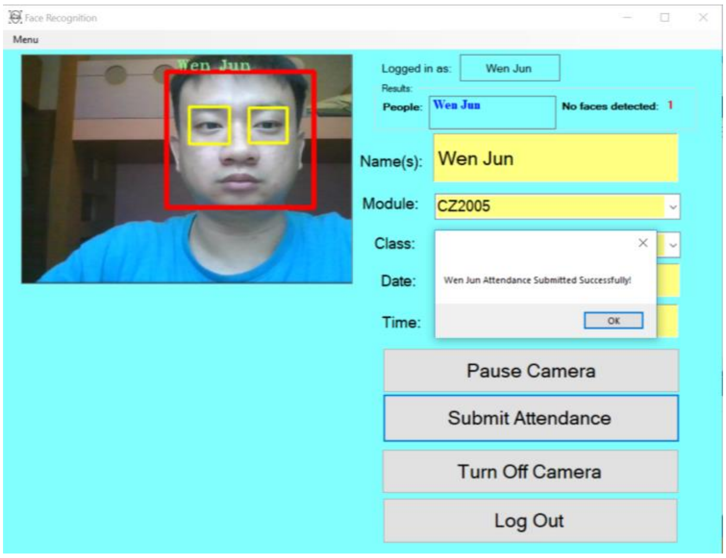


Figure 15: Figure showing a student taking his attendance individually

Students can also choose to take their attendance as a group of up to five people. The diagram in Figure 16 shows how five people can take their attendance at the same time.

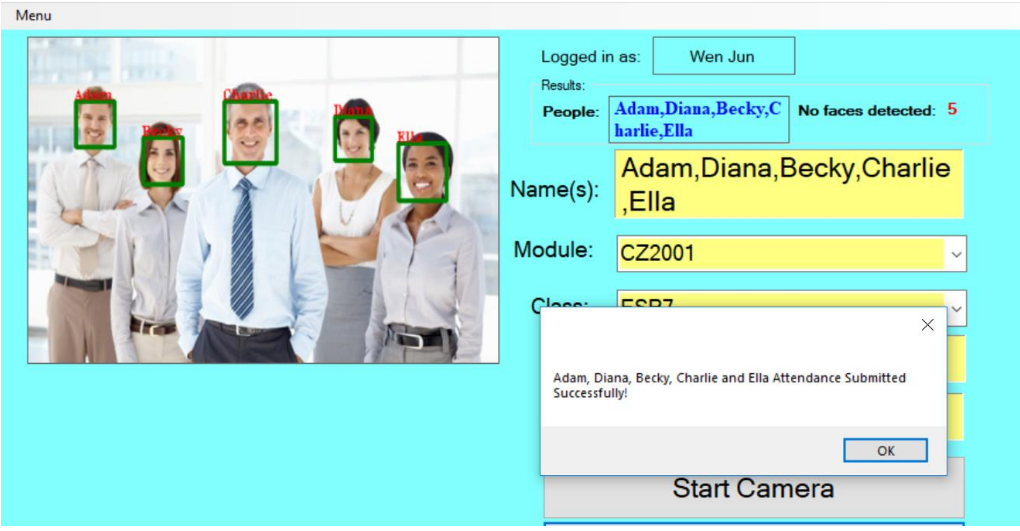
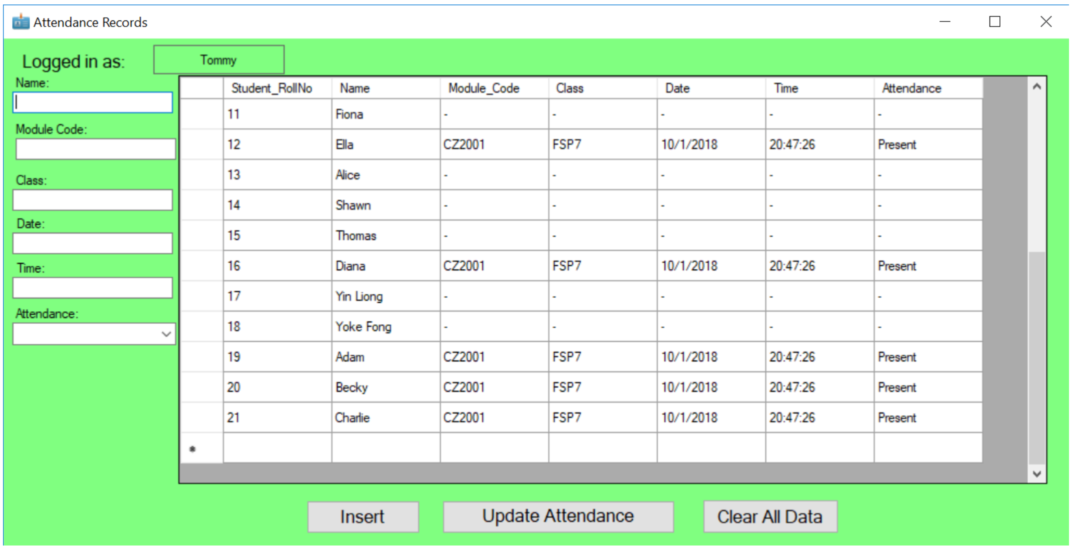


Figure 16: Figure showing how a group of students can take their attendance at a time

4. The students' faces from the live image taken from the camera are being detected using the Viola Jones detection framework if there are any.
5. The detected faces are then being extracted.
6. Face recognition is then performed by comparing the detected faces against the face images in the training set and show the match found.
7. When a match is found, a rectangle will be drawn around the students' faces with the students' names outside the rectangles to indicate that the students' faces have been detected. Eigenfaces are then used to verify these faces one by one.
8. If the students' attendance has been submitted successfully, a voice conversion system will also read out the message "Attendance Submitted Successfully". The students' attendance will then be recorded and stored in the database as shown in Figure 17. However, only teachers are allowed to view the attendance records of the students.



Student_RollNo	Name	Module_Code	Class	Date	Time	Attendance
11	Fiona	-	-	-	-	-
12	Ella	CZ2001	FSP7	10/1/2018	20:47:26	Present
13	Alice	-	-	-	-	-
14	Shawn	-	-	-	-	-
15	Thomas	-	-	-	-	-
16	Diana	CZ2001	FSP7	10/1/2018	20:47:26	Present
17	Yin Liong	-	-	-	-	-
18	Yoke Fong	-	-	-	-	-
19	Adam	CZ2001	FSP7	10/1/2018	20:47:26	Present
20	Becky	CZ2001	FSP7	10/1/2018	20:47:26	Present
21	Charlie	CZ2001	FSP7	10/1/2018	20:47:26	Present

Figure 17:Figure showing students' attendance records stored in the database

4.1.1 Circumstances when the system is not able to capture the student’s attendance

There are circumstances when the face recognition attendance monitoring system is not able to capture the students’ attendance successfully. These circumstances may arise due to the following reasons:

1. If the student does not belong to the class, the face recognition system may not be able to recognize the students’ faces if it is not able to find a match after comparing the students’ faces with all the training images in the training set. In such cases, the face will be classified as “UNKNOWN” (i.e. not recognized as a student from the class) and the “Submit Attendance” button will be disabled as shown in Figure 18.

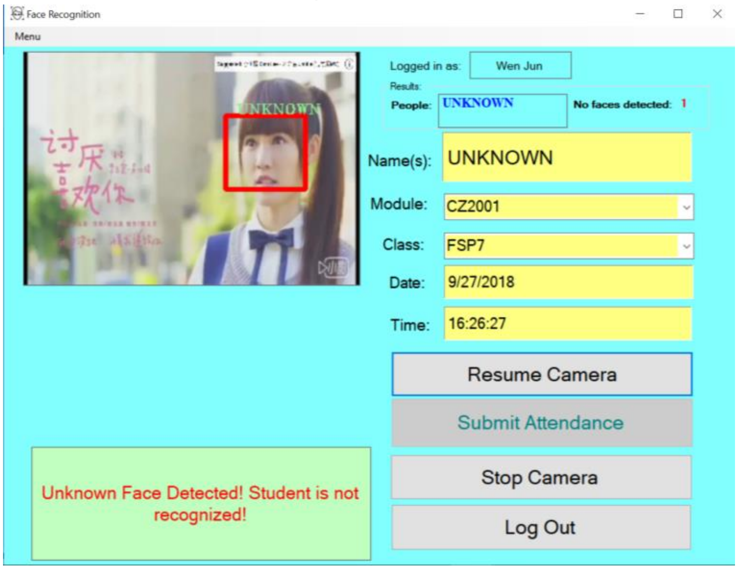


Figure 18: Figure showing a student's face being detected as unknown

2. If there are no faces detected by the camera and the students click on the “Submit attendance” button, a voice conversion system will read out the message “No faces detected, please try again.

4.1.2 Other Settings

Students can also choose to get their eyes detected in addition to recognizing their faces by selecting Other Settings > Choose what to detect > Detect face and eyes. The eyes will only be detected if they are open as shown in Figure 19 below.

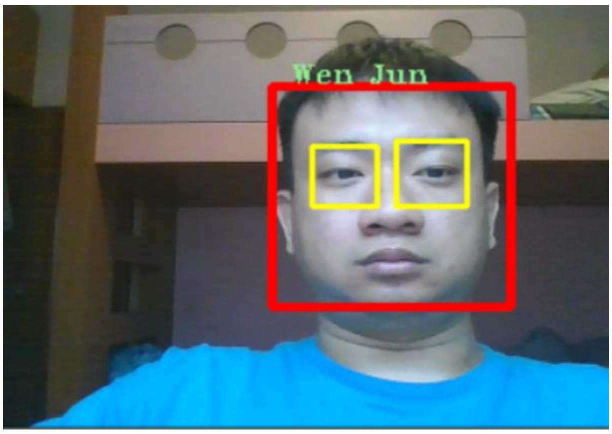


Figure 19: Figure showing both the student's faces and eyes being detected

Students can also adjust the settings such that the image frames captured by the camera are in grayscale as shown in Figure 20 to improve the face recognition accuracy as coloured images may also introduce unnecessary information to the image.

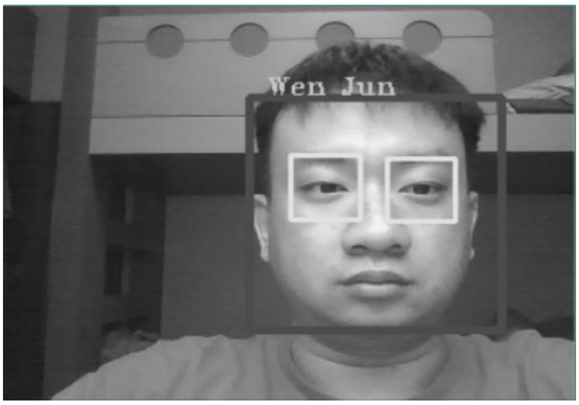


Figure 20: Face being detected in grayscale

4.2 How teachers can use the application

Teachers can view the attendance records of students which are captured by the face recognition system. They can also modify the attendance records of the students if the face recognition system did not take down the attendance of the students correctly as shown in Figure 21.

To log into the application, teachers will need to type in their user name, password and select the user type as "Teacher".

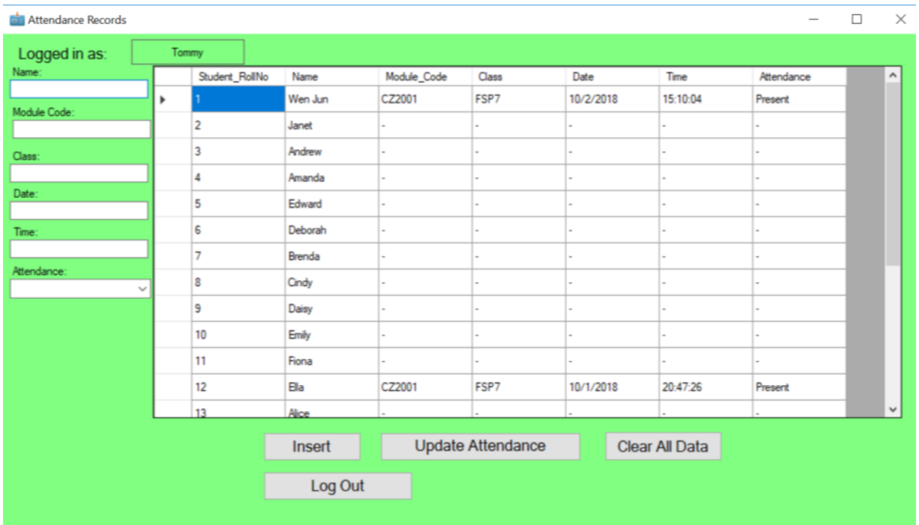


Figure 21: Figure showing the page where teachers can view, update or modify the attendance records of the students

For example, if student "Wen Jun" is registered to be in the wrong class in the database, the teacher can change the class name in the database by amending the various entries of the student's information from the data grid and then click the update button to update the attendance records as shown below in Figure 22.

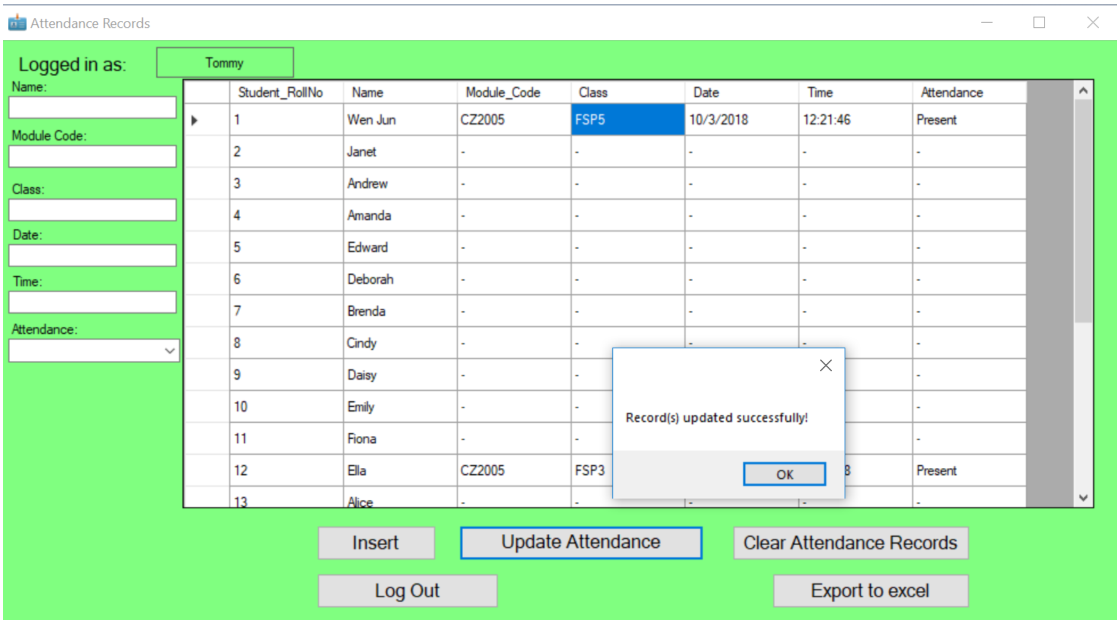


Figure 22:Figure showing Wen Jun's class being amended from "FSP3" to "FSP5"

As mentioned in the previous page, teachers can also manually insert the students' attendance if the application fails to recognize the students' faces correctly. They can do so by typing' the various fields into the textboxes on the left of the form and click the insert button as shown in Figure 23.

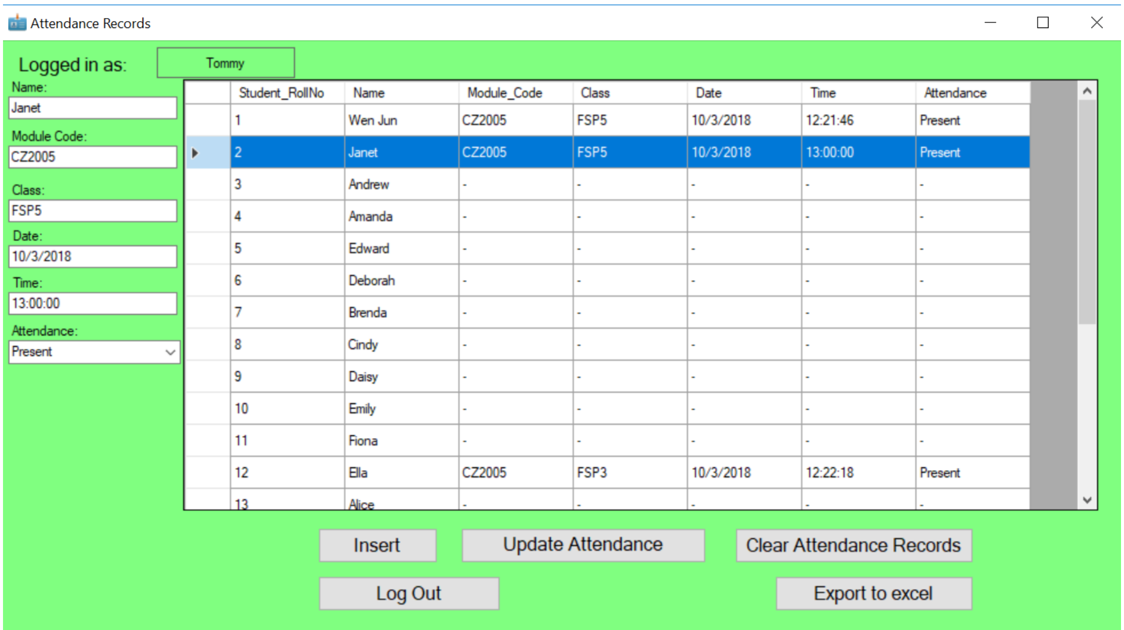


Figure 23: Figure showing the attendance records of Janet being inserted successfully

Teachers can also export the students' attendance records to an excel sheet automatically after they have verified the students' attendance records for the day by clicking on the "Export to excel" button. The students' attendance records will be saved in an excel sheet as shown in Figure 24.

	A	B	C	D	E	F	G
1	Student_RollNo	Name	Module_Code	Class	Date	Time	Attendance
2	1	Wen Jun	CZ2005	FSP3	10/4/2018	21:19:44	Present
3	2	Janet	CZ2005	FSP5	10/3/2018	13:00:00	Present
4	3	Andrew	-	-	-	-	-
5	4	Amanda	-	-	-	-	-
6	5	Edward	-	-	-	-	-
7	6	Deborah	-	-	-	-	-
8	7	Brenda	-	-	-	-	-
9	8	Cindy	-	-	-	-	-
10	9	Daisy	-	-	-	-	-
11	10	Emily	-	-	-	-	-
12	11	Fiona	-	-	-	-	-
13	12	Ella	CZ2005	FSP3	10/3/2018	12:22:18	Present
14	13	Alice	-	-	-	-	-
15	14	Shawn	-	-	-	-	-
16	15	Thomas	-	-	-	-	-
17	16	Diana	CZ2005	FSP3	10/3/2018	12:22:18	Present
18	17	Yin Liong	-	-	-	-	-
19	18	Yoke Fong	-	-	-	-	-
20	19	Adam	CZ2005	FSP3	10/3/2018	12:22:18	Present
21	20	Becky	CZ2005	FSP3	10/3/2018	12:22:18	Present
22	21	Charlie	CZ2005	FSP3	10/3/2018	12:22:18	Present

Figure 24: Figure showing students attendance records saved in an excel sheet

4.3 How administrators can use the application

Administrators can use the application for two purposes:

1. To obtain more training images of students from various sources such as the students' Facebook accounts, matriculation cards, class photos (if any) and CCTV cameras to be placed into the training set.
2. To conduct face recognition experiments on students' faces to determine how various factors such as the number, size and type of training images affect the face recognition accuracy and the face recognition speed.

For administrators to log into the system to obtain more training images of students, they will need to key in their user name followed by their passwords and then select the user type as "Admin" and then select "Obtain Training Images of Students from a picture or video option".

4.3.1 Obtain training images of students from pictures

Administrators can obtain training images of students from photos taken from the students Facebook accounts, matriculation cards or class photos (if any). These images are then added into the training set as shown in Figure 25.

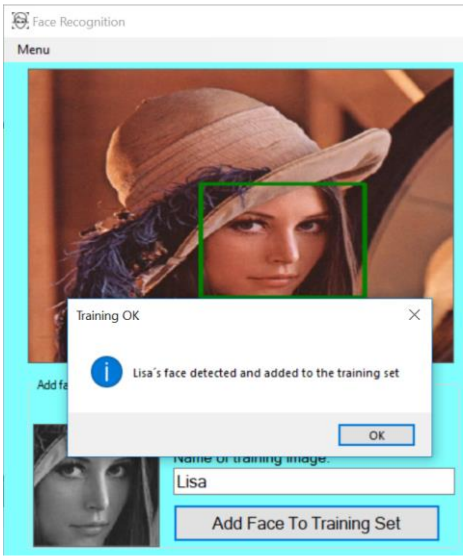


Figure 25: Figure showing a student's face being added into the training set from a photo

4.3.2 Obtain training images of students from videos

Administrators can also obtain the training images of students from the image frames taken from the CCTV cameras when the students are in the school compounds. These images are then added into the training set as shown in Figure 26.

Figure 26: Figure showing a student's face added to the training set from an image frame captured by a CCTV camera

4.3.3 Conduct face recognition experiments

Administrators can also use the application by conducting face recognition experiments to determine how various settings such as the number, size, type of training images used and the various face recognition and face detection settings will affect the face recognition accuracy and face recognition speed. In this way, administrators will then be able to find out more ways to improve the system's face recognition accuracy and speed.

In order for administrators to conduct face recognition experiments, they will need to log into the system by typing in their user name and password, select the user type as "Admin" and then click the "Log in" button and then select the "Conduct Face Recognition Experiments" option.

The administrator will be directed to the following page where he or she can conduct face recognition experiments as shown in Figure 27 below.

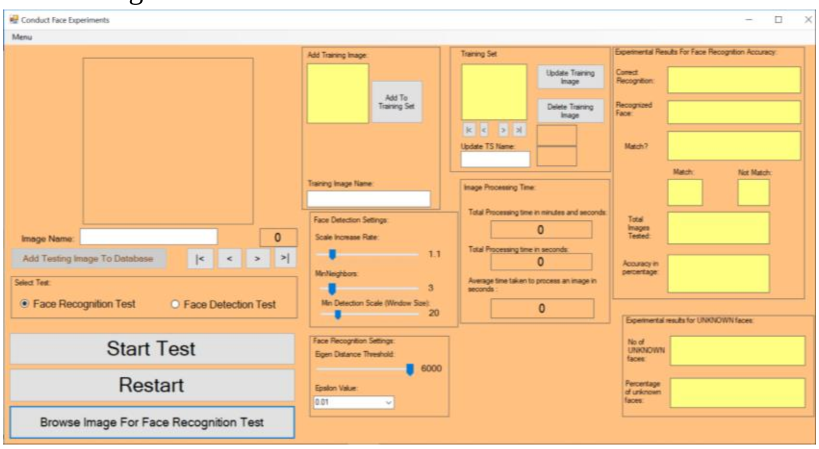


Figure 27: Figure showing the page which the administrator can conduct face experiments

The training images can be added to the database by the administrator as follows after clicking the "Browse Image for Face Recognition Test" button followed by typing in the name of the training image and then click on the "Add To Training Set" button as shown in Figure 28.

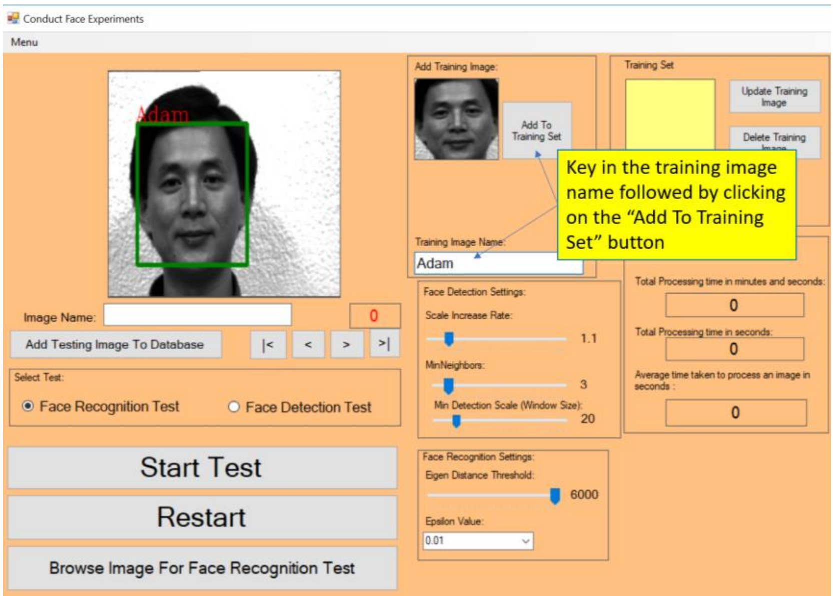


Figure 28:Figure showing the administrator key in the training image name followed by clicking on "Add Image to training set" button

Each testing image is given a name which corresponds to the face that needs to be **correctly recognized** as shown in Figure 29. The testing images can similarly be added to the database by clicking on the "Browse Image For Face Recognition Test" button and then give a name to the testing image and then click on the "Add Testing Image To Database" button.

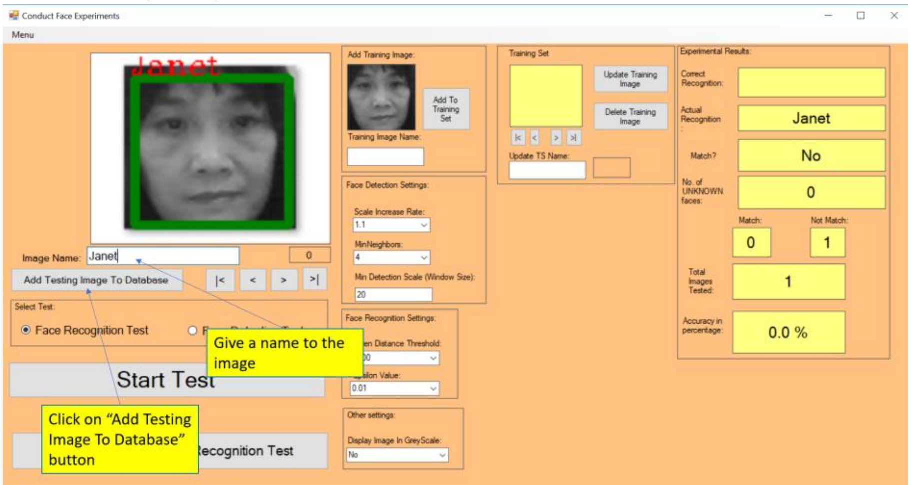


Figure 29: Figure showing how administrators can assign a name to the testing image and then add it to the testing image database

How the face recognition experiments will be conducted will be explained further in **chapter 6** on the **Experiment Set Up** section.

Chapter 5 Overview of the System

5.1 Architecture diagram of the system

The three-tier architecture diagram is used for the system. The three-tier architecture diagram is classified into three different tiers, namely the presentation tier, the logic tier and the data tier. The three-tier architecture allows any of the tiers to be replaced or upgraded independently due to changes in requirements.

The **presentation tier** is the tier which contains components which users can access directly. The presentation tier usually stores the user interface.

The **logic tier** is the tier which is used to control an application's functionality by performing detailed processing.

The **data tier** is the tier where information is stored and retrieved from a database or file system. Database servers and file shares are usually found in the data tier.

The diagram in Figure 30 shows the architecture diagram of the attendance monitoring system and its various components.

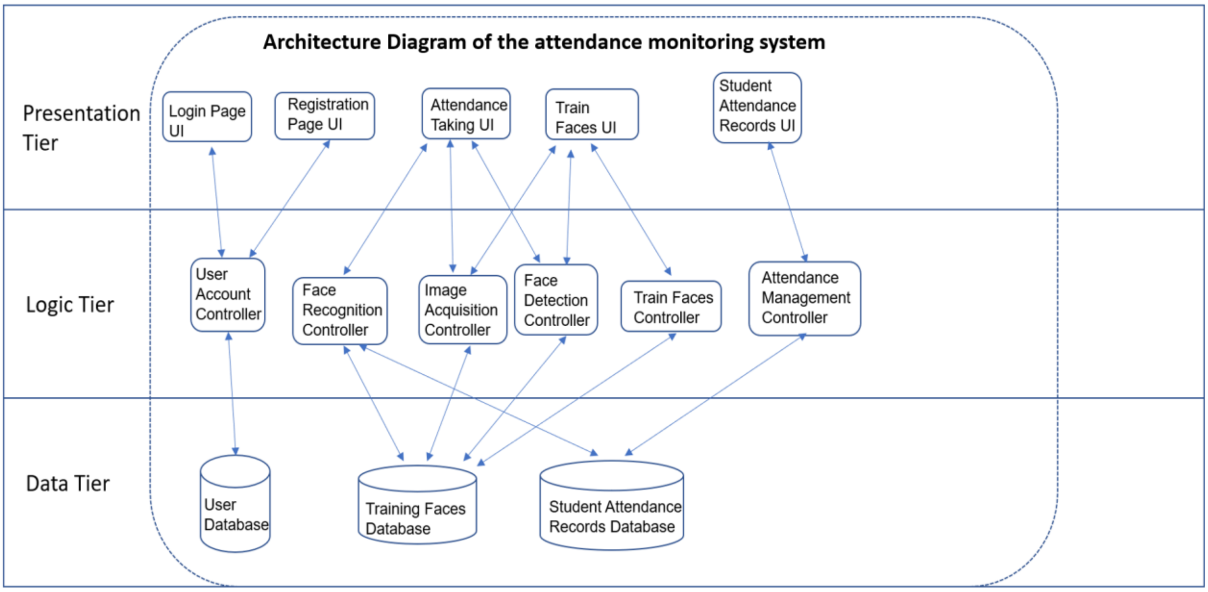


Figure 30:Architecture Diagram of the Attendance Monitoring System

5.2 Diagram showing the relationships between the individual modules

The following diagram shown in Figure 31 below gives a summary of the relationships between the various modules for the face recognition to be done.

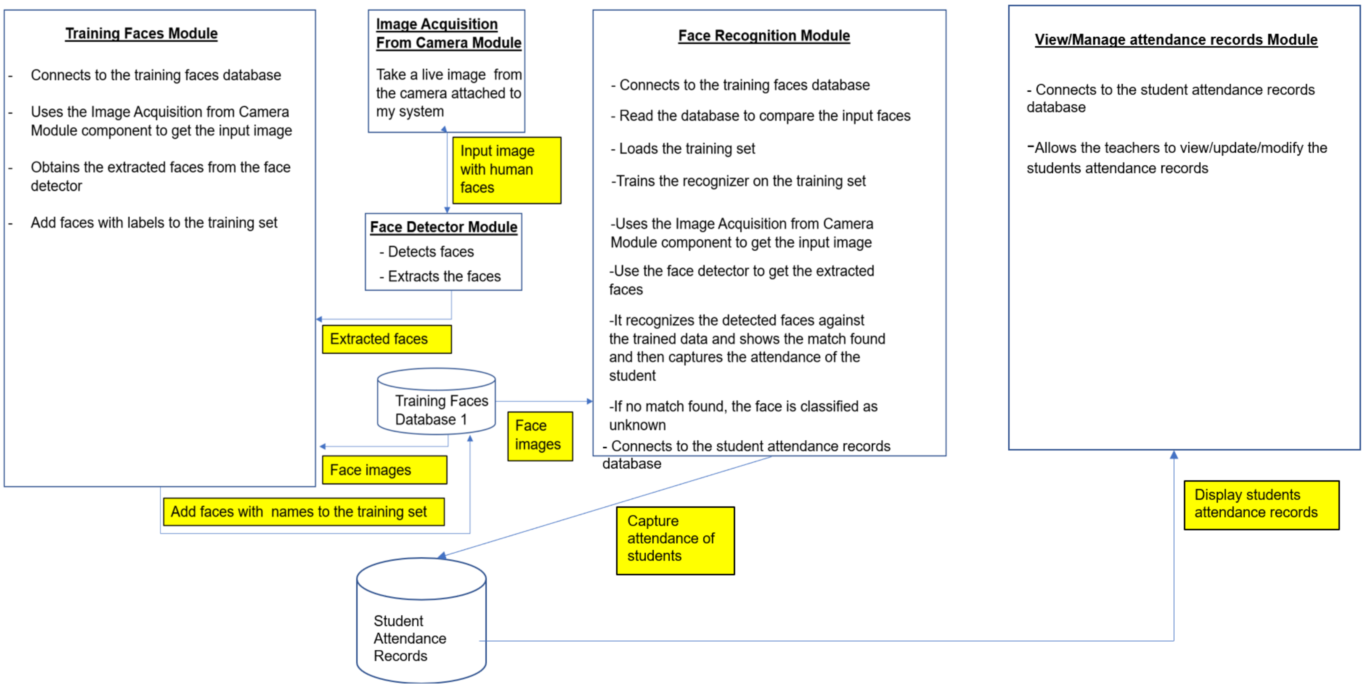


Figure 31:Diagram illustrating how face recognition is done

5.3 Libraries used

The libraries which are used for my project are OpenCV and EmguCV.

OpenCV (Open Source Computer Vision) refers to a library of programming functions aimed at real time computer vision. OpenCV is used extensively in face recognition systems.

EmguCV is a cross platform .Net Wrapper to the OpenCV image processing library. It allows OpenCV functions to be called from .NET compatible languages such as C#, Visual Basic. EmguCV together with C# .Net, are libraries which are used to capture and process image of the capture device such as cameras in real time. [9]

Chapter 6 Experiment Set Up

The objectives of these experiments were to determine the factors that affect the face recognition accuracy as well as the face recognition speed. This can eventually help to improve the accuracy and speed of the face recognition system by tuning the various settings to recognize students' faces with a higher success rate.

The experiments were conducted on 20 persons where 7 training images were used for each person and 4 testing images were used for each person. Hence, a total of 140 training images and 80 testing images were used for the experiments. Out of the 20 persons, the training and testing images of 15 persons were obtained from the Yale Face database which contains images of persons with different expressions, orientations, positions of the face and different lighting conditions. The training and testing images of the remaining 5 persons were obtained from the test subjects. The images from the test subjects were obtained by asking the test subjects to sit in front of the camera to get their training and testing images taken with different expressions, positions and orientations of the face. This helped to improve the face recognition accuracy rate as face recognition methods that make use of Principal Component Analysis perform more than one comparison and matches between the images stored in the training set and the faces detected. The faces of the test subjects were being detected by the Viola-Jones detection framework.

7 of these detected face images for each person were then cropped, converted to greyscale and stored into the training images folder. Every training image was given a name which corresponds to the faces in the training images.

The remaining 4 of these detected face images for each person were the testing images which were stored in the testing images database. Every testing image will be given a name which corresponds to the correct name of the face in the testing image which needs to be recognized. For example, if the testing image contains a face of John, the name of this testing image should also be John.

The 80 face images in the testing images database were then compared one by one with all the 140 other face images in the training images database using Principal Component Analysis with Eigenfaces.

The name of the recognized face will be placed outside the rectangle of the detected face and will correspond to the recognized face label. The name of the testing image will also be the name of the face which needs to be recognized which will correspond to the correct recognition label as shown in Figure 32.

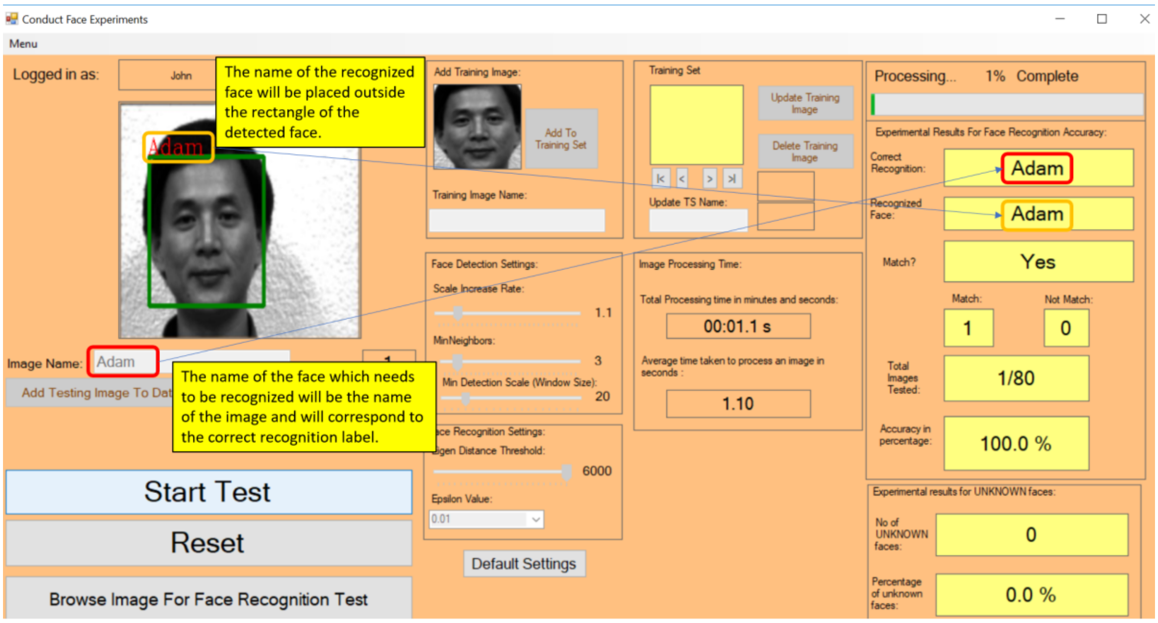


Figure 32

The testing images are considered to be recognized correctly when the name of the recognized face of the testing image is the same as the name of the testing image. The number of testing images being recognized correctly will then be increased by 1 as shown in Figure 33.

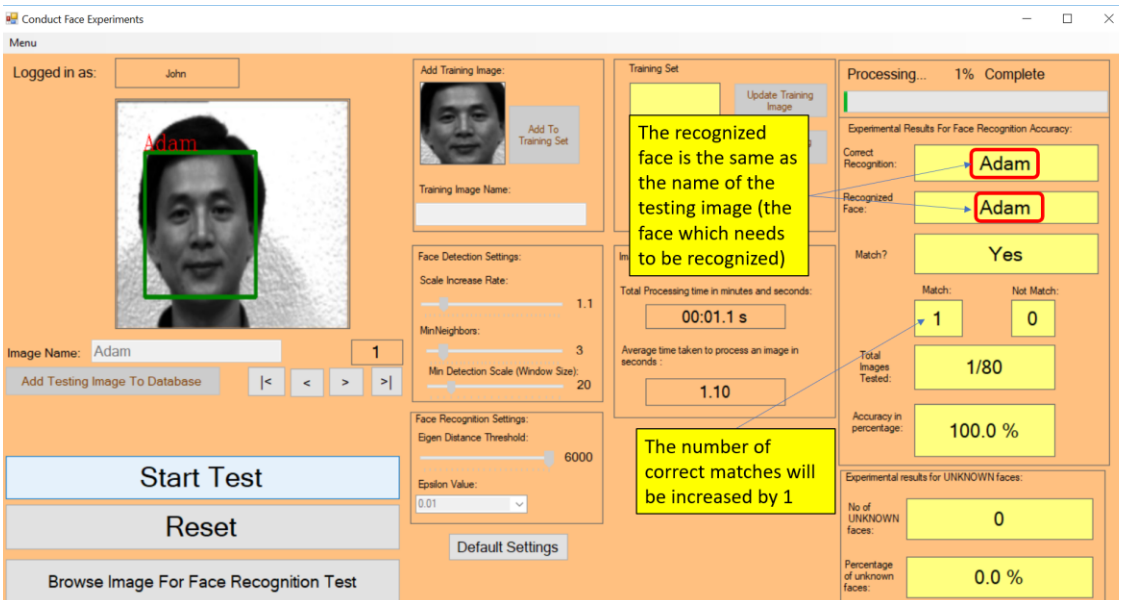


Figure 33: Figure showing the situation where a face in the testing image is being recognized correctly

However, the testing images are considered to be recognized wrongly under two scenarios:

1. The name of the recognized face is different from the name of the testing image as shown in Figure 34.

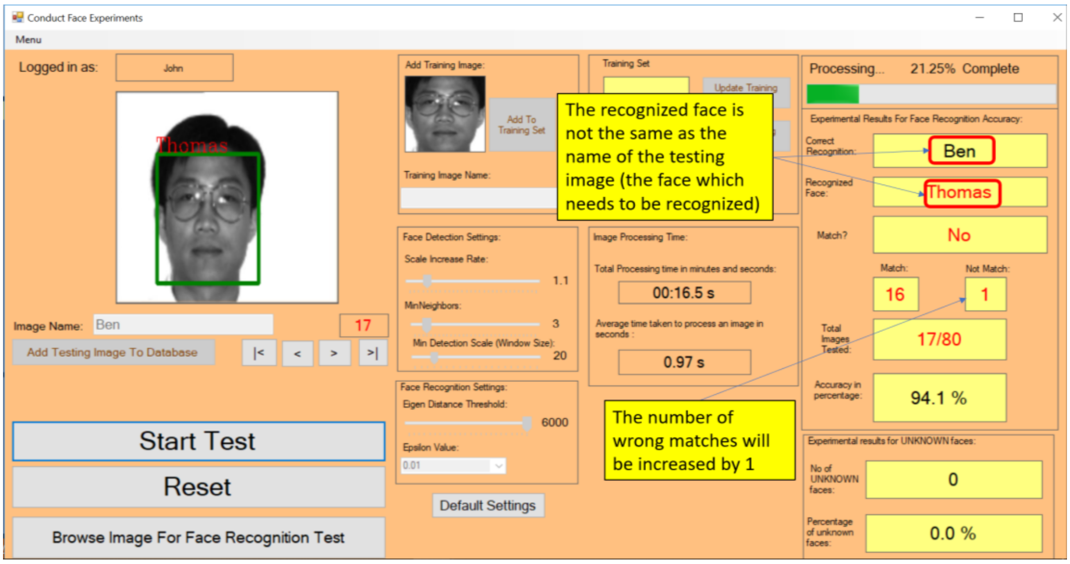


Figure 34: Figure showing the name of the recognized face is different from the name of the testing image

2. The name of the recognized face is labelled as "UNKNOWN" as shown in Figure 35. (This means that the face of the testing image is not recognized when it is compared with all the images in the training set to search for a match.)

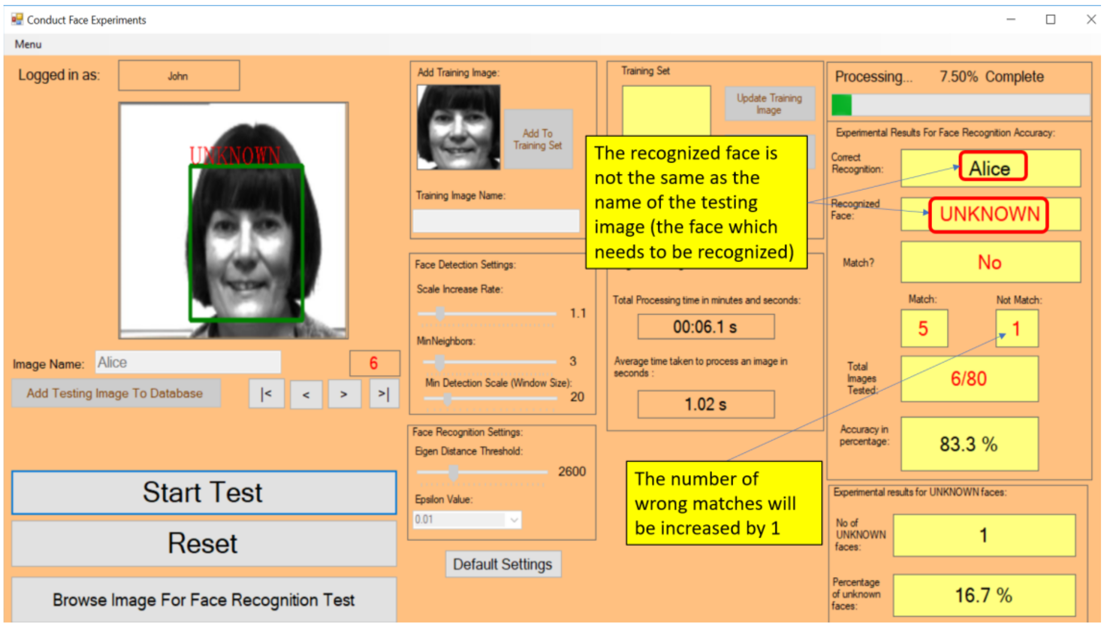


Figure 35: Figure showing a face in the testing image being recognized as "UNKNOWN"

After the test is run on all the 80 face testing images, the number of faces being recognized correctly, the number of faces being recognized wrongly and the face recognition accuracy (in percentage) are determined. The total time taken to process all the 80 face testing images (in seconds) as well as the average time taken to process an image (in seconds) will also be determined. The percentage of faces being detected as "UNKNOWN" (not recognized) will also be determined.

An example is shown in Figure 36 where the Eigen distance threshold is set to 4000 and the epsilon value is set to 0.001 after the test is run on all the 80 testing images.

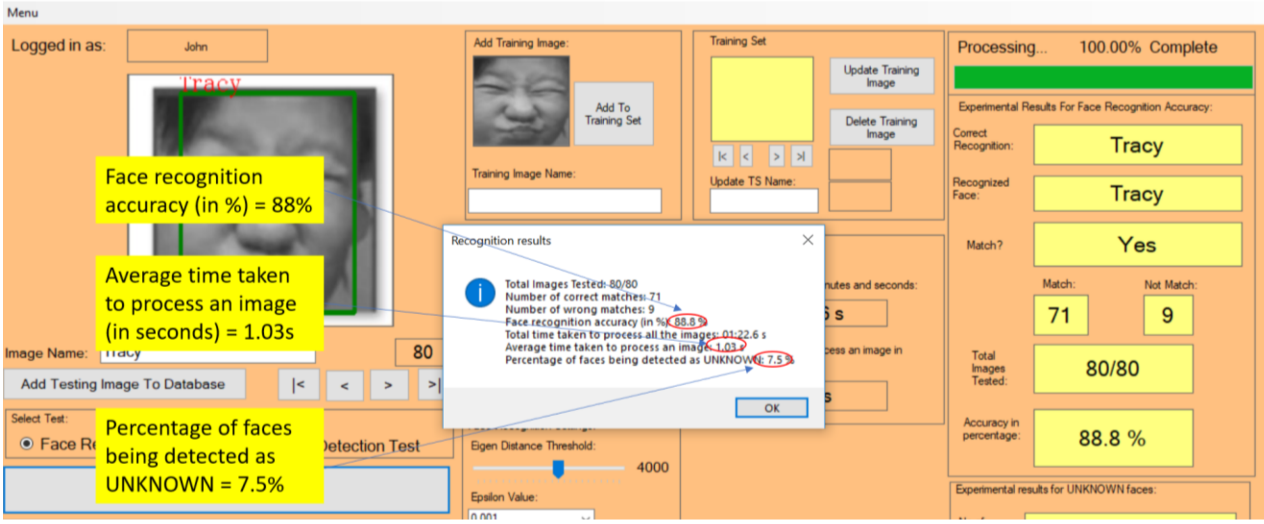


Figure 36: Figure showing an example after the experiment is run on all the 80 testing images

Viola-Jones detection algorithms were used in the face detection process and Eigenfaces face recognition algorithms were used in the face recognition process. The face recognition accuracy in percentage were determined by the formula:

$$\text{Face recognition accuracy (in \%)} = \frac{Total\ number\ of\ faces\ being\ recognized\ correctly}{Total\ number\ of\ testing\ images} \text{ x } 100\%$$

The face recognition speed (measured by the time taken to process an image) were determined by the formula:

$$\text{Average time taken to process an image (in seconds)} = \frac{Total\ time\ taken\ to\ process\ all\ the\ testing\ images(in\ seconds)}{Total\ number\ of\ testing\ images}$$

The percentage of unknown faces detected were determined by the formula:

$$\text{Percentage of unknown faces detected} = \frac{Number\ of\ unknown\ faces\ detected}{Total\ number\ of\ testing\ images} \text{ x } 100\%$$

A total of 7 experiments were conducted. These experiments covered how the various factors affect the face recognition accuracy, face recognition speed and the percentage of faces detected as “UNKNOWN” in the system. The experiments which were conducted to determine the relationship between the various factors and face recognition accuracy were as follows:

1. How the eigen distance threshold values affect the face recognition accuracy.
2. How the epsilon values affect the face recognition accuracy.
3. How the number of training images affect the face recognition accuracy.

The experiments which were conducted to study the relationship between the various factors and the face recognition speed were as follows:

1. How the eigen distance threshold values affect the face recognition speed.
2. How the epsilon values affect the face recognition speed.
3. How the number of training images affect the face recognition speed.

In addition, there was one experiment which was conducted to study the relationship between the eigen distance threshold values and the number of faces detected as “UNKNOWN”. (Not recognized).

Chapter 7 Experiment Results

7.1 Experiment Results for Experiment 1

This experiment was conducted to determine how the eigen distance threshold values affects the face recognition accuracy with the **epsilon value set** to a **constant value of 0.01.**

A total of 3 data sets were being used for the experiment. The testing images and training images were swapped randomly in all the 3 data sets for cross-validation to be done. The average face recognition accuracy from the results of the 3 data sets were then determined.

The experiment results in the 3 data sets for **experiment 1** which shows the relationship between the eigen distance threshold values and the face recognition accuracy are shown in Table 1 as follows:

Eigen distance threshold	Accuracy Rate of Data Set 1 (in %)	Accuracy Rate of Data Set 2 (in %)	Accuracy Rate of Data Set 3 (in %)
6000	92.5	82.5	93.8
5800	92.5	82.5	93.8
5600	92.5	82.5	93.8
5400	92.5	82.5	93.8
5200	92.5	82.5	93.8
5000	92.5	82.5	93.8
4800	92.5	82.5	93.8
4600	92.5	82.5	93.8
4400	91.3	82.5	92.5
4200	90	82.5	92.5
4000	90	81.3	92.5
3800	88.8	80	92.5
3600	86.3	80	92.5
3400	82.5	78.8	91.3
3200	80	75	90
3000	78.8	72.5	88.8
2800	72.5	71.3	85
2600	65	65	80
2400	52.5	60	68.8
2200	46.3	53.8	65
2000	36.3	46.3	55
1800	26.3	31.3	46.3
1600	17.5	17.5	37.5
1400	11.3	7.5	30

Table 1: Table showing how the eigen distance threshold values affects the face recognition accuracy results in the three data sets

The average face recognition accuracy is then determined by taking the average face recognition accuracy from the results of the three data sets and plot them as a graph as shown in Figure 37.

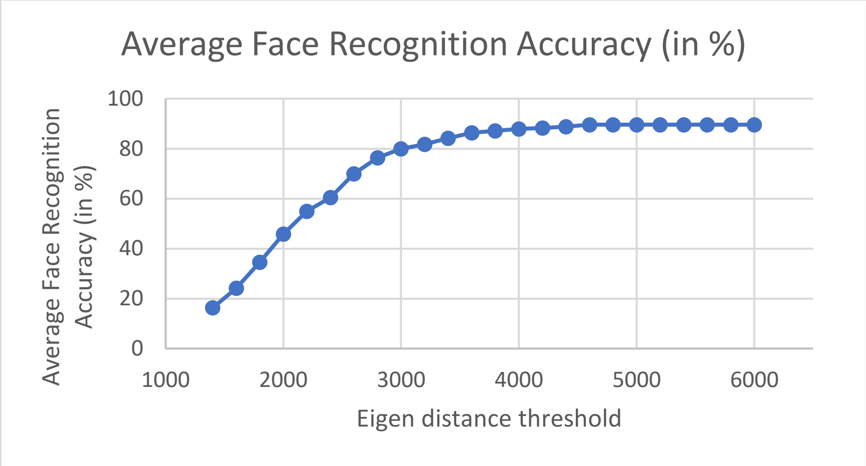


Figure 37: Figure showing the relationship between the Eigen distance threshold values and the average face recognition accuracy

7.2 Experiment Results for Experiment 2

This experiment was conducted to determine how the epsilon values affect the face recognition accuracy of the system with the eigen distance threshold value set at a constant value **of 5000.**

A total of 3 data sets were being used for the experiments. The testing images and training images were swapped randomly in all the 3 data sets for cross-validation to be done. The average face recognition accuracy from the results of the 3 data sets were then determined.

After the experiment was conducted on the 3 data sets, the Face Recognition Accuracy of the 3 data sets were determined and represented in Table 2.

Epsilon Value	Face Recognition Accuracy of Data Set 1 (in %)	Face Recognition Accuracy of Data Set 2 (in %)	Face Recognition Accuracy of Data Set 3 (in %)
0.001	91.3	82.5	96.3
0.002	91.3	82.5	96.3
0.004	92.5	82.5	95
0.006	92.5	82.5	96.3
0.008	92.5	83.8	96.3
0.01	92.5	82.5	93.8
0.02	91.3	81.3	93.8
0.04	90	81.3	95
0.06	85	81.3	93.8
0.08	85	75	92.5
0.1	78.8	72.5	92.5
0.15	77.5	70	91.3
0.2	80	62.5	86.3
0.25	73.8	62.5	85
0.3	73.8	62.5	85
0.35	43.8	47.5	85

0.4	43.8	47.5	73.8
0.6	41.3	43.8	57.5
0.8	41.3	43.8	57.5
1	16.3	17.5	18.8
1.2	5	5	5

Table 2: Table showing the relationships between the epsilon values and the face recognition accuracy

The average face recognition accuracy determined from the average values from the 3 data sets based on the epsilon values are represented as a graph as shown in Figure 38.

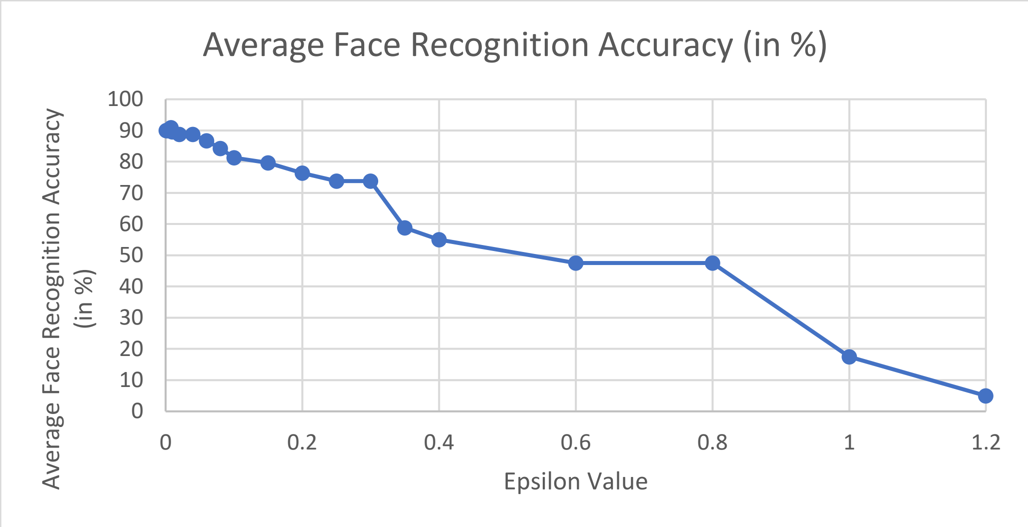


Figure 38:Relationship between the epsilon values and the average face recognition accuracy

7.3 Experiment Results for Experiment 3

Experiments were also conducted to determine how the number of training images for each person affects the face recognition accuracy. The eigen distance threshold value was set to **6000** and the epsilon value was set to **0.01** for each test.

After the experiment was conducted, the number of training images and testing images used for each person and their corresponding face recognition accuracy are shown in Table 3 and the graph in Figure 39 as follows:

Training Images for each person	Training Images (in total) for all 20 persons	Testing Images per person	Testing Images (in total) for all 20 persons	Number of Images Correctly Recognized	Number of Images Wrongly Recognized	**Accuracy of Face Recognition (in %)**
1	20	5	100	88	12	**88**
2	40	5	100	89	11	**89**
3	60	5	100	92	8	**92**
4	80	5	100	94	6	**94**
5	100	5	100	94	6	**94**
6	120	5	100	96	4	**96**
7	140	4	80	76	4	**95**
8	160	3	60	55	5	**91.7**
9	180	2	40	37	3	**92.5**

Table 3: Table showing the relationship between the number of training images per person and the Face Recognition Accuracy

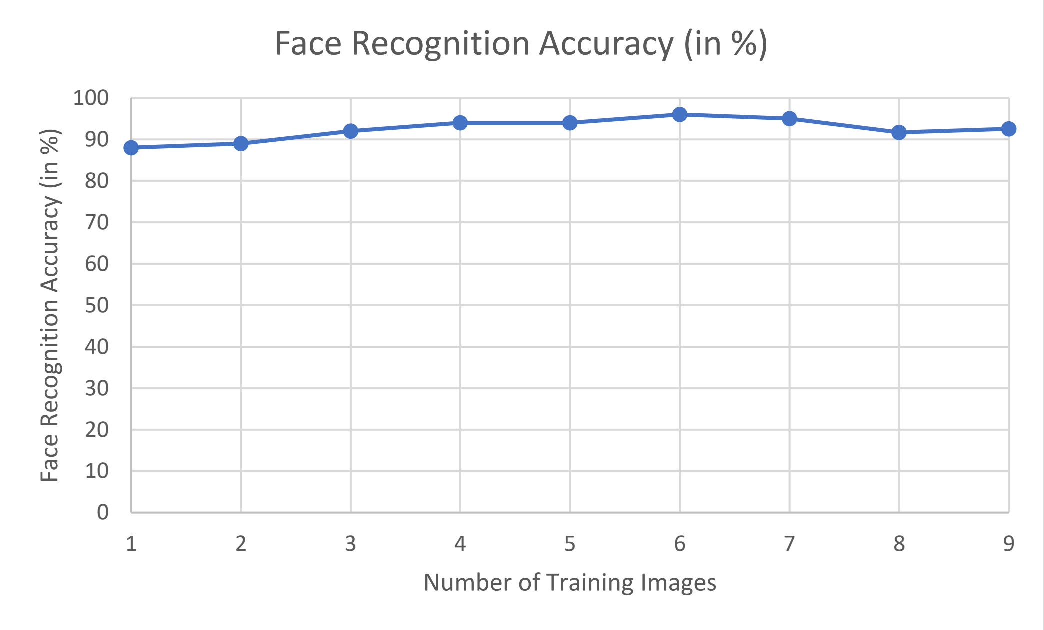


Figure 39: Figure showing the relationship between the number of training images per person and face recognition accuracy

7.4 Experiment Results for Experiment 4

This experiment was conducted to determine how the eigen distance threshold values affects the speed of the face recognition with the **epsilon value set** at a **constant value of 0.01.**

A total of 3 data sets were being used for the experiments. The testing images and training images were swapped randomly in all the 3 data sets for cross-validation to be done. The average face recognition speed from the results of the 3 data sets were then determined.

The experiment results in the 3 data sets for **experiment 4** which shows the relationship between the Eigen distance threshold values and the face recognition accuracy are shown in the graph in Figure 40 as follows:

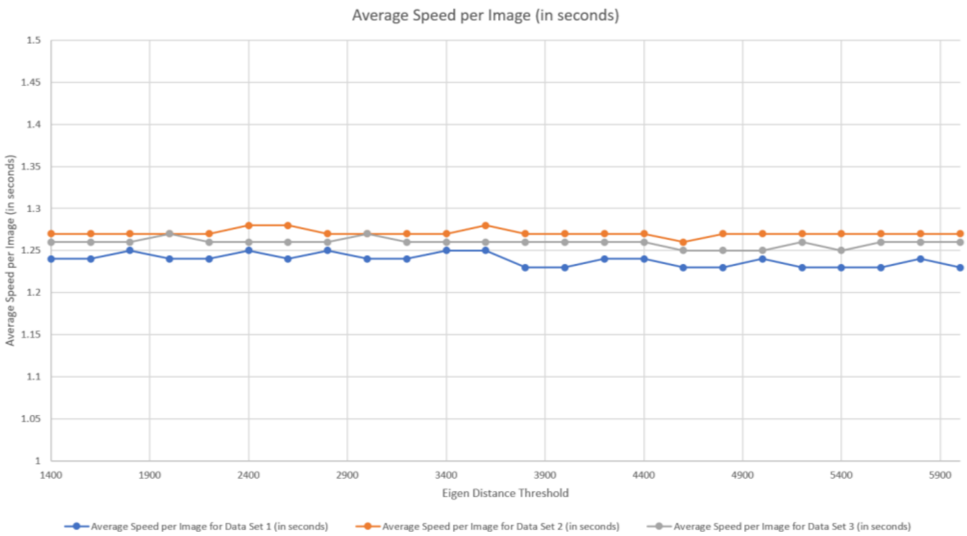


Figure 40: Figure showing the relationship between the Eigen distance threshold and the Average Speed per image (in seconds)

The average face recognition speed was determined from the average values from the 3 data sets and represented in the form of a graph as shown in Figure 41 as follows:

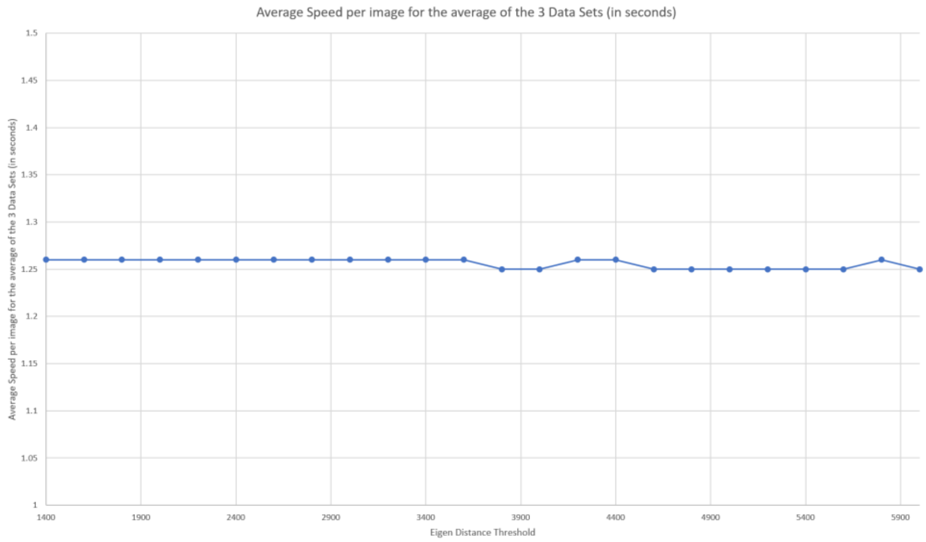


Figure 41

7.5 Experiment Results for Experiment 5

This experiment was conducted to determine how the epsilon values affect the face recognition speed with a constant eigen distance threshold value set to 5000.

A total of 3 data sets were being used for the experiments. The testing images and training images were swapped randomly in all the 3 data sets for cross-validation to be done. The average face recognition speed from the results of the 3 data sets were then determined.

The experiment results in the 3 data sets for **experiment 5** which shows the relationship between the epsilon values and the face recognition speed are shown in the Table 4 as follows:

Epsilon Value	Average Time Taken to Process each image in Data Set 1 (in seconds)	Average Time Taken to Process each image in Data Set 2 (in seconds)	Average Time Taken to Process each image in Data Set 3 (in seconds)
0.001	1.4	1.43	1.42
0.002	1.35	1.38	1.38
0.004	1.29	1.32	1.32
0.006	1.26	1.3	1.3
0.008	1.24	1.29	1.28
0.01	1.23	1.27	1.25
0.02	1.2	1.27	1.23
0.04	1.19	1.25	1.21
0.06	1.18	1.25	1.2
0.08	1.17	1.25	1.2
0.1	1.18	1.25	1.2

0.15	1.17	1.24	1.19
0.2	1.17	1.24	1.19
0.25	1.17	1.25	1.19
0.3	1.17	1.24	1.18
0.35	1.16	1.23	1.19
0.4	1.17	1.24	1.19
0.6	1.17	1.23	1.19
0.8	1.16	1.24	1.18
1	1.16	1.23	1.18
1.2	1.16	1.24	1.18

Table 4: Table showing the relationship between the epsilon values and the face recognition speed

The average face recognition speed will then be obtained from the 3 data sets and are displayed in the graph in Figure 42.

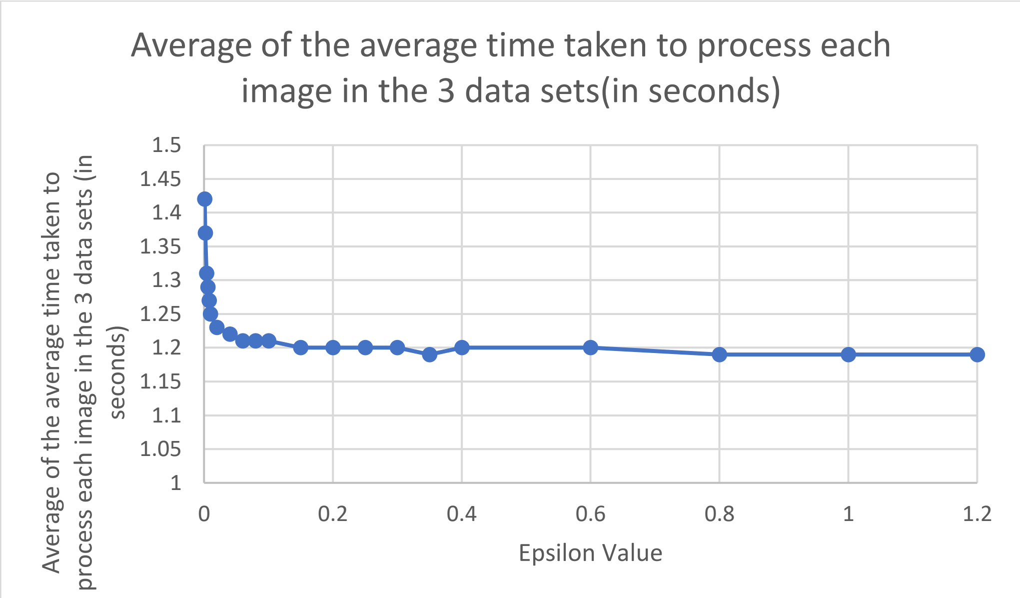


Figure 42: Figure showing the relationship between the epsilon values and the average time taken to process each image (in seconds)

7.6 Experiment Results for Experiment 6

Experiments were also conducted to determine how the number of training images affects the face recognition **speed**. The eigen distance threshold value was set to 6000 and the epsilon value was set to 0.01 for each test. The **average** time taken to process each image (in seconds) based on the number of training images are determined from the experiment. The experimental results were determined and displayed in Table 5 and the graph in Figure 43.

Training Images per person	Training Images (in total)	Testing Images per person	Testing Images (in total)	Time taken to process all images in the testing set (in seconds)	**Time taken to process each image (in seconds)**
1	20	5	100	10.8	**0.11**
2	40	5	100	18.4	**0.18**

3	60	5	100	30.2	**0.3**
4	80	5	100	46.6	**0.47**
5	100	5	100	67.8	**0.68**
6	120	5	100	92.4	**0.92**
7	140	4	80	98.5	**1.23**
8	160	3	60	96	**1.6**
9	180	2	40	79.4	**1.99**

Table 5: Table showing the relationship between the number of training images per person and time taken to process each image (in seconds)

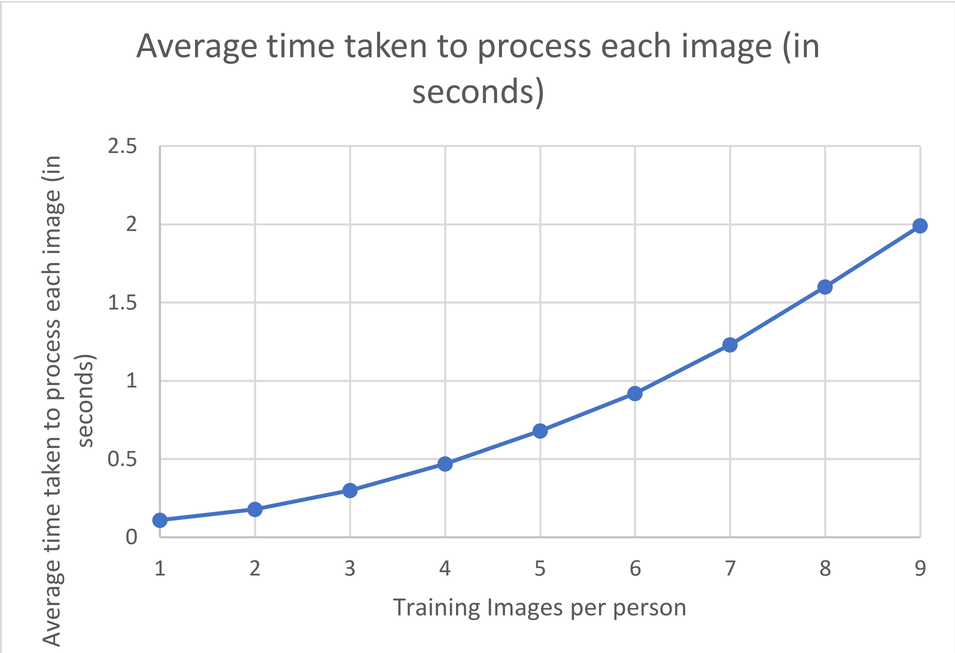


Figure 43: Figure showing the relationship between the number of training images per person and the time taken to process each image (in seconds)

7.7 Experiment Results for Experiment 7

This experiment was conducted to determine how the eigen distance threshold values affects the percentage of faces being detected as "UNKNOWN" (i.e. The faces in the testing images are not recognized even after comparing with all the training images in the training set).

A total of 3 data sets were being used for the experiments. The testing images and training images were swapped randomly in the 3 data sets for cross-validation to be done. The average percentage of faces being detected as "UNKNOWN" from the results of the 3 data sets were then determined.

The experiment results in the 3 data sets for **experiment 7** which shows the relationship between the Eigen distance threshold values and the percentage of faces being detected as "UNKNOWN" are shown in Table 6 as follows:

Eigen distance threshold	Percentage Of Unknown Faces For Data Set 1 (in %)	Percentage Of Unknown Faces For Data Set 2 (in %)	Percentage Of Unknown Faces For Data Set 3 (in %)
6000	0	0	0

5800	0	0	0
5600	0	0	0
5400	0	0	0
5200	0	0	0
5000	0	0	0
4800	1.3	0	0
4600	1.3	0	0
4400	2.5	0	2.5
4200	3.8	0	2.5
4000	5	2.5	3.8
3800	6.3	6.3	3.8
3600	10	6.3	5
3400	17.5	8.8	6.3
3200	20	15	10
3000	21.3	18.8	11.3
2800	27.5	22.5	15
2600	35	28.8	20
2400	47.5	38.8	31.3
2200	53.8	45	35
2000	63.8	53.8	45
1800	73.8	68.8	53.8
1600	82.5	82.5	62.5
1400	88.8	92.5	70

Table 6: Table showing the relationship between the Eigen distance threshold values and the percentage of faces being detected as "UNKNOWN" in the 3 data sets

The average percentage of UNKNOWN faces from the 3 data sets based on the eigen distance threshold values were then determined and displayed in the graph in Figure 44 as follows:

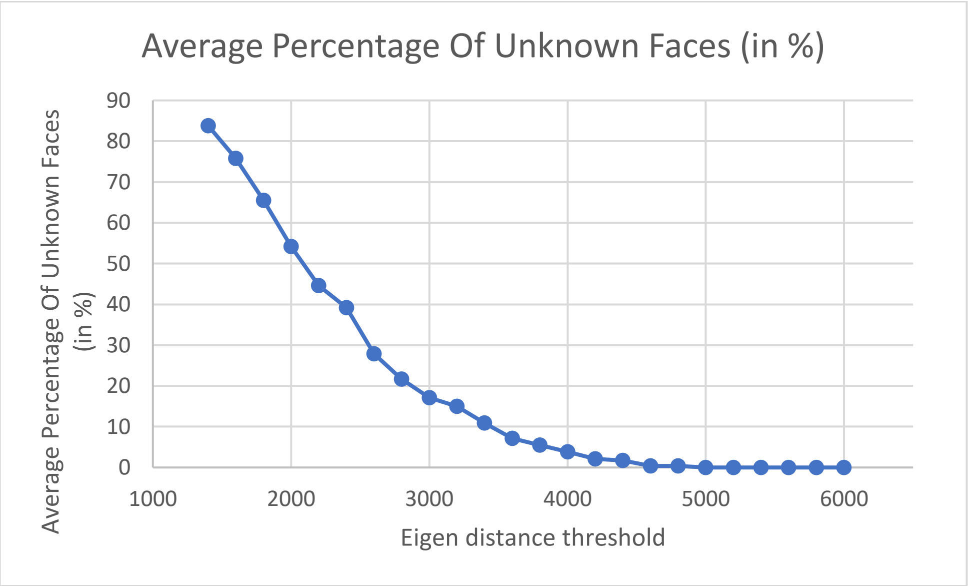


Figure 44: Figure showing the relationship between the Eigen distance threshold and the Average percentage of Unknown Faces detected

7.8 Experiment Findings

As can be seen in experiments 1 and 4, increasing the eigen distance threshold values will result in an increase in the face recognition accuracy while maintaining a constant face recognition speed. Eigen distance threshold values which are 5000 or larger gives the highest face recognition accuracy.

As can be seen in experiments 2 and 5, reducing the epsilon values will result in an increase in the face recognition accuracy but a longer time will be needed to process the images. An epsilon value of 0.01 gives the highest face recognition accuracy, with an average rate of 1.25 seconds needed to process each testing image.

As can be seen in experiments 3 and 6, the face recognition accuracy increases slightly when the number of training images increases from 2 to 6, where the highest face recognition accuracy is achieved when the number of training images is 6. When the number of training images is more than 6, the face recognition accuracy decreases slightly. Hence the optimal number of training images needed to achieve the best face recognition accuracy results is 6. The average time taken to process each image increases exponentially when the number of training images increases.

As can be seen in experiment 7, when the eigen distance threshold value increases, the percentage of faces detected as "UNKNOWN" (i.e. not recognized) decreases. The percentage of faces detected as "UNKNOWN" is 5% or less when the eigen distance threshold value is more than 4000.

In conclusion, from the above results, an eigen distance threshold value of 6000, epsilon value of 0.01 and when the number of training images is 6, the face recognition system gives the highest face recognition accuracy at a reasonable speed.

Chapter 8 Super-resolution techniques to improve the accuracy of face recognition

Many face recognition systems have been developed till date. However, it is generally assumed that the face region being used by these recognition systems are huge enough to consist of enough information for the recognition to take place. [15]
When we take photos of our classmates and friends using our mobile phones, the pictures taken may be of low-resolution. When low-resolution images or image sequences are used, face recognition becomes a very challenging task to accomplish. When the resolution of an image is decreased, there is a loss of facial component details. This will result in a reduction in recognition rates. [8]

8.1 Overview of problems to be addressed using super resolution

1) The camera being used in the face recognition based attendance system may be of low resolution and may not be able to recognize the faces of the students who are trying to take their attendance correctly. Using super-resolution techniques, the students' faces which were detected by the attendance system can be re-constructed to a high-resolution image in order for the students' faces to be recognized accurately by the system.

2) The training images extracted from the students' Facebook accounts to be placed into the training set might have been of low resolution due to the fact that they might have been taken by a low-resolution camera. Hence, using super-resolution techniques, high-resolution images

can be re-constructed from the low-resolution images in the training set. This will help to enhance the face recognition accuracy of the system.

Super-resolution refers to the task of estimating a high-resolution image from its low-resolution counterpart. [17] It helps to recover lost high-frequency details in face images. Super-resolution method reconstructs a high-resolution image or sequence from low resolution images by performing a bunch of operations from these low-resolution images. The high-resolution image which is reconstructed needs to preserve as much content as possible from the low-resolution image. The reconstructed high-resolution image will be less blurry and contains more detail than the low-resolution image. Hence, using super resolution algorithms, the resolution of an imaging system can be enhanced. This will result in an increase in the recognition rates by reproducing facial features in high resolution.

Super resolution techniques are mainly classified into reconstruction-based and recognition-based. Reconstruction-based techniques can super-resolve any image sequence so long as the motion between observations are able to be modelled. Recognition-based techniques learn the features of the low-resolution input images and synthesize the corresponding high-resolution output by trying to address the aliasing artefacts which are present in the low-resolution input image. The reconstruction process can be done by comparing a patch of pixels in the low-resolution input with the closest matching low-resolution patch in the training set. The patch of pixels with the low-resolution input will be replaced with its corresponding high-resolution patch. [11]

Super resolution techniques can recognize local features in low resolution images and improve the resolution of images in an appropriate manner by combining the non-redundant information in low-resolution images to form a higher resolution one. It takes away the degradations caused by the imaging process of a low-resolution camera. A super-resolved image can then be reconstructed by a set of images using super resolution. [10]

Super-resolution of the images which are captured by the camera used in the system are done by the following steps:

1. A dictionary full of facial images is generated for super-resolution.
2. These collected images are then made to become smaller in order to create rough images.
3. Similar patterns for the images are then taken out and registered in the dictionary as a set as shown in Figure 45.

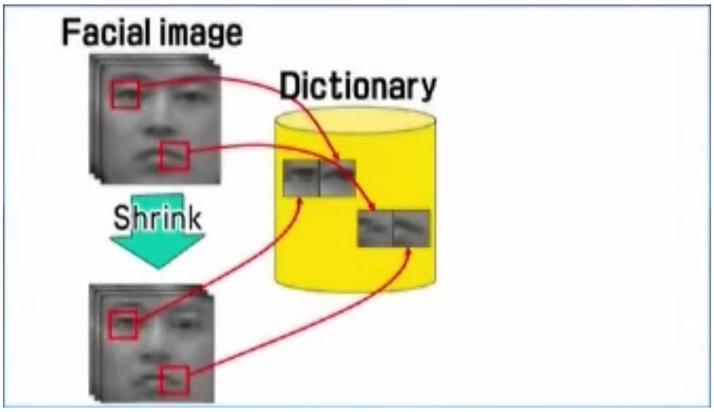


Figure 45

4. The dictionary becomes complete when other parts of the face are registered.
5. When a low-resolution image is entered, one facial area is extracted.
6. A similar facial pattern is then searched from the dictionary as shown in Figure 46.
7. When the pattern is found, the high-resolution version is being exported.

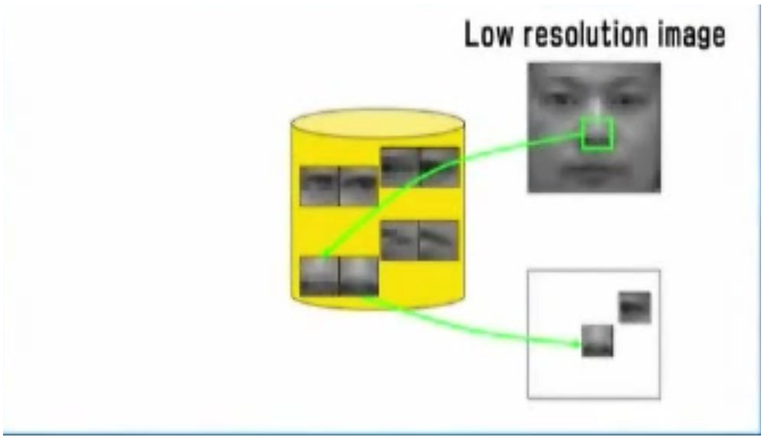


Figure 46

8. This form of image processing is then done for the whole face in order to create a super-resolution image. [11]

8.2 How users can use the super-resolution application

Users will need to super-resolve the images using the application through the following steps:

1. Create a folder to store the low-resolution images with the same image content. The content of the low-resolution images must be similar in content, even though they can be slightly displaced or rotated. For example, the folder may contain the following images as shown in Figure 47.

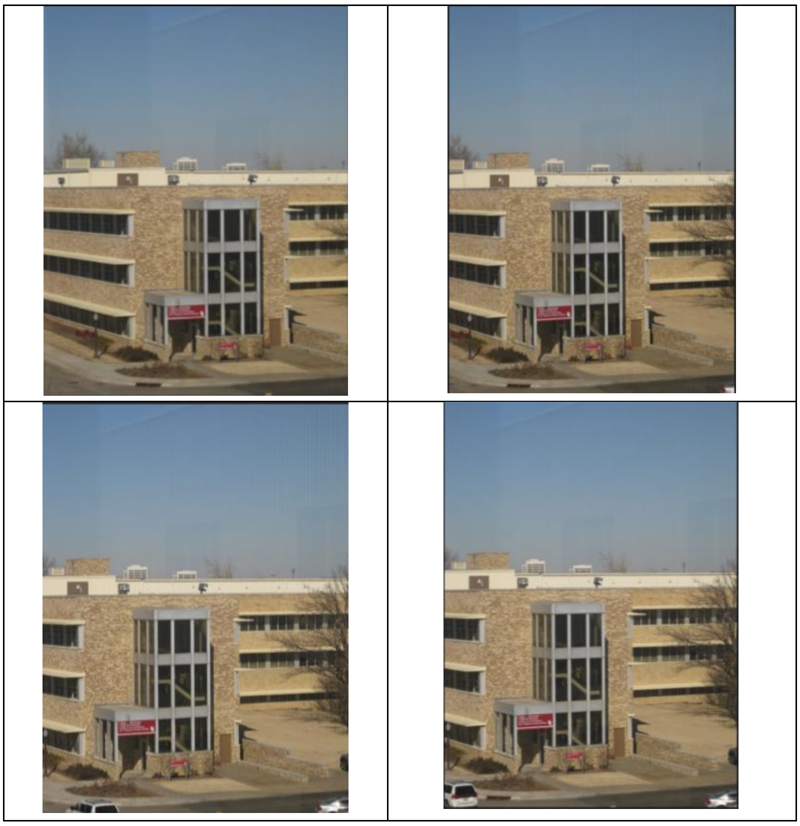


Figure 47: Figure showing the image content permitted in the list of low-resolution images

As can be seen in Figure 47, the image content in the four images are approximately the same even though the images are taken at slightly different angles from one another.

2. Crop the area of the low-resolution image which needs to be super-resolved and then click on ‘SR(Colour)’ if it is a coloured image. Otherwise, click on ‘SR(Grayscale)’ if it is a grayscale image.

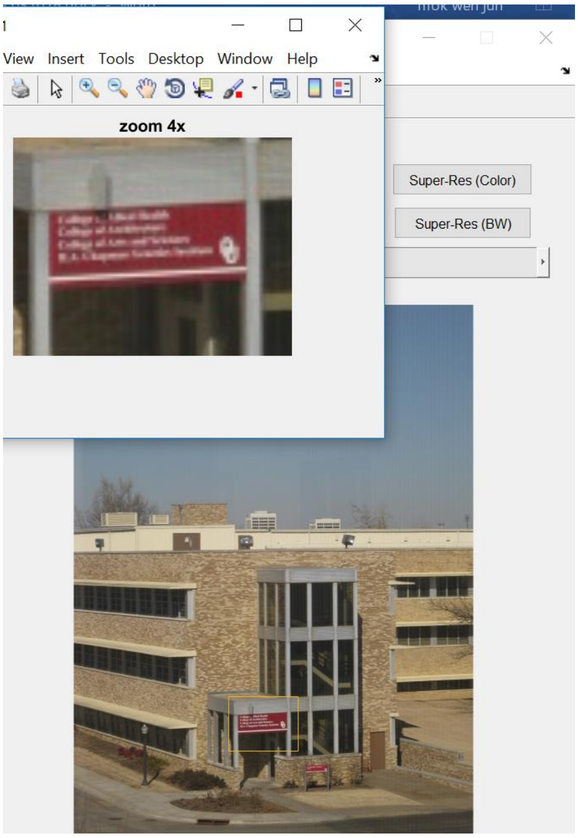


Figure 48: Figure showing a portion of a low-resolution image being cropped to be super-resolved

3. Wait for 2-5 minutes for the image to be super-resolved. The super-resolved image from the cropped area will be displayed as shown in Figure 49.

Figure 49: Figure showing the super-resolved image of the cropped portion in Figure 48

8.3 Facial images which were super-resolved

I have also conducted experiments on facial images as shown in Table 7 where the low-resolution face images (on the left of the table) are super-resolved to the high-resolution images. (on the right of the table).

Input Image	Super-Resolved Image
	
	
	
	
	

Table 7: Table showing the low-resolution images (left) and their corresponding super-resolved images (right)

Chapter 9 Recommendation in future work

The face recognition based attendance system can be further improved through the following ways:

1. The system can be further enhanced to cater to the needs of new students who join the class.
2. More variations among the images of students should be taken and placed into the training set since this will help to increase the face recognition accuracy. The training set should contain images of the same student with different facial expressions, with or without glasses, different positions of the face, viewing angles of the face and different lighting conditions as far as possible.
3. More research can be conducted to determine how the speed of face recognition can be improved when there are a large number of training images which may reduce the processing speed. An example may be to split the training set into smaller groups.
4. The system can be further enhanced to determine the mood of the students in class by detecting the expressions of the students. This can help to improve the ways teachers conduct their lessons to grab the students' attention.
5. A mobile application which is used as a face recognition based attendance system can also be developed. This mobile application will be a much more convenient way for students to take their attendance via face recognition as students do not need to queue up to get their attendance taken.
6. Other techniques such as the Local Binary Patterns Histograms (LBPH), Linear Discriminate Analysis, Fisherfaces and the Hidden Markov model can be explored to determine more effective means of face recognition.

Chapter 10 Conclusions

This project focuses on developing a face recognition based attendance system for students to take their attendance in a classroom. This system is very useful, efficient and convenient method for students to take their attendance. It saves a lot of the teachers' time and effort in taking the attendance of students especially when there are a large number of students in the class since up to five students are able to take their attendance at a time. It also helps to prevent errors in attendance taking and prevents students from signing the attendance of another student. Research was also being done on the various applications of face recognition in the market and how Principal Component Analysis and Eigenfaces can be used to recognize faces. Super-resolution techniques were also used to improve the resolution of low-resolution images to improve the face recognition accuracy of the system. Experiments were also conducted to determine how various factors such as the eigen distance threshold values, epsilon values and the number of training images affects the face recognition accuracy and face recognition speed of the system.

References

[1] Bouzidi, F., Elbouz, M., Alfalou, A., & Fakhfakh, A. (2016). Joint pre-processing phase to optical correlator for face recognition applications. *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP).*

[2] Facial recognition system. (2018, August 13). Retrieved from https://en.wikipedia.org/wiki/Facial_recognition_system

[3] Poornima, S., Sripriya, N., Vijayalakshmi, B., & Vishnupriya, P. (2017). Attendance monitoring system using facial recognition with audio output and gender classification. *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP).*

[4] MFlow - Seamless Airport Passenger Automation & Journey Management System. (n.d.). Retrieved from https://www.humanrecognitionsystems.com/product/mflow/

[5] Navarrete, P., & Ruiz-Del-Solar, J. (2002). Comparative Study between Different Eigenspace-Based Approaches for Face Recognition. *Advances in Soft Computing — AFSS 2002 Lecture Notes in Computer Science,* 178-184

[6] Turk, M., & Pentland, A. (1991). Eigenfaces for Recognition. *Journal of Cognitive Neuroscience, 3*(1), 71-86.

[7] V. (2014, January 19). PCA 10: Eigen-faces. Retrieved from https://www.youtube.com/watch?v=_lY74pXWlS8

[8] Lin, F., Fookes, C., Chandran, V., & Sridharan, S. (n.d.). Super-Resolved Faces for Improved Face Recognition from Surveillance Video. *Advances in Biometrics Lecture Notes in Computer Science,* 1-10.

[9] Bilgazyev, E., Efraty, B., Shah, S. K., & Kakadiaris, I. A. (2011). Improved face recognition using super-resolution. *2011 International Joint Conference on Biometrics (IJCB).*

[10] Super-resolution imaging. (2018, June 19). Retrieved from https://en.wikipedia.org/wiki/Super-resolution_imaging.

[11] N. (2015, July 14). Super-resolution Technology [NEC official]. Retrieved from https://www.youtube.com/watch?v=8DPHSwpDNVc.

[12] Kaur, R., & Himanshi, E. (2015). Face recognition using Principal Component Analysis. *2015 IEEE International Advance Computing Conference (IACC).*

[13] OpenCV: Face Detection using Haar Cascades. (n.d.). Retrieved from https://docs.opencv.org/3.4.1/d7/d8b/tutorial_py_face_detection.html

[14] Rojas, S.A. (n.d.). Multiple face detection and recognition in real time. Retrieved from https://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real

[15] Face super-resolution reconstruction and recognition using non-local similarity dictionary learning based algorithm. (2016). *IEEE/CAA Journal of Automatica Sinica,3*(2), 213-224.

[16] Viola–Jones object detection framework. (2018, June 19). Retrieved from https://en.wikipedia.org/wiki/Viola–Jones_object_detection_framework

[17] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., . . . Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[18] Andrade, N. N., Martin, A., & Monteleone, S. (2013). "All the better to see you with, my dear": Facial recognition and privacy in online social networks.

[19] Sennaar, K. (2018, September 11). Facial Recognition Applications - Security, Retail, and Beyond. Retrieved from https://www.techemergence.com/facial-recognition-applications/