SEA College of Science, Commerce and Arts

"AI Face Recognition and Attendance System"

A project report submitted in partial fulfillment of the requirement for Bachelor of Computer Application, Sixth semester under Bengaluru North University during the academic year 2022-2023.

Bachelor of Computer Application

Submitted By

Gagan S R R2013419

Under the Guidance of

Mrs. P Santhi Theja

Head of the Department of BCA

SEA College of Science, Commerce and Arts

Bengaluru North University
Ekta Nagar, Near Ayyappanagar Circle,
Devasandra Main Road, Vigronagar Post,
K.R.Puram, Bangalore-560049.

SEA College of Science, Commerce and Arts

Bengaluru North University

Certificate

This is to certify that the project entitled **"AI Face Recognition and Attendance System",** is a Bonfied work done by **Gagan S R (R2013419)** Under of guidance of **Mrs. P Santhi Theja, Head of the Department of BCA,** which is in partial fulfillment of the requirement for the award of Bachelor of Science from Bengaluru North University, November 2022.

Internal Guide **Head of the Department**

Principal

Examiners:

1) **Date:**

Place: Bangalore

2)

Acknowledgment

Above all, we want to thank God Almighty for the blessings, health, wisdom, knowledge, and strength he has given to us. It is only by his will that we are able to start and complete this project.

We wish to express our sincere gratitude to the Late A. Krishnappa, the chairman of South East Asian Educational Trust for providing good infrastructure and adequate equipment's for the proper functioning of the institution.

We also thank **Dr. Muthe Gowda T N,** the Principal of SEA College of Science, Commerce and Arts for his tireless effort in managing the College making sure that everything runs in proper order.

We sincerely thank **Mrs. P Santhi Theja,** Head of the Department of BCA for his constant support, advice, and guidance which greatly contributed to the successful completion of this project. We also wish to express our gratitude the all the staff in the Computer Science Department of their support, concern and encouragement not only to the project but also in our studies.

We also want to thank our parents for their guidance, support, encouragement, advice and prayers. And also, for providing all the materials we need, not only for our studies but also for our well-being if not for them, we will not be here today.

We also want to thank our brothers and sisters for their support and advice which helped in the successful completion of this project.

Project Report on

AI Face Recognition

And

Attendance System

FACE RECOGNITION BASED AUTOMATED STUDENT ATTENDANCE SYSTEM

ABSTRACT

Face is the representation of one's identity. Hence, we have proposed an automated student attendance system based on face recognition. Face recognition system is very useful in life applications especially in security control systems. The airport protection system uses face recognition to identify suspects and FBI (Federal Bureau of Investigation) uses face recognition for criminal investigations. In our proposed approach, firstly, video framing is performed by activating the camera through a user-friendly interface. The face ROI is detected and segmented from the video frame by using Viola-Jones algorithm. In the pre-processing stage, scaling of the size of images is performed if necessary in order to prevent loss of information. The median filtering is applied to remove noise followed by conversion of colour images.

CHAPTER 1
INTRODUCTION

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

1.1 Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision. Robinson-Riegler, G., & Robinson-Riegler, B. (2008) mentioned that after visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyse the information. The analysed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person (Margaret Rouse, 2012).

Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities (Robert Silk, 2017). Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes (Sidney Fussell, 2018). Furthermore, Intel Company allows the users to use face recognition to get access to their online account (Reichert, C., 2017). Apple allows the users to unlock their mobile phone, iPhone X by using face recognition (deAgonia, M., 2017).

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principle component analysis (PCA) to solve face recognition problem. Many studies on face recognition were then conducted continuously until today (Ashley DuVal, 2012).

1.2 Problem Statement

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasizes its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers does not have to count the number of students several times to ensure the presence of the students.

The paper proposed by Zhao, W et al. (2003) has listed the difficulties of facial identification. One of the difficulties of facial identification is the identification between known and unknown images. In addition, paper proposed by Pooja G.R et al. (2010) found out that the training process for face recognition student attendance system is slow and time-consuming. In addition, the paper proposed by Priyanka Wagh et al. (2015) mentioned that different lighting and head poses are often the problems that could degrade the performance of face recognition based student attendance system.

Hence, there is a need to develop a real time operating student attendance system which means the identification process must be done within defined time constraints to prevent omission. The extracted features from facial images which represent the identity of the students have to be consistent towards a change in background, illumination, pose and expression. High accuracy and fast computation time will be the evaluation points of the performance.

1.3 Aims and Objectives

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- To detect the face segment from the video frame.
- To extract the useful features from the face detected.
- To classify the features in order to recognize the face detected.
- To record the attendance of the identified student.

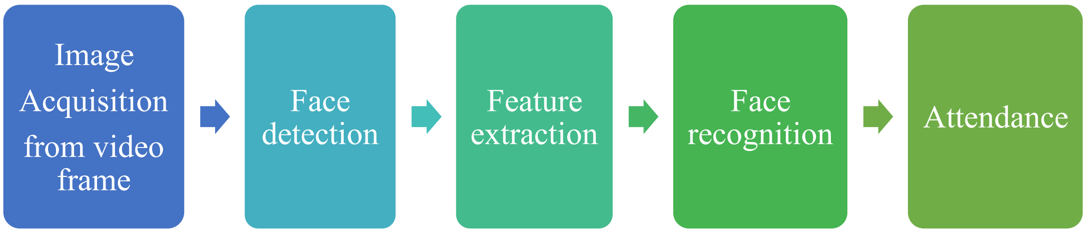


Figure 1.1 Block Diagram of the General Framework

1.4 Thesis Organization

Chapter 2 includes a brief review of the approaches and studies that have been done previously by other researchers whereas Chapter 3 describe proposed methods and approaches used to obtain the desired output. The results of the proposed approach would be presented and discussed in Chapter 4. The conclusion, as well as some recommendations would be included in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

2.1 Student Attendance System

Arun Katara et al. (2017) mentioned disadvantages of RFID (Radio Frequency Identification) card system, fingerprint system and iris recognition system. RFID card system is implemented due to its simplicity. However, the user tends to help their friends to check in as long as they have their friend's ID card. The fingerprint system is indeed effective but not efficient because it takes time for the verification process so the user has to line up and perform the verification one by one. However for face recognition, the human face is always exposed and contain less information compared to iris. Iris recognition system which contains more detail might invade the privacy of the user. Voice recognition is available, but it is less accurate compared to other methods. Hence, face recognition system is suggested to be implemented in the student attendance system.

Table 2.1 Advantages & Disadvantages of Different Biometric System (Arun Katara et al., 2017)

System type	**Advantages**	**Disadvantages**
RFID card system	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	-	Less accurate compared to others
Iris recognition system	Accurate	Privacy Invasion

2.2 Face Detection

Difference between face detection and face recognition are often misunderstood. Face detection is to determine only the face segment or face region from image, whereas face recognition is to identify the owner of the facial image. S.Aanjanadevi et al. (2017) and Wei-Lun Chao (2007) presented a few factors which cause face detection and face recognition to encounter difficulties. These factors consist of background, illumination, pose, expression, occlusion, rotation, scaling and translation. The definition of each factor is tabulated in Table 2.2.

Table 2.2 Factors Causing Face Detection Difficulties (S.Aanjanadevi et al., 2017)

Background	Variation of background and environment around people in the image which affect the efficiency of face recognition.
Illumination	Illumination is the variation caused by various lighting environments which degrade the facial feature detection.
Pose	Pose variation means different angle of the acquired the facial image which cause distortion to recognition process, especially for Eigen face and Fisher face recognition method.
Expression	Different facial expressions are used to express feelings and emotions. The expression variation causes spatial relation change and the facial-feature shape change.
Occlusion	Occlusion means part of the human face is unobserved. This will diminish the performance of face recognition algorithms due to deficiency information.
Rotation, scaling and translation	Transformation of images which might cause distortion of the original information about the images.

There are a few face detection methods that the previous researchers have worked on. However, most of them used frontal upright facial images which consist of only one face. The face region is fully exposed without obstacles and free from the spectacles.

Akshara Jadhav et al. (2017) and by P. Arun Mozhi Devan et al. (2017) suggested Viola-Jones algorithm for face detection for student attendance system. They concluded that out of methods such as face geometry- based methods, Feature Invariant methods and Machine learning based methods, Viola-Jones algorithm is not only fast and robust, but gives high detection rate and perform better in different lighting condition. Rahul V. Patil and S. B. Bangar (2017) also agreed that Viola-Jones algorithm gives better performance in different lighting condition. In addition, in the paper by Mrunmayee Shirodkar et al. (2015), they mentioned that Viola-Jones algorithm is able to eliminate the issues of illumination as well as scaling and rotation. In addition, Naveed Khan Balcoh (2012) proposed that Viola-Jones algorithm is the most efficient among all algorithms for instance the AdaBoost algorithm, the FloatBoost algorithm, Neural Networks, the S-AdaBoost algorithm, Support Vector Machines (SVM) and the Bayes classifier.

Varsha Gupta and Dipesh Sharma (2014) studied Local Binary Pattern (LBP), Adaboost algorithm, local successive mean quantization transform (SMQT) Features, sparse network of winnows (SNOW) Classifier Method and Neural Network-based face detection methods in addition to Viola-Jones algorithm. They concluded that Viola-Jones algorithm has the highest speed and highest accuracy among all the methods. Other methods for instance Local Binary Pattern and SMQT Features have simple computation and able to deal with illumination problem, their overall performance is weaker than Viola-Jones algorithm for face detection. The advantages and disadvantages of the methods is studied and tabulated in Table 2.3.

Table 2.3 Advantages & Disadvantages of Face Detection Methods (Varsha Gupta and Dipesh Sharma, 2014)

Face detection method	Advantages	Disadvantages
Viola jones algorithm	1. High detection speed 2. High accuracy.	1. Long training time. 2. Limited head pose. 3. Not able to detect dark faces.

Local Binary pattern	1. Simple computation. 2. High tolerance against the monotonic illumination changes.	1. Only used for binary and grey images. 2. Overall performance is inaccurate compared to Viola-Jones algorithm.
AdaBoost algorithm (part of Viola jones algorithm)	Need not to have any prior knowledge about face structure.	The result highly depends on the training data and affected by weak classifiers.
SMQT Features and SNOW Classifier Method	1. Capable to deal with lighting problem in object detection. 2. Efficient in computation.	The region contain very similar to grey value regions will be misidentified as face.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm.

2.2.1 Viola-Jones Algorithm

Viola-Jones algorithm which was introduced by P. Viola, M. J. Jones (2001) is the most popular algorithm to localize the face segment from static images or video frame. Basically the concept of Viola-Jones algorithm consists of four parts. The first part is known as Haar feature, second part is where integral image is created, followed by implementation of Adaboost on the third part and lastly cascading process.

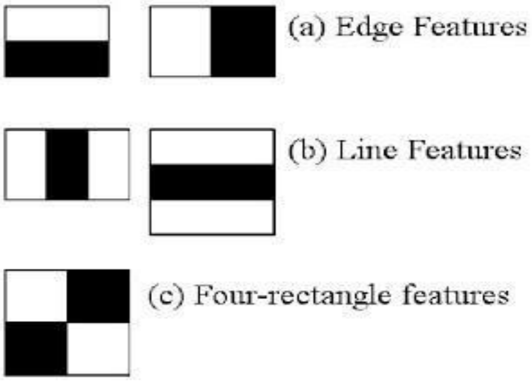


Figure 2.1 Haar Feature (Docs.opencv.org, 2018)

Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles (Mekha Joseph et al., 2016). Figure 2.1 shows several types of Haar features. The features perform as window function mapping onto the image. A single value result, which representing each feature can be computed by subtracting the sum of the white rectangle(s) from the sum of the black rectangle(s) (Mekha Joseph et al., 2016). The illustration is shown in Figure 2.2.

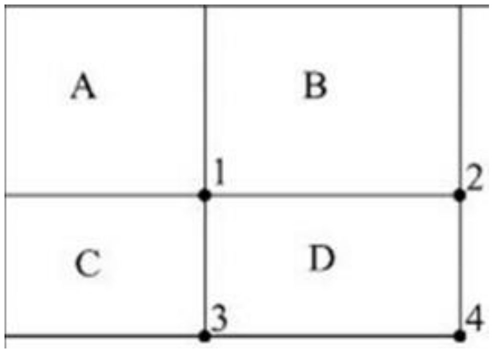


Figure 2.2 Integral of Image (Srushti Girhe et al., 2015)

The value of integrating image in a specific location is the sum of pixels on the left and the top of the respective location. In order to illustrate clearly, the value of the integral image at location 1 is the sum of the pixels in rectangle A. The values of integral image at the rest of the locations are cumulative. For instance, the value at location 2 is summation of A and B, (A + B), at location 3 is summation of A and C, (A + C), and at location 4 is summation of all the regions, (A + B + C + D) (Srushti

Girhe et al., 2015). Therefore, the sum within the D region can be computed with only addition and subtraction of diagonal at location 4 + 1 − (2 + 3) to eliminate rectangles A, B and C.

Burak Ozen (2017) and Chris McCormick (2013), they have mentioned that Adaboost which is also known as 'Adaptive Boosting' is a famous boosting technique in which multiple "weak classifiers" are combined into a single "strong classifier". The training set is selected for each new classifier according to the results of the previous classifier and determines how much weight should be given to each classifier in order to make it significant.

However, false detection may occur and it was required to remove manually based on human vision. Figure 2.3 shows an example of false face detection (circle with blue).

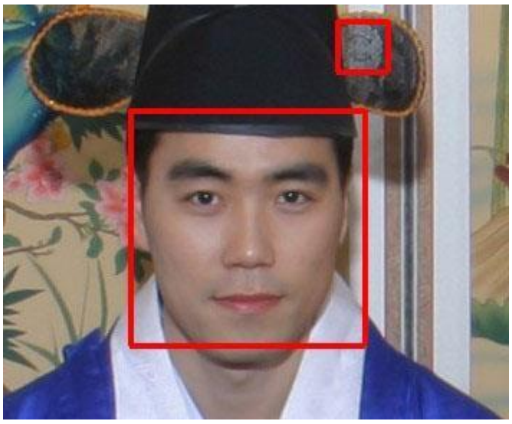


Figure 2.3 False Face Detection (Kihwan Kim, 2011)

2.3 Pre-Processing

Subhi Singh et al. (2015) suggested cropping of detected face and colour image was converted to grayscale for pre-processing. They also proposed affine transform to be applied to align the facial image based on coordinates in middle of the eyes and scaling of image to be performed. Arun Katara et al (2017), Akshara Jadhav et.al (2017), Shireesha Chintalapati, and M.V. Raghunadh (2013), all of the 3 papers have proposed histogram equalization to be applied to facial image, and scaling of images was performed for pre-processing.

Pre-processing enhances the performance of the system. It plays an essential role to improve the accuracy of face recognition. Scaling is one of the important preprocessing steps to manipulate the size of the image. Scaling down of an image increases the processing speed by reducing the system computations since the number of pixels are reduced. The size and pixels of the image carry spatial information. Gonzalez, R. C. and Woods (2008) mentioned spatial information is a measure of the smallest discernible detail in an image. Hence, spatial information has to be manipulated carefully to avoid distortion of images to prevent checkerboard effect. The size should be same for all the images for normalization and standardization purposes. Subhi Singh et al (2015) proposed PCA (Principal Component Analysis) to extract features from facial images, same length and width of image is preferred, thus images were scaled to 120 × 120 pixels.

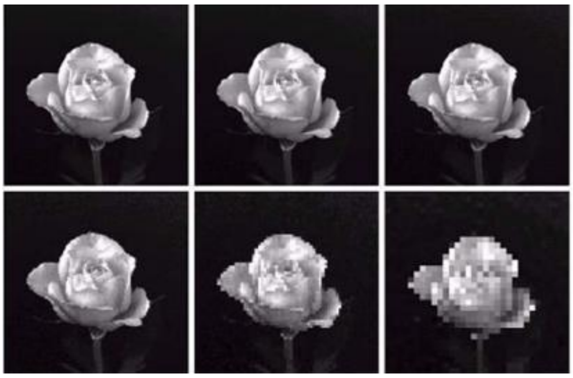


Figure 2.4 Images Show Checkerboard Effect Significantly Increasing from Left to Right (Gonzalez*,* R. C.*, &* Woods, 2008)

Besides scaling of images, colour image is usually converted to grayscale image for pre-processing. Grayscale images are believed to be less sensitive to illumination condition and take less computational time. Grayscale image is 8 bit image which the pixel range from 0 to 255 whereas colour image is 24 bit image which pixel can have 16 77 7216 values. Hence, colour image requires more storage space and more computational power compared to grayscale images. (Kanan and Cottrell, 2012). If colour image is not necessary in computation, then it is considered as noise.

In addition, pre-processing is important to enhance the contrast of images. In the paper of Pratiksha M. Patel (2016), he mentioned that Histogram equalization is one of the methods of pre-processing in order to improve the contrast of the image. It provides uniform distribution of intensities over the intensity level axis, which is able to reduce uneven illumination effect at the same time.

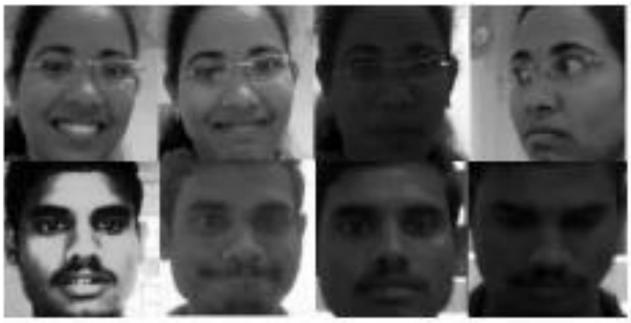


Figure 2.5 Facial Images Were Converted To Grayscale, Histogram Equalization Was Applied and Images Were Resized to 100x100 (Shireesha Chintalapati and M.V. Raghunadh, 2013)

There are a few methods to improve the contrast of images other than Histogram Equalization. Neethu M. Sasi and V. K. Jayasree (2013) studied Histogram Equalization and Contrast Limited Adaptive Histogram Equalization (CLAHE) in order to enhance myocardial perfusion images. Aliaa A. A. Youssif (2006) studied contrast enhancement together with illumination equalization methods to segment retinal vasculature. In addition, in paper by A., I. and E.Z., F. (2016) Image Contrast Enhancement Techniques and performance were studied. Unlike Histogram equalization, which operate on the data of the entire image, CLAHE operates on data of small regions throughout the image. Hence, the Contrast Limited Adaptive Histogram Equalization is believed to outperform the conventional Histogram Equalization. Summary of the literature review for contrast improvement is tabulated in Table 2.4.

Table 2.4 Summary of Contrast Improvement

Method	**Concept**	**Advantages**	**Disadvantages**

Histogram equalization	Contrast enhancement is performed by transforming the intensity values, resulting in uniformly distributed histogram.	1. Less sensitive to noise.	1. It depends on the global statistics of an image. 2. It cause over enhancement for some part, while peripheral region need more enhancement.
Contrast Limited Adaptive Histogram Equalization (CLAHE)	Unlike, HE which works on entire image, it works on small data regions. Each tile's contrast is enhanced to ensure uniformly distributed histogram. Bilinear interpolation is then used to merge the neighbouring tiles.	1. It prevent over enhancement as well as noise amplification.	1. More sensitive to noise compared to histogram equalization.

2.4 Feature Extraction

The feature is a set of data that represents the information in an image. Extraction of facial feature is most essential for face recognition. However, selection of features could be an arduous task. Feature extraction algorithm has to be consistent and stable over a variety of changes in order to give high accuracy result.

There are a few feature extraction methods for face recognition. In the paper of Bhuvaneshwari et al. (2017), Abhishek Singh and Saurabh Kumar (2012) and Liton Chandra Paul and Abdulla Al Sumam (2012), they proposed PCA for the face recognition. D. Nithya (2015) also used PCA in face recognition based student attendance system. PCA is famous with its robust and high speed computation.

Basically, PCA retains data variation and remove unnecessary existing correlations among the original features. PCA is basically a dimension reduction algorithm. It compresses each facial image which is represented by the matrix into single column vector. Furthermore, PCA removes average value from image to centralize the image data. The Principle Component of distribution of facial images is known as Eigen faces. Every single facial image from training set contributes to Eigen faces. As a result, Eigen face encodes best variation among known facial images. Training images and test images are then projected onto Eigen face space to obtain projected training images and projected test image respectively. Euclidean distance is computed by comparing the distance between projected training images and projected test image to perform the recognition. PCA feature extraction process includes all trained facial images. Hence, the extracted feature contains correlation between facial images in the training set and the result of recognition of PCA highly depends on training set image.

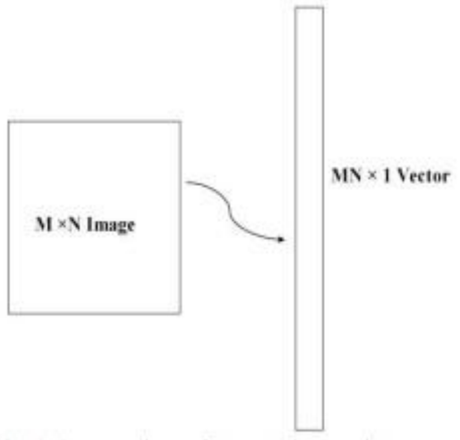


Figure 2.6 PCA Dimension Reduction (Liton Chandra Paul and Abdulla Al Sumam, 2012)

LDA (Linear discriminant analysis) also known as Fisher face is another popular algorithm for face recognition. In the paper by Suman Kumar Bhattacharyya and Kumar Rahul (2013), LDA was proposed for face recognition. LDA extract features by grouping images of the same class and separate images of different classes. LDA is able to perform well even with different facial expressions, illumination and pose due to its class separation characteristic. Same class is defined by facial images of the same individual, but with different facial expressions, varying lighting or pose, whereas facial images of person with different identity are categorized as different classes. Same class images yield within-class scatter matrix meanwhile different class images yield between-class scatter matrix. LDA manage to maximize the ratio of the determinant of the between-class scatter matrix over the determinant of the within class scatter matrix. LDA is believed to have lower error rates compared to PCA only if more samples per class are trained and small size of different class.

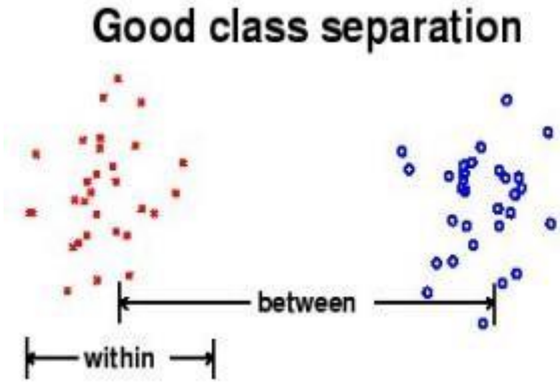


Figure 2.7 Class Separation in LDA (Suman Kumar Bhattacharyya and Kumar Rahul, 2013)

The original LBP (Local Binary Patterns) operator was introduced by the paper of Timo Ojala et al. (2002). In the paper by Md. Abdur Rahim et al. (2013), they proposed LBP to extract both texture details and contour to represent facial images. LBP divides each facial image into smaller regions and histogram of each region is extracted. The histograms of every region are concatenated into a single feature vector. This feature vector is the representation of the facial image and Chi square statistic is used to measure similarities between facial images. The smallest window size of each region is 3 by 3. It is computed by thresholding each pixel in a window where middle pixel is the threshold value. The neighborhood larger than threshold value is assigned to 1 whereas the neighborhood lower than threshold value is assigned to 0. Then the resulting binary pixels will form a byte value representing center pixel.

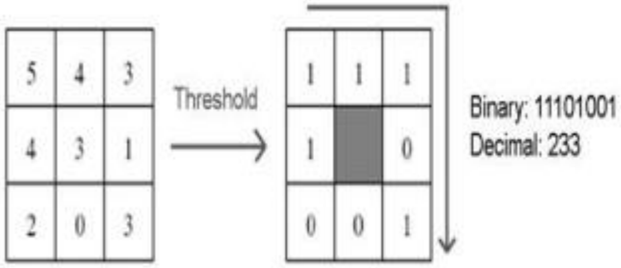


Figure 2.8 LBP Operator (Md. Abdur Rahim et.al, 2013)

LBP has a few advantages which make it popular to be implemented. It has high tolerance against the monotonic illumination changes and it is able to deal with variety of facial expressions, image rotation and aging of persons. These overwhelming characteristics cause LBP to be prevalent in real-time applications.

Neural network is initially used only in face detection. It is then further studied to be implemented in face recognition. In the paper by Manisha M. Kasar et al. (2016), Artificial Neural Network (ANN) was studied for face recognition. ANN consists of the network of artificial neurons known as "nodes". The nodes act as human brain in order to make recognition and classification. These nodes are interconnected and values are assigned to determine the strength of their connections. High value indicates strong connection. Neurons were categorized into three types of nodes or layers which are input nodes, hidden nodes, and output nodes. Input nodes are given weight based on its impact. Hidden nodes consist of some mathematical function and thresholding function to perform prediction or probabilities that determine and block unnecessary inputs and result is yield in output nodes. Hidden nodes can be more than one layer. Multiple inputs generate one output at the output node.

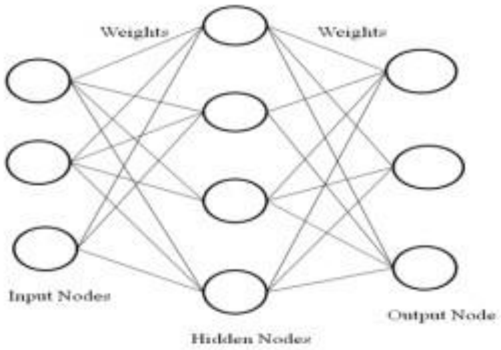


Figure 2.9 Artificial Neural Network (ANN) (Manisha M. Kasar et al., 2016)

Convolutional Neural Network (CNN) is another neural network algorithm for face recognition. Similar to ANN, CNN consists of the input layer, hidden layer and output layer. Hidden layers of a CNN consists of multiple layers which are convolutional layers, pooling layers, fully connected layers and normalization layers. However, a thousand or millions of facial images have to be trained for CNN to work accurately and it takes long time to train, for instance Deepface which is introduced by Facebook.

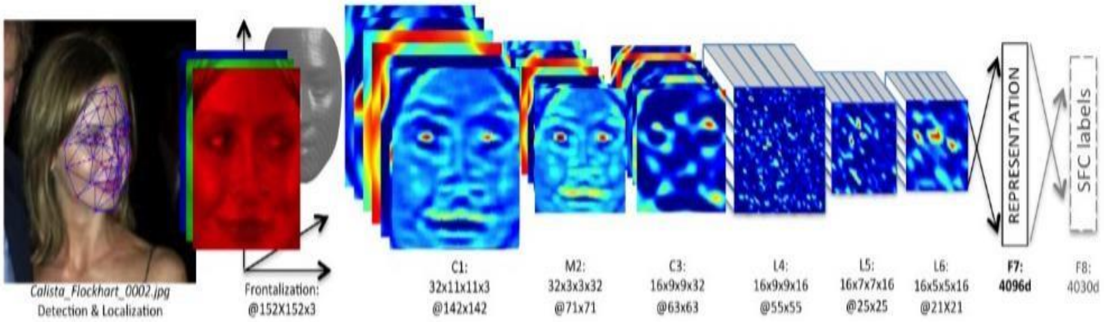


Figure 2.10 Deepface Architecture by Facebook (Yaniv Taigman et al, 2014)

2.4.1 Types of Feature Extraction

Divyarajsinh N. Parmar and Brijesh B. Mehta (2013) face recognition system can be categorized into a few Holistic-based methods, Feature-based methods and Hybrid methods. Holistic-based methods are also known as appearance-based methods, which mean entire information about a face patch is involved and used to perform some transformation to obtain a complex representation for recognition. Example of Holistic-based methods are PCA(Principal Component Analysis) and LDA(Linear dependent Analysis).On the other hand, feature-based methods directly extract detail from specific points especially facial features such as eyes, noses, and lips whereas other information which is considered as redundant will be discarded. Example of feature-based method is LBP (Local Binary Pattern). These methods mentioned are usually combined to exist as Hybrid method, for example Holistic-based method combine with Feature-based in order to increase efficiency.

2.5 Feature Classification And Face Recognition

Classification involves the process of identification of face. Distance classifier finds the distance between the test image and train image based on the extracted features. The smaller the distance between the input feature points and the trained feature points, the higher the similarity of the test image and training image. In other words, the facial images with the smallest/minimum distance will be classified as the same person. Deepesh Raj (2011) mentioned several types of distance classifiers such as Euclidean

Distance, City Block Distance and Mahalanobis distance for face recognition. Md. Abdur Rahim et al. (2013) implemented Chi-Square statistic as distance classifier for LBP operator. The equation of each classification method is defined below.

Chi square distance is defined as

(2.1)

$$\chi^2 = \sum \frac{\text{oberved frequency} - \text{expected frequency}}{\text{expected frequency}}.$$

Chi-square statistic is usually used to compare between two bins of histogram. The City Block Distance or Manhattan Distance is known as L1-norm which is defined in

(2.2)

$$d(x, y) = |x - y|$$

The Euclidean distance is known as L2-norm which is defined in

(2.3)

$$d(x, y) = |x - y|^2$$

where, X is the input feature points and Y is the trained featured points.

The Mahalanobis distance is defined in

(2.4)

$$d(x, y) = \frac{(y - m_x)}{S_x}(y - m_x)^T$$

where m_x is mean of x and S_x is covariance matrix of x.

Md. Abdur Rahim et.al (2013), after performing the LBP feature extraction, Chi-Square statistic is suggested to be used as dissimilarity measures for histograms to compute the distance between two images. Abhishek Singh and Saurabh Kumar (2012) proposed Euclidean distance to compute the distance between two images after PCA feature extraction was performed. Threshold can be set for the distance calculated from the classifier. A face is classified as belonging to a class only if its distance is below the chosen threshold, otherwise the face is classified as unknown.

2.6 Evaluation

Different databases are used in order to evaluate the system performance. The database provided by previous researchers with different variable conditions, for example, lighting and expression will be used to justify the system and for study purpose. Furthermore, our own database will be used to analyse the system for real time application. From the literature review of the previous researchers, the common method to justify the performance of the system is by finding the accuracy of recognition.

The formula for accuracy or recognition rate is defined below:

$$accuracy = \frac{total\ matched\ images}{total\ tested\ images} x100 \tag{2.5}$$

Table 2.5 Summary of Feature Extraction, The Accuracy Obtained from Handbook of Research on Emerging Perspectives in Intelligent Pattern Recognition (NK Kamila, 2015)

Method	Advantages	Disadvantages	Accuracy (ATT database)
Eigen face/ Kernel PCA (Principal component Analysis)	High speed in training and recognition.	Face recognition is depend on training database.	77.97 %

Fisher face/ LDA (Linear Discriminant Analysis)	Images of individual with different illumination, facial expressions able to be recognized if more samples are trained.	1. Bigger database is required because images of different expression of the individual have to be trained in same class. 2. It depend more on database compared to PCA.	82.45 %
LBP(Local Binary Pattern)	It is able to overcome variety of facial expressions, varying illumination, image rotation and aging of person.	Training time is longer than PCA and LDA.	90.93 %
Neural network	High accuracy only if large database is trained.	1. Required long time to train. 2. Database is extremely large to have high accuracy.	N.A

CHAPTER 3

METHODOLOGY

3.1 Methodology Flow

The approach performs face recognition based student attendance system. The methodology flow begins with the capture of image by using simple and handy interface, followed by pre-processing of the captured facial images, then feature extraction from the facial images, subjective selection and lastly classification of the facial images to be recognized. Both LBP and PCA feature extraction methods are studied in detail and computed in this proposed approach in order to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. An algorithm to combine enhanced LBP and PCA is also designed for subjective selection in order to increase the accuracy. The details of each stage will be discussed in the following sections.

The flow chart for the proposed system is categorized into two parts, first training of images followed by testing images (recognize the unknown input image) shown in Figure 3.1 and Figure 3.2 respectively.

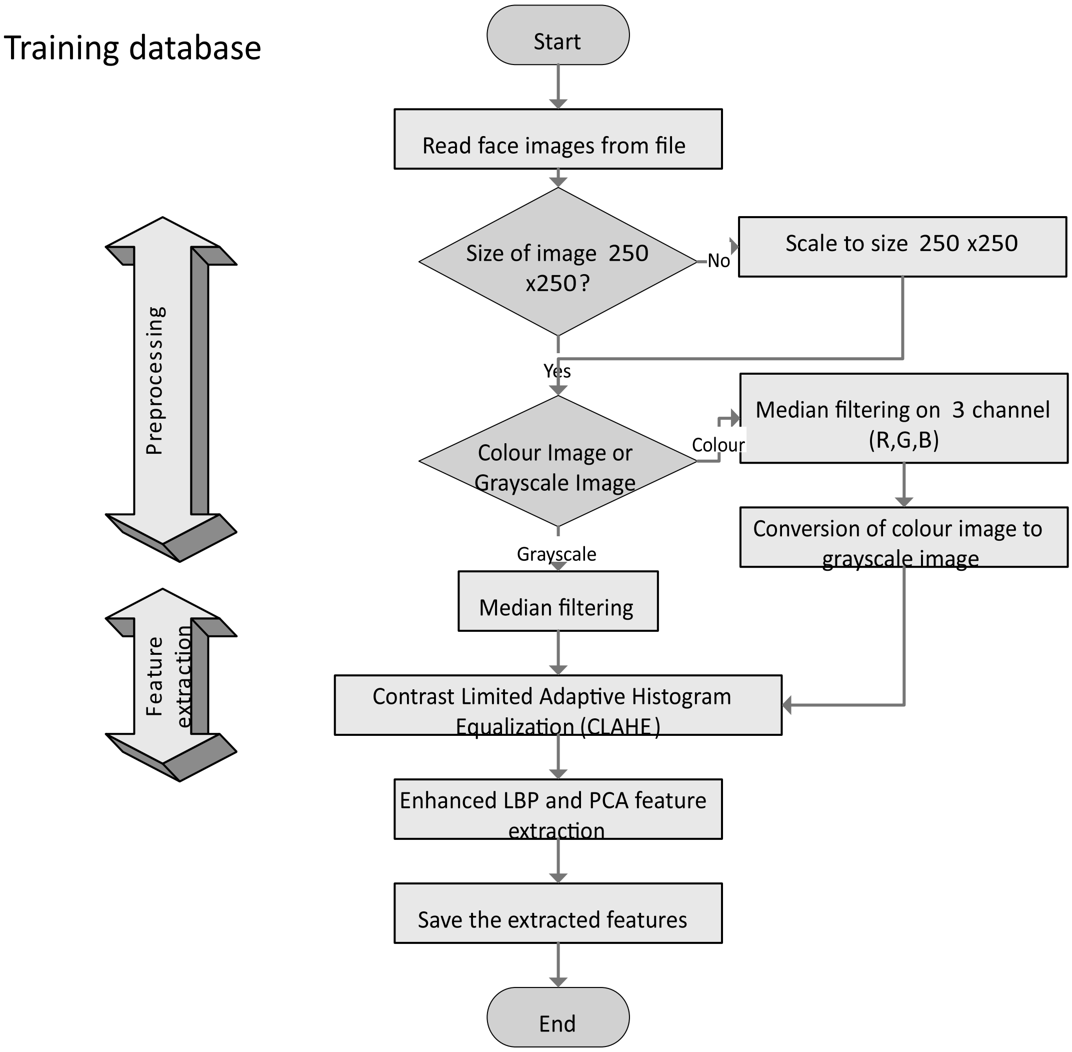

Training database
Start
Read face images from file
Size of image 250 x250?
No
Scale to size 250 x250
Yes
Preprocessing
Colour Image or Grayscale Image
Colour
Median filtering on 3 channel (R,G,B)
Conversion of colour image to grayscale image
Grayscale
Median filtering
Feature extraction
Contrast Limited Adaptive Histogram Equalization (CLAHE)
Enhanced LBP and PCA feature extraction
Save the extracted features
End

Figure 3.1 Flow of the Proposed Approach (Training Part)

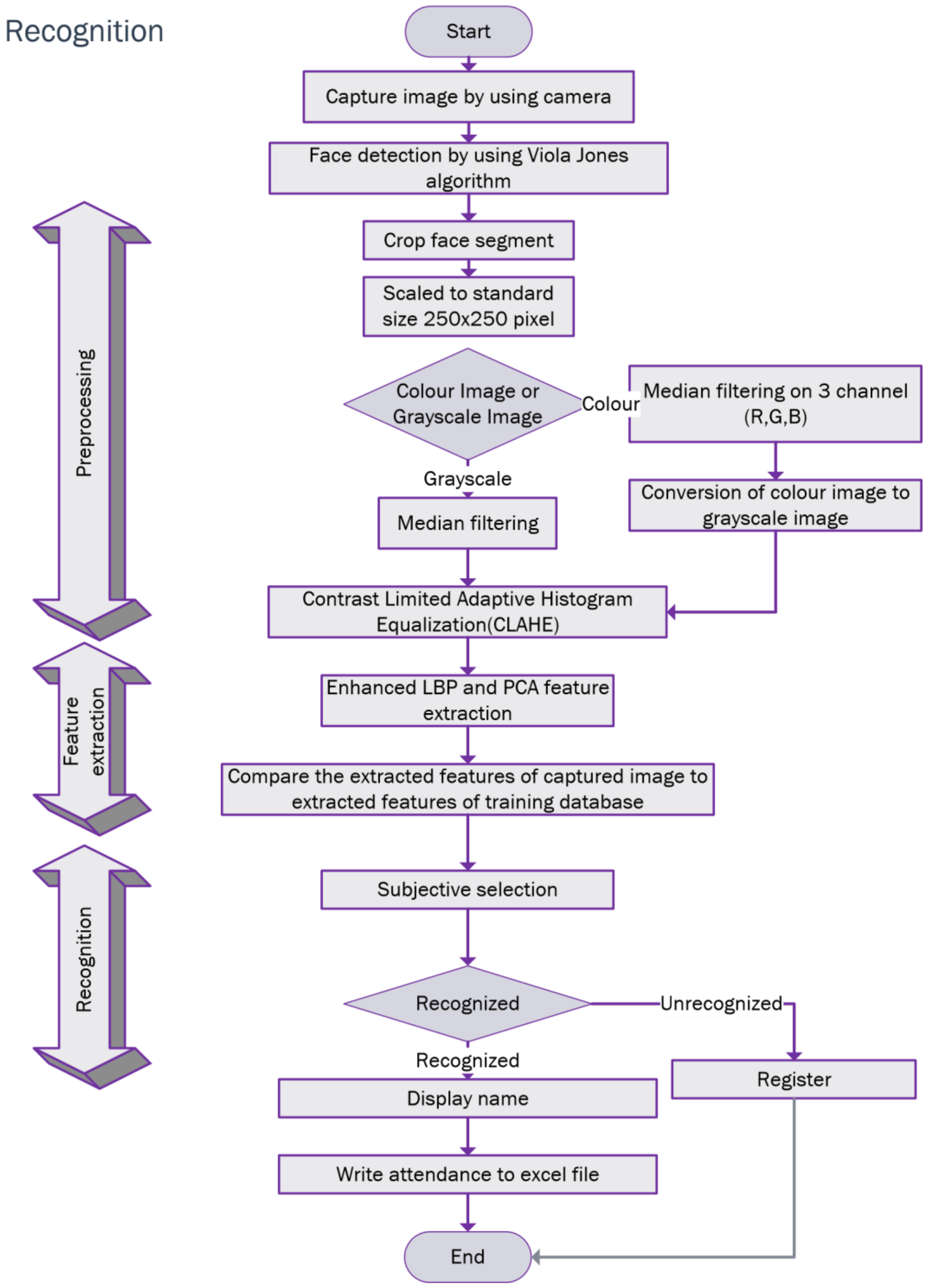

Recognition
Start
Capture image by using camera
Face detection by using Viola Jones algorithm
Crop face segment
Scaled to standard size 250x250 pixel
Colour Image or Grayscale Image
Colour
Median filtering on 3 channel (R,G,B)
Grayscale
Median filtering
Conversion of colour image to grayscale image
Contrast Limited Adaptive Histogram Equalization(CLAHE)
Enhanced LBP and PCA feature extraction
Compare the extracted features of captured image to extracted features of training database
Subjective selection
Recognized
Unrecognized
Recognized
Display name
Register
Write attendance to excel file
End
Preprocessing
Feature extraction
Recognition

Figure 3.2 Flow of the Proposed Approach (Recognition Part)

3.2 Input Images

Although our own database should be used to design real time face recognition student attendance system, the databases that are provided by the previous researchers are also used to design the system more effectively, efficiently and for evaluation purposes.

Yale face database is used as both training set and testing set to evaluate the performance. Yale face database contains one hundred and sixty-five grayscale images of fifteen individuals. There are eleven images per individual; each image of the individual is in different condition. The conditions included centre-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised and wink. These different variations provided by the database is able to ensure the system to be operated consistently in variety of situations and conditions.

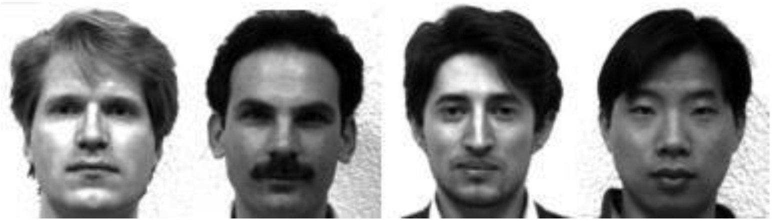


Figure 3.3 Sample Images in Yale Face Database (Cvc.cs.yale.edu, 1997)

For our own database, the images of students are captured by using laptop built in camera and mobile phone camera. Each student provided four images, two for training set and two for testing set. The images captured by using laptop built in camera are categorized as low quality images, whereas mobile phone camera captured images are categorized as high quality images. The high quality images consists of seventeen students while low quality images consists of twenty-six students. The recognition rate of low quality images and high quality images will be compared in Chapter 4 to draw a conclusion in term of performance between image sets of different quality.

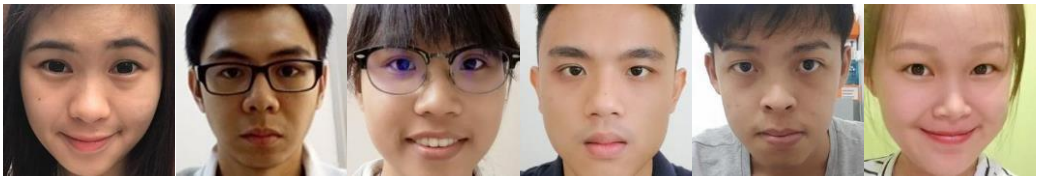


Figure 3.4 Sample of High Quality Images

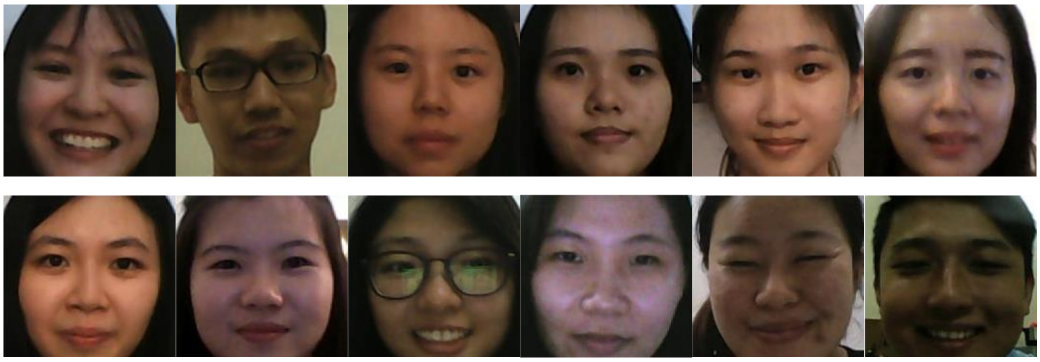


Figure 3.5 Sample of Low Quality Images

3.2.1 Limitations of the Images

The input image for the proposed approach has to be frontal, upright and only a single face. Although the system is designed to be able to recognize the student with glasses and without glasses, student should provide both facial images with and without glasses to be trained to increase the accuracy to be recognized without glasses. The training image and testing image should be captured by using the same device to avoid quality difference. The students have to register in order to be recognized. The enrolment can be done on the spot through the user-friendly interface.

These conditions have to be satisfied to ensure that the proposed approach can perform well.

3.3 Face Detection

Viola-Jones object detection framework will be used to detect the face from the video camera recording frame. The working principle of Viola-Jones algorithm is mentioned in Chapter 2. The limitation of the Viola-Jones framework is that the facial image has to be a frontal upright image, the face of the individual must point towards the camera in a video frame.

3.3.1 Pre-Processing

Testing set and training set images are captured using a camera. There are unwanted noise and uneven lighting exists in the images. Therefore, several pre-processing steps are necessary before proceeding to feature extraction.

Pre-processing steps that would be carried out include scaling of image, median filtering, conversion of colour images to grayscale images and adaptive histogram equalization. The details of these steps would be discussed in the later sections.

3.3.1.1 Scaling of Image

Scaling of images is one of the frequent tasks in image processing. The size of the images has to be carefully manipulated to prevent loss of spatial information. (Gonzalez, R. C., & Woods, 2008), In order to perform face recognition, the size of the image has to be equalized. This has become crucial, especially in the feature extraction process, the test images and training images have to be in the same size and dimension to ensure the precise outcome. Thus, in this proposed approach test images and train images are standardize at size 250 × 250 pixels.

3.3.1.2 Median Filtering

Median filtering is a robust noise reduction method. It is widely used in various applications due to its capability to remove unwanted noise as well as retaining useful detail in images. Since the colour images captured by using a camera are RGB images, median filtering is done on three different channels of the image. Figure 3.3 shows the image before and after noise removal by median filtering in three channels. If the input image is a grayscale image, then the median filtering can be performed directly without separating the channels.

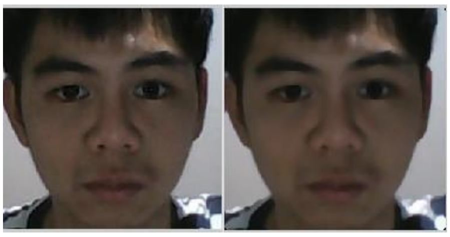


Figure 3.6 Median Filtering Done on Three Channels

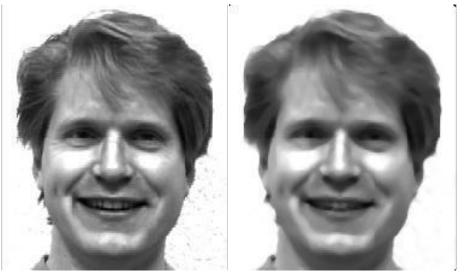


Figure 3.7 Median Filtering Done on a Single Channel

3.3.1.3 Conversion to Grayscale Image

Camera captures color images, however the proposed contrast improvement method CLAHE can only be performed on grayscale images. After improving the contrast, the illumination effect of the images able to be reduced. LBP extracts the grayscale features from the contrast improved images as 8 bit texture descriptor (Ojala, T. et al., 2002).Therefore, color images have to be converted to grayscale images before proceeding to the later steps. By converting color images to grayscale images, the complexity of the computation can be reduced resulting in higher speed of computation (Kanan and Cottrell, 2012). Figure 3.4 shows the conversion of images to grayscale image.

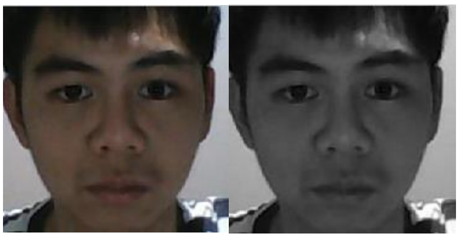
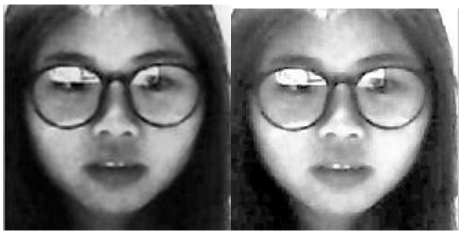


Figure 3.8 Conversion of Image to Grayscale Image

3.3.1.4 Contrast Limited Adaptive Histogram Equalization

Histogram equalization or histogram stretching is a technique of image contrast enhancement. (Pratiksha M. Patel, 2016). The contrast improvement is usually performed on the grayscale images. Image contrast is improved by stretching the range of its pixel intensity values to span over the desired range of values, between 0 and 255 in grayscale. The reason that Contrast Limited Adaptive Histogram Equalization (CLAHE) is used instead of histogram equalization is because histogram equalization depends on the global statistics. Hence, it causes over enhancement of some parts of image while other parts are not enhanced properly. This distorts the features of the image. It is a serious issue because the features of the image have to be extracted for the face recognition. Thus, CLAHE which is depend on local statistic is used. The result of CLAHE will be discussed in Chapter 4.



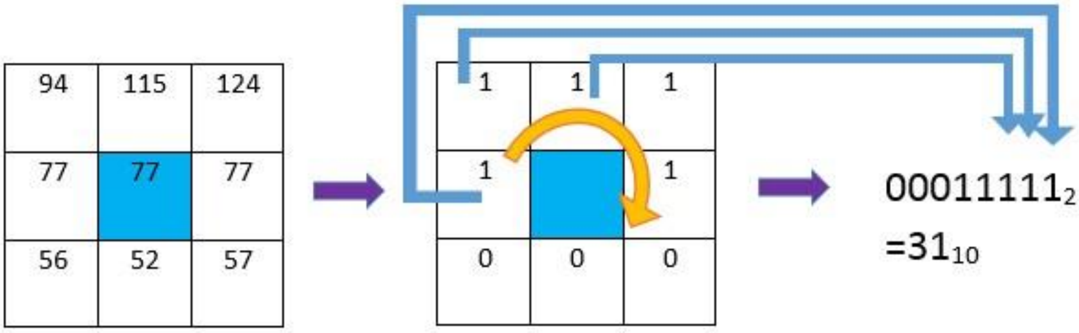
CLAHE HE Figure 3.9 Contrast Improvement

3.4 Feature Extraction

Different facial images mean there are changes in textural or geometric information. In order to perform face recognition, these features have to be extracted from the facial images and classified appropriately. In this project, enhanced LBP and PCA are used for face recognition. The idea comes from nature of human visual perception which performs face recognition depending on the local statistic and global statistic features. Enhanced LBP extracts the local grayscale features by performing feature extraction on a small region throughout the entire image. On the other hand, PCA extracts the global grayscale features which means feature extraction is performed on the whole image.

3.4.1 Working Principle of Original LBP

LBP is basically a texture based descriptor which it encoded local primitive into binary string. (Timo Ojala et al., 2002). The original LBP operator works on a 3×3 mask size. 3×3 mask size contains 9 pixels. The center pixel will be used as a threshold to convert the neighboring pixels (the other 8 pixels) into binary digit. If the neighboring pixel value is larger than the center pixel value, then it is assigned to 1, otherwise it is assigned to 0. After that, the neighborhoods pixel bits are concatenated to a binary code to form a byte value representing the center pixel. Figure 3.6 shows an example of LBP conversion.



94	115	124
77	77	77
56	52	57

1	1	1
1		1
0	0	0

00011111_2
$=31_{10}$

Figure 3.10 Example of LBP Conversion

$$LBP = \sum_{0}^{7} f(Pn - Pc).2^n \tag{3.1}$$

where Pc indicates centre pixel and Pn (n = 0,…, 7) are 8 of its neighbouring pixels respectively.

The starting point of the encoding process can be any of neighbouring pixels as long as the formation of binary string is following the order either in clockwise or anticlockwise rotation. The thresholding function f(y) can be written as follows

(3.2)

$$f(y) = \begin{cases} 0 & y < 0; \\ 1 & y \geq 0; \end{cases}$$

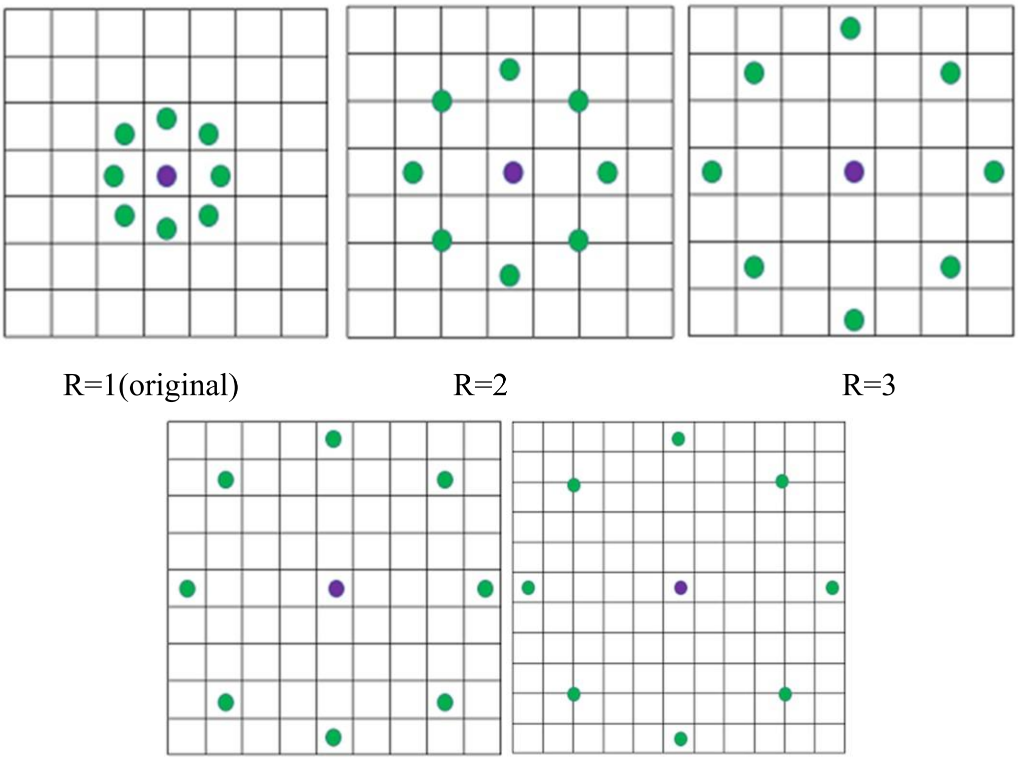
3.4.2 Working Principle of Proposed LBP

The original LBP operator is composed of 3 × 3 filter size with 9 pixels. Instead of the circular pattern, it looks more rectangular in shape. The 9 pixels adjacent to each other means every detail will be taken as sampling points even the non-essential details. It is more affected by uneven lighting condition because the small filter size emphasizes small scale detail (Lee and Li, 2007), even the shadow created by non-uniform lighting condition. In our proposed approach, a larger radius size, R is implemented in LBP operator. In the paper of Md. Abdur Rahim et.al (2013), the equation of modifying the radius size has been introduced. However, the paper did not mention the effect of changing the radius size. In the proposed approach, analysis is done on different radius sizes in order to enhance the system and reduce the illumination effect. By increasing the radius size, the filter size will be increased. R indicates radius from the centre pixel, θ indicates the angle of the sampling point with respect to the center pixel and P indicates number of sampling points on the edge of the circle taken to compare with the centre pixel. Given the neighbouring's notation (P, R,θ) is implemented, the coordinates of the centre pixel (Xc, Yc) and the coordinates of the P neighbours (Xp, Yp) on the edge of the circle with radius R can be computed with the sines and cosines shown in the equation (Md. Abdur Rahim et.al,2013):

(3.3)

$$Xp = Xc + Rcos(\theta/P)$$
$$Yp = Yc + Rsin(\theta/P)$$

Although the radius has been increased, total 8 sampling points are taken which is similar to the original LBP operator. In the approach, CLAHE is performed on the grayscale input facial images to improve the contrast. The contrast improved images remain as grayscale images. The proposed LBP operator extracts the grayscale features from the contrast improved grayscale images which requires only 8 bit computation. After that, the pixels at the sampling points will be encoded as 8 bit binary string in the same way as original LBP operator encoding process. Enhanced LBP with radius size two, perform better compared to original LBP and has more consistent recognition rate compared to other radius size. Hence, enhanced LBP with radius size two will be used as proposed approach. The proposed LBP operator will be further explained in Chapter 4 (result and discussion).



R=1(original) R=2 R=3

R=4 R=5

Figure 3.11 LBP with Different Radius Sizes

Basically, the increasing in the size of the radius means extending the circular pattern of LBP externally. The green spots within the blocks indicate the sampling pixels to be encoded into binary string. For the sampling pixel located in between the blocks, it indicates the average pixel value is computed from the adjacent pixels (diagonal).

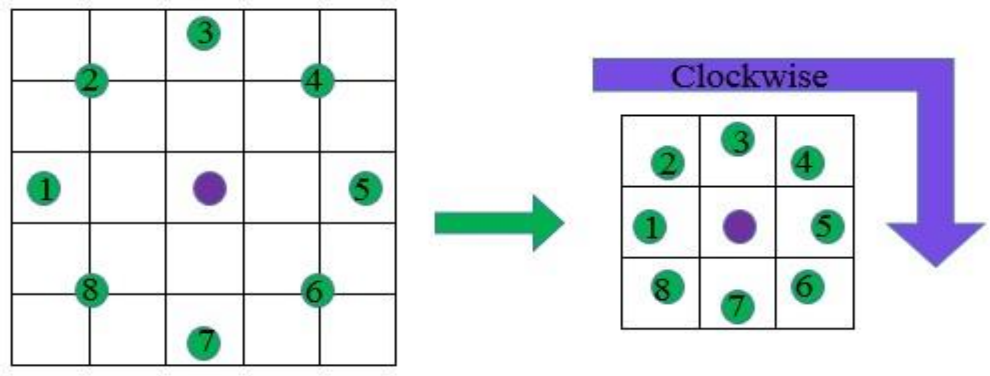


Figure 3.12 Proposed LBP Operator with Radius 2 and Its Encoding Pattern.

The feature vector of the image is constructed after the Local Binary Pattern of every pixel is calculated. The histogram of the feature vector image is computed in order to be classified by distance classifier. However, it loss spatial information because histogram representation does not include spatial information but only discrete information. (Gonzalez, R. C., & Woods, 2008). In order to overcome this problem, the feature vector image is then divided into blocks. A histogram is constructed in each region respectively. Every bin in a histogram represents a pattern and contains the frequency of its appearance in the region. The feature vector of entire image is then constructed by concatenating the regional histograms in the sequence to one histogram. (Md. Abdur Rahim et al., 2013). This histogram remains its regional spatial information and represents the identity of single image which is then classified to perform the recognition.

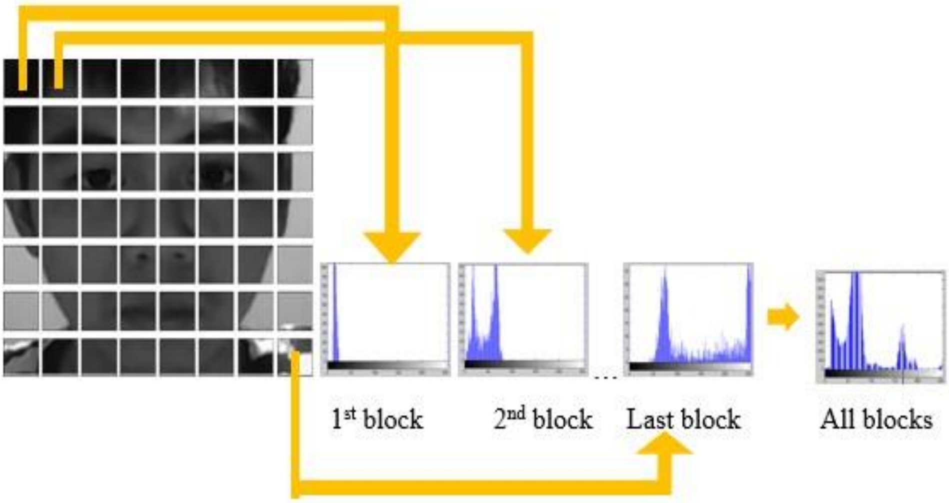


Figure 3.13 Histogram of Image Blocks

3.4.3 Working Principle of PCA

In this proposed approach, PCA face recognition is studied, as it is one of the popular face recognition methods that was suggested and used by the previous researchers. The accuracy of PCA is computed in order to compare with the enhanced LBP.

PCA includes a few steps which will briefly be described in the following paragraphs. For PCA, the image scale, length (M) and height (M) is not so important. This is because PCA is mostly dealing with number of total images, N instead of M. However, same size of test image and training image is a must for PCA computation. Same length and height of the image is assumed in the following equation for illustration. Given a training set of N images with size $M \times M$, the first step of PCA is to convert two dimensional vectors to one dimensional vector. The one dimensional vector can be either column vector or row vector. In this approach, the column vector conversion is done. For each facial image with matrix notation $M \times M$ will be converted to column vector Γi, with dimension $M^2 \times 1$.There are N facial images, each face is represented by column vector Γ1, Γ2, Γ3, .., ΓN. Feature vector of each face is stored in this column vector. The dimension reduced face matrix is constructed by concatenating every single column vector.

PCA is briefly explained by using the equation in the following steps. Step1: Prepare the data,

(3.4)

Γ1 Γ2 Γ3 ΓN Γ1 Γ2 ΓN Dimension reduced

matrix

$$\begin{bmatrix} a11 \\ a12 \\ a13 \\ \vdots \\ aM^2 \end{bmatrix} \begin{bmatrix} a11 \\ a12 \\ a13 \\ \vdots \\ aM^2 \end{bmatrix} \begin{bmatrix} a11 \\ a12 \\ a13 \\ \vdots \\ aM^2 \end{bmatrix} \cdots \begin{bmatrix} a11 \\ a12 \\ a13 \\ \vdots \\ aM^2 \end{bmatrix} = \begin{bmatrix} a11 & a12 & \dots & a1N \\ a21 & a22 & \dots & a2N \\ a31 & a32 & \dots & a3N \\ \vdots & \vdots & \vdots & \vdots \\ aM^2 1 & aM^2 2 & \dots & aM^2 N \end{bmatrix}$$

Step 2: Obtain the mean/average face vector

Next, the average face vector which is also known as mean face is calculated. The mean is computed row by row between the column vectors. The equation of mean face is shown below.

(3.5)

$$\varphi = \frac{1}{N}\sum_{i=1}^{N} \Gamma_i$$

Mean face, φ (3.6)

$$= \begin{bmatrix} \frac{a11 + a12 + \cdots + a1N}{N} \\ \frac{a21 + a22 + \cdots + a2N}{N} \\ \vdots \end{bmatrix}$$

Step 3: Subtract the mean/average face vector

In order to ensure the image data is centred at the origin, the mean face is subtracted from each column vector.

(3.7)

$$\Phi_i = \Gamma_i - \varphi \qquad i = 1,2,\dots,N$$

Dimension reduced matrix Mean face, φ (3.8)

$$\begin{bmatrix} a11 & a12 & \dots & a1N \\ a21 & a22 & \dots & a2N \\ a31 & a23 & \dots & a3N \\ \vdots & \vdots & \vdots & \vdots \\ aM^21 & aM^22 & \dots & aM^2N \end{bmatrix} - \begin{bmatrix} \dfrac{a11 + a12 + \cdots + a1N}{N} \\ \dfrac{a21 + a22 + \cdots + a2N}{N} \\ \vdots \end{bmatrix}$$

matrix A , Φ

$$= \begin{bmatrix} b11 & b12 & \dots & b1N \\ b21 & b22 & \dots & b2N \\ b31 & b23 & \dots & b3N \\ \vdots & \vdots & \vdots & \vdots \\ bM^21 & bM^22 & \dots & bM^2N \end{bmatrix}$$

Step 4: Calculate the covariance matrix

(3.9)

$$C = \frac{1}{N}\sum_{i=1}^{N} \Phi_i \Phi_i^T = AA^T \ , (M^2 x M^2)$$

$$A = [\Phi_1 \, \Phi_2 \, \cdots \Phi_N \,] \ , (M^2 x N)$$

where A is the matrix constructed from the concatenation of the column vectors after remove the mean face.

The purpose of covariance matrix to be constructed is to compute the eigenvectors and eigenvalues. However, AA^T have dimension $M^2 \times M^2$ which is extremely large to be calculated. AA^T ,and $A^T A$ have the same eigenvalues, λ and their eigenvectors can be related as $u_i = Av_i$.Hence $A^T A$ which have dimension $N^2 x N^2$ is calculated instead of AA^T because$N^2 \ll M^2$, less computational time is required.

Step 5: Calculate the eigenvectors and eigenvalues from the covariance matrix.

(3.10)

$$u_i = Av_i \qquad \text{i =1,2,…,N-1}$$

$u_i$is the eigenvector of AA^T whereas v_i is eigenvector of $A^T A$.Eigenvalues of $A^T A$, are calculated and sorted. Eigenvalues less than 1 are eliminated so the number of non-zero eigenvectors may be less than (N-1). (Kalyan Sourav

Dash, 2014).The eigenvectors of AA^T,$U = [u_i \quad \cdots \quad u_{N-1}]$ is also known as Eigen face. Eigen face is the principle component distribution of facial image.

Step 6: Projection of facial image to Eigen face.

$$\Omega_i = U^T(\Gamma_i - \varphi) \qquad \text{i =1,2,…,N-1} \tag{3.11}$$

The facial image is projected on the Eigen face by using the equation to obtain the projected image Ω. $\Gamma_i - \varphi$ is the centered vector, which the mean face is removed.

Steps 1 to 6 are used to train the training image set. For test image only step 1,2, 3 and 6 is required. Step 4 and 5 are not required for test image as the Eigen face is needed only to compute once while training. The Euclidean distance is then used as distance classifier to calculate the shortest distance between the projected image and projected test image for recognition.

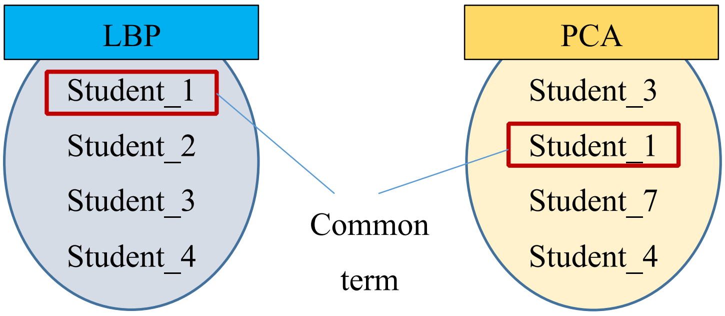
3.4.4 Feature Classification

Chi-square statistic is used as a dissimilarity measure for LBP to determine the shortest distance between training image and the testing image. On the other hand, Euclidean distance is used to compute the shortest distance between trained and test image after PCA feature extraction. Both classifiers, Chi-square statistic and Euclidean distance determine the closest or nearest possible training image to the testing image for face recognition. However, the nearest result might not be always true. Therefore, an algorithm to combine enhanced LBP and PCA is applied in order to increase the accuracy of the system.

3.4.5 Subjective Selection Algorithm and Face Recognition

The feature classification that has been performed in previous part gives the closest result but not absolute. In order to increase the accuracy and suppress the false recognition rate, an algorithm to combine enhanced LBP and PCA is designed in this proposed approach.

In this proposed approach, best five results are obtained from enhanced LBP and PCA. This means that five individuals which have closest distance with respect to input image will be identified. LBP and PCA are two different algorithms which have a different working principle. Hence, LBP and PCA will not have exactly the same five individuals identified. In order to ensure the system capability to suppress the false recognition, one is only classified as recognized if and only if he or she is the first common individual that is identified by both LBP and PCA. From chapter 2, LBP shows higher accuracy compared to PCA. Thus, LBP is designed to have higher priority compared to PCA. This is shown in the Figure 3.14, Student_1 is recognized instead of Student_3 because LBP is prioritized. As a result, the first common individual is selected from PCA with respect to LBP and classified as recognized. If there is no common term between LBP and PCA then the system will not recognize any subject. This subjective selection algorithm is designed to be automated in the system.



The input image will be recognized as Student_1.

Figure 3.14 Subjective Selection Algorithm

CHAPTER 4

RESULT AND DISCUSSION

4.1 Result

In this proposed approach, face recognition student attendance system with userfriendly interface is designed by using GUI(Graphic User Interface). A few buttons are designed in the interface, each provides specific function, for example, start button is to initialize the camera and to perform face recognition automatically according to the face detected, register button allows enrolment or registrations of students and update button is to train the latest images that have been registered in the database. Lastly, browse button and recognize button is to browse facial images from selected database and recognized the selected image to test the functionality of the system respectively.

The tkinter package ("Tk interface") is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.

Architecture

Tcl/Tk is not a single library but rather consists of a few distinct modules, each with separate functionality and its own official documentation. Python's binary releases also ship an add-on module together with it.

Tcl

Tcl is a dynamic interpreted programming language, just like Python. Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit. The Tcl library has a C interface to create and manage one or more instances of a Tcl interpreter, run Tcl commands and scripts in those instances, and add custom commands implemented in either Tcl or C. Each interpreter has an event queue, and there are facilities to send events to it and process them. Unlike Python, Tcl's execution model is designed around cooperative multitasking, and Tkinter bridges this difference (see Threading model for details).

Tk

Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets. Each Tk object embeds its own Tcl interpreter instance with

Tk loaded into it. Tk's widgets are very customizable, though at the cost of a dated appearance. Tk uses Tcl's event queue to generate and process GUI events.

Ttk

Themed Tk (Ttk) is a newer family of Tk widgets that provide a much better appearance on different platforms than many of the classic Tk widgets. Ttk is distributed as part of Tk, starting with Tk version 8.5. Python bindings are provided in a separate module, tkinter.ttk.

Running python -m tkinter from the command line should open a window demonstrating a simple Tk interface, letting you know that tkinter is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded. See the source code for the _tkinter module for more information about supported versions.

Tkinter is not a thin wrapper, but adds a fair amount of its own logic to make the experience more pythonic. This documentation will concentrate on these additions and changes, and refer to the official Tcl/Tk documentation for details that are unchanged.Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications.

To create a tkinter app:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

```
import tkinter
```

There are two main methods used which the user needs to remember while creating the Python application with GUI.

1. **Tk(screenName=None, baseName=None, className='Tk', useTk=1) :** To create a main window, tkinter offers a method 'Tk(screenName=None, baseName=None, className='Tk', useTk=1)'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:
   m=tkinter.Tk() where m is the name of the main window object
2. **mainloop():** There is a method known by the name mainloop() is used when your application is ready to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.
   m.mainloop()

pack() method:It organizes the widgets in blocks before placing in the parent widget.
grid() method:It organizes the widgets in grid (table-like structure) before placing in the parent widget.
place() method:It organizes the widgets by placing them on specific positions directed by the programmer.
There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

Button:To add a button in your application, this widget is used.
The general syntax is:
w=Button(master, option=value)
master is the parameter used to represent the parent window.
There are number of options which are used to change the format of the Buttons. Number of options can be passed as parameters separated by commas. Some of them are listed below.

```
from tkinter import *
m1 = PanedWindow()
m1.pack(fill = BOTH, expand = 1)
left = Entry(m1, bd = 5)
```

```
m1.add(left)
m2 = PanedWindow(m1, orient = VERTICAL)
m1.add(m2)
top = Scale( m2, orient = HORIZONTAL)
m2.add(top)
mainloop()
```

activebackground: to set the background color when button is under the cursor.
activeforeground: to set the foreground color when button is under the cursor.
bg: to set the normal background color.
command: to call a function.
font: to set the font on the button label.
image: to set the image on the button.
width: to set the width of the button.
height: to set the height of the button.

```
import tkinter as tk
r = tk.Tk()
r.title('Counting Seconds')
button = tk.Button(r, text='Stop', width=25, command=r.destroy)
button.pack()
r.mainloop()
```

Output:

Canvas: It is used to draw pictures and other complex layout like graphics, text and widgets.
The general syntax is:
w = Canvas(master, option=value)
master is the parameter used to represent the parent window.
There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bd: to set the border width in pixels.

bg: to set the normal background color.
cursor: to set the cursor used in the canvas.
highlightcolor: to set the color shown in the focus highlight.
width: to set the width of the widget.
height: to set the height of the widget.

```
from tkinter import *
master = Tk()
w = Canvas(master, width=40, height=60)
w.pack()
canvas_height=20
canvas_width=200
y = int(canvas_height / 2)
w.create_line(0, y, canvas_width, y )
mainloop()
```

Output:

CheckButton: To select any number of options by displaying a number of options to a user as toggle buttons. The general syntax is:

w = CheckButton(master, option=value)

There are number of options which are used to change the format of this widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

Title: To set the title of the widget.
activebackground: to set the background color when widget is under the cursor.
activeforeground: to set the foreground color when widget is under the cursor.
bg: to set the normal background color.
command: to call a function.
font: to set the font on the button label.
image: to set the image on the widget.

```
from tkinter import *
```

```
master = Tk()
var1 = IntVar()
Checkbutton(master, text='male', variable=var1).grid(row=0, sticky=W)
var2 = IntVar()
Checkbutton(master, text='female', variable=var2).grid(row=1, sticky=W)
mainloop()
```

Output:

Entry:It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.

The general syntax is:

w=Entry(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

bd: to set the border width in pixels.
bg: to set the normal background color.
cursor: to set the cursor used.
command: to call a function.
highlightcolor: to set the color shown in the focus highlight.
width: to set the width of the button.
height: to set the height of the button.

```
from tkinter import *
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
```

mainloop()

Output:

Frame: It acts as a container to hold the widgets. It is used for grouping and organizing the widgets. The general syntax is:

w = Frame(master, option=value)

master is the parameter used to represent the parent window.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

highlightcolor: To set the color of the focus highlight when widget has to be focused.

bd: to set the border width in pixels.

bg: to set the normal background color.

cursor: to set the cursor used.

width: to set the width of the widget.

height: to set the height of the widget.

In this part, enhanced LBP with radius two is chosen and used as proposed algorithm. The analysis of choosing the radius size will be further explained in the discussion.

4.2 Discussion

This proposed approach provides a method to perform face recognition for student attendance system, which is based on the texture based features of facial images. Face recognition is the identification of an individual by comparing his/her real-time captured image with stored images in database of that person. Thus, training set has to be chosen based on the latest appearance of an individual other than taking important factor for instance illumination into consideration.

The proposed approach is being trained and tested on different datasets. Yale face database which consists of one hundred and sixty-five images of fifteen individuals with multiple conditions is implemented. However, this database consists of only grayscale images. Hence, our own database with color images which is further categorized into high quality set and the low quality set, as images are different in their quality: some images are blurred while some are clearer. The statistics of each data set have been discussed in the earlier chapter.

Viola-Jones object detection framework is applied in this approach to detect and localize the face given a facial image or provided a video frame. From the detected face, an algorithm that can extract the important features to perform face recognition is designed.

Some pre-processing steps are performed on the input facial image before the features are extracted. Median filtering is used because it is able to preserve the edges of the image while removing the image noises. The facial image will be scaled to a suitable size for standardizing purpose and converted to grayscale image if it is not a grayscale image because CLAHE and LBP operator work on a grayscale image.

One of the factors that are usually a stumbling stone for face recognition performance is uneven lighting condition. Hence, many alternatives have been conducted in this proposed approach in order to reduce the non-uniform lighting condition.Before feature extraction takes place, pre-processing is performed on the cropped face image (ROI) to reduce the illumination problem.

In the previous chapters, Contrast Limited Adaptive Histogram Equalization (CLAHE) is proposed in pre-processing in order to improve the image contrast and reduce the illumination effect. Most of the previous researchers have implemented histogram equalization in their approach. In order to study the difference between the CLAHE and histogram equalization, comparison is made and tabulated in Table 4.2.

For the comparison, our own database and Yale face database are used. From the result tabulated, CLAHE appears to perform better compared to histogram equalization. From the image of our own database, the left hand side of the original image appears to be darker compared to right hand side. However, histogram equalization does not improve the contrast effectively, which causes the image remains darker at left hand side. Unlike histogram equalization, CLAHE appears to improve the contrast more evenly throughout the entire facial image. This could help to reduce uneven illumination. In Yale face database, CLAHE prevents some region appears to be washed out as well as reduce over enhancement of noise. Besides, CLAHE shows a clear edge and contour compared to histogram equalization. In addition, by referring to the histograms, the pixel is widely span over the intensity scale axis 0 to 255 for CLAHE whereas for histogram equalization the pixel span from 0 to only about 200 over the intensity scale axis. Hence, it can be said that the contrast of the image is more evenly improved throughout the image by CLAHE compared to histogram equalization based on the result obtained.

After pre-processing, useful feature is extracted by using enhanced LBP (local Binary pattern).Unlike the original LBP operator, enhanced LBP operator consists of different radius size is proposed as mentioned in previous chapters. This different radius size enhanced LBP operator is less affected by uneven lighting compared to original LBP operator. The extracted feature for different radius is shown and tabulated in Table 4.3.The results show when the radius increased, the images are smoothen.

For evaluation purpose, Yale face database with different condition is used for comparison. The normal facial image of each individual in Yale face database is trained and the facial images with varying condition is input as the test image. The recognition rate with the different radius size of LBP operator is computed and tabulated in Table 4.4.

From the Table 4.4,when the radius size increase,only facial images with conditions right light, left light and center light are affected whereas for the other conditions the recognition rate remains constant.This shows that by increasing the radius, uneven lighting effect can be reduced without distorting the detail of the image. From Figure 4.6, the line graph shows that the accuracy of different light conditions increase when radius increases. In addition,it shows that among the different lighting conditions,the system work the best in left light condition followed by center light condition and the last is right light condition.

The recognition rate of LBP operator with different radius is then computed by using our own database. However, LBP operator with different radius does not give significant results because there is no critical illumination problem exists in the images of our own database. Hence,the pixels of good quality images of our own database are modified to generate the illumination effects in order to determine the impact of different size LBP operator. Figure 4.7 shows conditions I, II, III and IV which illustrate different illumination effects.

By increasing the radius size, the detail information is simplified and the contour or shape of the face is emphasized. This illustrates that some of the useless or redundant information is removed and more emphasis is on the critical details for recognition.

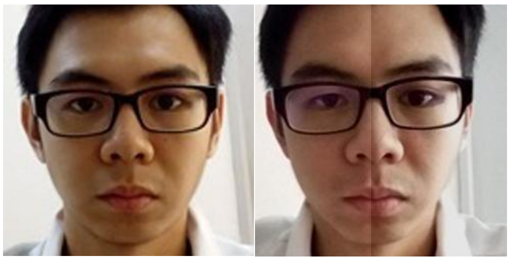
As it is proven any increasing radius LBP performs better compared to original by reducing illumination effect, consistency of the system is also emphasized other than accuracy of the system. From the Table 4.5, although radius three and radius four have higher average accuracy compared to radius two, radius two is more consistent toward different conditions. As the conditions I, II, III and IV is self-simulated conditions, in real time face recognition, the illumination condition is unpredictable. Hence, radius two gives consistent result which is (94.12 %) in condition I, condition III and condition IV is chosen and used as proposed algorithm.

The fact that, the radius might not be the larger the better because larger radius with respect to larger filter size emphasizes complementary information to small scale detail but at the same time it loss discriminative information. The discriminative information is important, for instance to recognize students with glasses free condition.

Figure 4.6 Images of Students With or Without Wearing Glasses

However, it does prove that the enhanced LBP operator with increased radius performs better compared to original LBP in case of illumination effect reduction. Hence, the radius size of the LBP operator has to be wisely selected in order to reduce the illumination effect without sacrificing much of the recognition rate.

From the result, the condition II appears to have lower accuracy compared to others. This is due to the lighting effect of the training image. The training images have its left side relatively darker compared to its right side which is directly opposite of the test image (condition II).



Training Image Test Image (Condition II) Figure 4.7
Training Image VS Testing Image

From the result of proposed LBP in Table 4.6, database with good quality colour images, achieves the highest accuracy (100 %) either one image or two images per individual is trained whereas database with poor quality color images have average accuracy of (86.54 %) when only one image per individual is trained and average accuracy of (88.46 %) when two images per individual are trained. It can be said that the approach works best with good quality images, poor quality images could degrade the performance of the algorithm. Poor quality images were captured by using Laptop camera. The poor quality images might include the relatively darker images, blur images or having too much unwanted noise. In blurred images, the face is blurred out. Unwanted noise can be reduced by applying median filtering, but for those blurred images there are no suitable ways to get rid of it.

4.3 Comparison of LBP and PCA

In this proposed approach, PCA face recognition is performed in order to identify the differences with respect to LBP by using the same database.From the result obtained in Table 4.7, supposedly PCA should have worked better with high quality images which is similar to enhanced LBP.However,it gives slightly lower accuracy in recognition in high quality images compared to low quality images.This is due to different size of the database are used in the proposed approach.For high quality images there are only seventeen students in the database, whereas low quality images involve twenty-six students, which is almost ten students more than high quality images.It is the PCA's nature to be more affected by the size of the database compared to LBP. Hence,the larger the size of the database which means the more students include in the database,the lower the recognition rate of PCA.

Furthermore,the enhanced LBP is compared with the PCA face recognition,by using the same pre-processing procedure and same image enhancement technique. From the Table 4.6 and Table 4.7, the average accuracy of PCA is lower compared to the LBP in all the databases, our own database with high and low quality images and also Yale face database is used respectively.Hence, it can be said that enhanced LBP works better compared to the PCA face recognition given the same dataset is used for training and testing.

An automated subjective selection algorithm involve both enhanced LBP and PCA is designed for face recognition.The best results from enhanced LBP and PCA correspondingly are compared to obtain a common result.This common result will be classified as recognized individual. By doing so, the system becomes more reliable,stable and consistent not only in different expression but also in different lighting condition.This is because two algorithms are used for generalization,one act as a reference to another one.Especially in the camera initializing stage,if the camera is started faster than the lighting source, a darker image will be captured. However the dark image is meaningless to be recognized. The combination of enhanced LBP and PCA able to block the meaningless image from being recognized. Overall accuracy with and without combination of LBP and PCA are tabulated in Table 4.8. It shows that with high quality images,make no difference with or without the algorithm.

However, for low quality images,it shows significant improvement in the accuracy with the algorithm.

4.4 Comparison with Previous Researches

Table 4.1 Summary of Comparison with Previous Researches

Paper/difference	Automated Class Attendance System based on face recognition using PCA Algorithm(D. Nithya, 2015)	Proposed algorithm	Automated Attendance Management System Based On Face Recognition Algorithms(Shireesha Chintalapati, M.V. Raghunadh ,2013)
Noise removal	None	Median filtering	None
Image enhancement	None	Contrast Limited Adaptive Histogram Equalization	Histogram equalization
Featured based	PCA	Enhanced LBP and PCA	PCA/LDA/LBPH
Database	Own database	Own database and Yale face database	NITW-database

Attendance	Write attendance to Excel file	Subjective selection by enhanced LBP and PCA,and write attendace to Excel file	Write attendance to Excel file

From the Table 4.10, proposed algorithm is compared with face recognition student attendance system proposed by previous researchers. The techniques used by the previous researchers to process the images is compared in this proposed approach.

In terms of image enhancement, the paper published in 2013 used histogram equalization to improve the image contrast, while another paper did not apply any technique to improve the image contrast. In this proposed algorithm, CLAHE is used to improve the image contrast. Histogram equalization, which is often used in x-ray applications, gives bone structure a clearer view. However, histogram equalization will tend to cause over enhancement to some of the regions and cause it to be washed out while other regions are not enhanced properly. Hence, CLAHE is implemented instead of histogram equalization to prevent over enhancement and improve the contrast more evenly throughout the image. The difference between CLAHE and histogram equalization is tabulated in result of the previous part.

The research, published in the year 2015 used PCA for feature extraction. While the paper published in the year 2013 used multiple feature extraction algorithms. These feature extraction algorithms are PCA, LDA and LBPH. In this proposed approach, other than enhanced LBP algorithm, PCA is also computed in order to make comparison and to understand their property and performance respectively. In the paper of year 2013, either one of the feature extraction methods PCA, LDA and LBPH is used each time. In this proposed approach, enhanced LBP and PCA are both used as combination to ensure consistent results.

The previous researcher who published the paper in 2015 used their own databases of images in study. The paper published in year 2013 used an image database of 80 individuals (NITW-database) with 20 images of each person, while the paper in year 2015 did not mention the size of image database used. The proposed algorithm uses multiple image databases, which include Yale face database with different lighting and expression for training and testing. In fact, Yale face database allows the study of performance of the proposed algorithm in uneven lighting and variety of expression condition. However, Yale face database consists of only grayscale images without background, thus our own database with colour images is also used in real time application to perform face recognition.

Face recognition is the process of identification of an individual by choosing the closest distance between test image and train image. Hence, quality of images plays an important role in performance of face recognition. Blurred images caused by movement tend to create the after image which can degrade the performance. Furthermore, the test images captured in extremely bright or dark condition can degrade the performance as well because its show a large variation with the train image provided train image is captured in moderate lighting. All these factors have to be taken into consideration when selecting images for testing and training purpose. It is always better to use more images for training, so that the result obtained provides a better generalization and in consequence provide better performance.

In addition, both papers did not apply technique for removal of image noise. In proposed algorithm, Median filtering is used to filter out noises in the image. If the noises on the images are not removed, the algorithm might recognize the noises as part of the crucial features. These will probably affect the overall performance of the algorithm.

Lastly, both papers writes student attendance to Excel file as post-processing. In the proposed approach, a subjective selection algorithm is designed to obtain a common result from enhanced LBP and PCA. This common result from enhanced LBP and PCA is classified as recognized individual and written to Excel file. This algorithm able to reduce false recognition, especially in camera initializing stage, when the camera light is not ready to function. Hence, the proposed algorithm makes the system to be more reliable by giving the consistent result.

4.5 Comparison with Luxand Face Recognition Application

Table 4.2 Comparison of Proposed Algorithm and Luxand Face Recognition

Algorithm	Individual	Accuracy	Training time	Testing time
Porposed algorithm	5	100 %	30 to 40 seconds	Average 10 seconds
Luxand Face Recognition	5	100 %	Almost instantaneous	Almost instantaneous

Luxand Face Recognition (Luxand.com, 2018) is an app that used to perform real-time face recognition.Luxand Face Recognition demo version was installed in the laptop.This is to compare with the proposed algorithm by using the same camera device. Five individuals in this proposed approach use Luxand Face Recognition and proposed algorithm to recognize their faces to make comparisons.

From Table 4.11, both of the algorithm is able to recognize all the five individuals. The proposed algorithm has to wait to update database,whenever a new individual is registered and added to database. The waiting time is about 30 seconds for each training. On the other hand,Luxand Face Recognition app allows the new individual to click on the face detected in the video frame to add their name for registration. This process lasts about 10 seconds. Hence,Luxand Face Recognition app have faster training time compared to the proposed algorithm.

In addition, Luxand Face Recognition app have almost the instantaneous recognition time. In contrast to Luxand Face Recognition, the proposed algorithm have recognition time,which is about 10 seconds. For the real-time video frame captured facial image which is not able to be recognized within 15 seconds will be classified as unrecognized. However,the five individual mentioned were all able to be recognized within 15 seconds. In short, it can be said that the Luxand Face Recognition app have shorter training time and testing time compared to the proposed algorithm.

4.6 Weakness of the Algorithm

The proposed algorithm can only work with a single face. Multiple faces appear in the same image causes each of them to be small. Small face region gives inaccurate features, this will decreases the performance of the system. Hence,whenever more than a face is detected, the system will not perform the recognition.

The LBP algorithm is highly sensitive to image quality and higly affected by the blurred image. LBP is the texture based descriptor which extracts the local grayscale features by performing feature extraction on a small region throughout the entire image. Hence, test image and train image have to be the same quality and captured by the same device in order to have high accuracy.

The laptop built in webcam is the default device in this proposed approach to capture image. The webcam and lighting source of the laptop have low performance which cause the captured images appear to be darker and blurred. This cause the system only function the best if the test image and train image are both captured at the same place under approximately same illumination.

Besides, false recognition occurs when the facial image is blurred. The blurred image caused by the after image created by movement will degrade the performance. The face feature extracted from the blurred image would be totally different compared to train image resulting in false recognition.

In addition, if an individual wears make up in the image for face recognition, the important features will be covered. Similarly, face region should not be covered by hair, beard or any accessories to ensure better performance. For instance, a girl provides a facial image with her face covered by hair, it causes false recognition to occur if the girl ties her hair. This is because anything covering the face region will be assumed as face feature. This causes a relatively large difference between test image and train image.

Different level of brightness or lighting could be a challenging problem for face recognition. Hence,limitation of the proposed algorithm is studied and analysis is conducted by modifying the pixels of high quality images in order to manipulate the brightness of the facial images. The recognition rate of facial images under different level of brightness is computed and tabulated.

Figure 4.12 shows images with different intensity by adding different constants to pixel. The performance of the proposed algorithm is tabulated in the Table 4.9.

From the Table 4.9, the proposed algorithm function the best when the intensity increase by a constant at the range of 25 and 50.Further increasing or decreasing the intensity level out of this range will cause the recognition rate to drop to (94.12 %) .Hence,it can be said that the system work better in a relatively brighter image then a darker image.

4.7 Problems Faced and Solutions Taken

One of the problems in real-time face recognition is the difficulty to obtain sufficient and suitable images for training and testing purpose. It is hard to obtain in real-time databases with a variety of variables, and it is hard to obtain publicly available databases. Yale face database is one of the databases that could be downloaded by the public. Hence, Yale face database is adopted and used in this proposed approach. However, Yale face database consists of only grayscale images without any background. Hence, our own database consists of colour images which is categorized to high- quality images and low quality-images are also used.

Besides, it is very difficult to obtain an open source or the free face recognition software in order to make comparisons. In this proposed approach, Luxand Face SDK window demo version software is downloaded and implemented in the laptop. By using laptop built in webcam to recognize faces, the proposed algorithm and Luxand Face SDK demo able to be compared.

From the Luxand Face recognition website (Luxand.com, 2018), they explained that the Face SDK is a high performance, multi-platform face recognition, identification and facial feature detection solution. For Luxand Face Recognition software, the self-learning Al enables video-based identification and the enrolment can be done at any time as simple as putting a name tag in a video, the system will identify that subject in all past, present and future videos. As a video-based identification software, it is believed to work better than key-frame based identification. Nevertheless, the detailed information of its working principle is unable to be obtained from their sites.

Viola-Jones algorithm can cause false face detection. This can be solved by increasing the detection threshold (Mathworks.com, 2018). The threshold indicates the number of detections needed to declare a final detection around an object. By using MATLAB built in function, MergeThreshold, the detection threshold can be adjusted to reduce the false face detection.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

In this approach, a face recognition based automated student attendance system is thoroughly described. The proposed approach provides a method to identify the individuals by comparing their input image obtained from recording video frame with respect to train image. This proposed approach able to detect and localize face from an input facial image, which is obtained from the recording video frame. Besides, it provides a method in pre-processing stage to enhance the image contrast and reduce the illumination effect. Extraction of features from the facial image is performed by applying both LBP and PCA. The algorithm designed to combine LBP and PCA able to stabilize the system by giving consistent results. The accuracy of this proposed approach is 100 % for high-quality images, 92.31 % for low-quality images and 95.76 % of Yale face database when two images per person are trained.

As a conclusion for analysis, the extraction of facial feature could be challenging especially in different lighting. In pre-processing stage, Contrast Limited Adaptive Histogram Equalization (CLAHE) able to reduce the illumination effect. CLAHE perform better compared to histogram equalization in terms of contrast improvement. Enhanced LBP with larger radius size specifically, radius size two, perform better compared to original LBP operator, with less affected by illumination and more consistent compared to other radius sizes.

5.2 Recommendation

In this proposed approach,there are a few limitations. First,the input image has to be frontal and a upright single facial image. Second,the accuracy might drop under extreme illumination problem. Third,false recognition might occur if the captured image is blurred. Besides, LBP is textural based descriptor which extracts local features. Hence, test image and train image have to be the same quality which is captured by using the same device in order to have high accuracy. Lastly, if an individual wears make up in the image for face recognition, the important features will be covered.

In fact, a better camera with a better lighting source able to reduce the illumination problem and also able to avoid the captured of blurred images. In this proposed approach, laptop built in camera is a default device. However the lighting source of the laptop camera is very dim, this cause the system to be unstable. For future work, a better camera and a better lighting source can be used in order to obtain better result. This can reduce the dependency on the brightness of environment, especially the places to capture test and train images. Furthermore, a face recognition system which has more faces other than a single facial image can be designed. This can increase the efficiency of the system. The test image and train image in this approach is highly related to each other and highly dependent on the image captured device. The capture device has to be the same for this approach to perform better. Thus, other algorithms can be used instead of LBP, for example A.I (artificial intelligence) algorithm which can be implemented to perform the face recognition. CNN (Convolution Neural Network) which is a hot topic recently, is a machine deep learning algorithm which is able to perform recognition with less dependency on a particular train image given a large database. However, CNN requires an extremely large database to increase its accuracy or having relatively small class size to have high performance.

In pre-processing stage, an algorithm, for instance affine transform can be applied to align the facial image based on coordinates in the middle of the eyes. This might help, especially in PCA algorithm, which it maps test image to train image to perform face recognition.

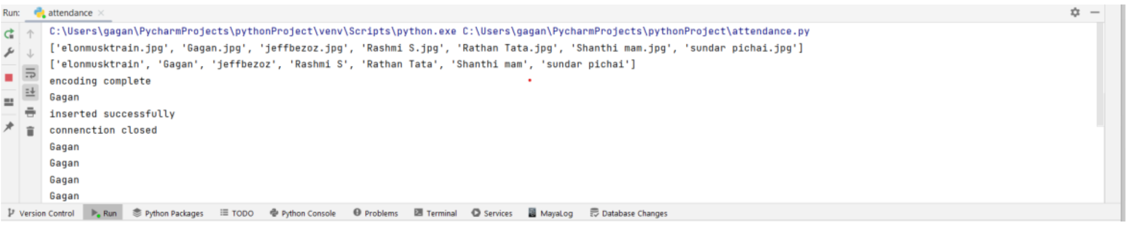
CHAPTER 6
CODING

```
import tkinter as tk
from tkinter import filedialog, simpledialog, messagebox
import shutil
import os
from tkinter import ttk
import pypyodbc as odbc


class LoadingDialog(tk.Toplevel):
    def __init__(self, parent):
        tk.Toplevel.__init__(self, parent)
        self.title("Loading")
        self.resizable(False, False)

        # Center the dialog on the screen
        self.withdraw()
        self.update_idletasks()
        self.geometry(f"+{int(self.winfo_screenwidth() / 2 -
self.winfo_width() / 2)}+{int(self.winfo_screenheight() / 2 -
self.winfo_height() / 2)}")
        self.deiconify()

        # Create a label for the loading message
        self.loading_label = tk.Label(self, text="Real-time Face
Recognition and Attendance")
        self.loading_label.pack(pady=10)
        self.loading_label = tk.Label(self, text="Loading...")
        self.loading_label.pack(pady=10)

        # Create a progress bar
        self.progress_bar = ttk.Progressbar(self, length=300,
mode='indeterminate')
        self.progress_bar.pack(pady=10)

        # Start the progress animation
        self.progress_bar.start()

        # Simulate loading by adding a delay
        self.after(4000, self.open_main_application)

    def open_main_application(self):
        self.progress_bar.stop()  # Stop the progress animation
        self.destroy()  # Close the loading dialog

        # Create a tkinter window and run the application in full-
screen mode
        root = tk.Tk()
        root.attributes('-fullscreen', True)
        root.title("AI Face Attendance System")

        # Prompt for login
        login(root)

        # Run the Tkinter event loop
        root.mainloop()


class FileUploader:
    def __init__(self, master):
        self.master = master
        master.attributes('-fullscreen', True)
        master.title("FACE ATTENDANCE SYSTEM")
        # label for heading
        username_label = tk.Label(master, text="AI FACE RECOGNITION
AND ATTENDANCE SYSTEM")
        username_label.pack()
        # Create a label for the file path
        self.file_path_label = tk.Label(master, text="")
        self.file_path_label.pack(pady=10)

        # Create a button to select a file
        self.select_file_button = tk.Button(master, text="Select
File", command=self.select_file)
        self.select_file_button.pack(pady=10)

        # Create a button to upload the file
        self.upload_button = tk.Button(master, text="Upload",
command=self.upload_file)
        self.upload_button.pack(pady=10)

        # Create an exit button
        self.exit_button = tk.Button(master, text="Exit",
command=master.destroy)
        self.exit_button.pack(pady=10)

    def select_file(self):
        # Open a file dialog to select a file
        file_path = filedialog.askopenfilename()

        # Update the file path label with the selected file path
        self.file_path_label.config(text=file_path)

    def upload_file(self):
        # Get the selected file path
        file_path = self.file_path_label.cget("text")

        # Get the filename and extension
        filename, extension = os.path.splitext(file_path)

        # Open a dialog to prompt the user to enter a new name for the
file
        new_filename = simpledialog.askstring("Rename File", "Enter a
new name for the file:")

        # Set the destination directory
        dest_dir =
'C:\\Users\gagan\PycharmProjects\pythonProject\Images'

        if new_filename is not None and new_filename.strip():
            # Copy the file to the destination directory with the new
name
            shutil.copy(file_path, os.path.join(dest_dir, new_filename
+ extension))

            # Show a message box to indicate that the upload was
successful
            messagebox.showinfo("Upload Successful", "The file has
been uploaded successfully.")
        else:
            messagebox.showerror("Error", "Please enter a valid new
name for the file.")
```

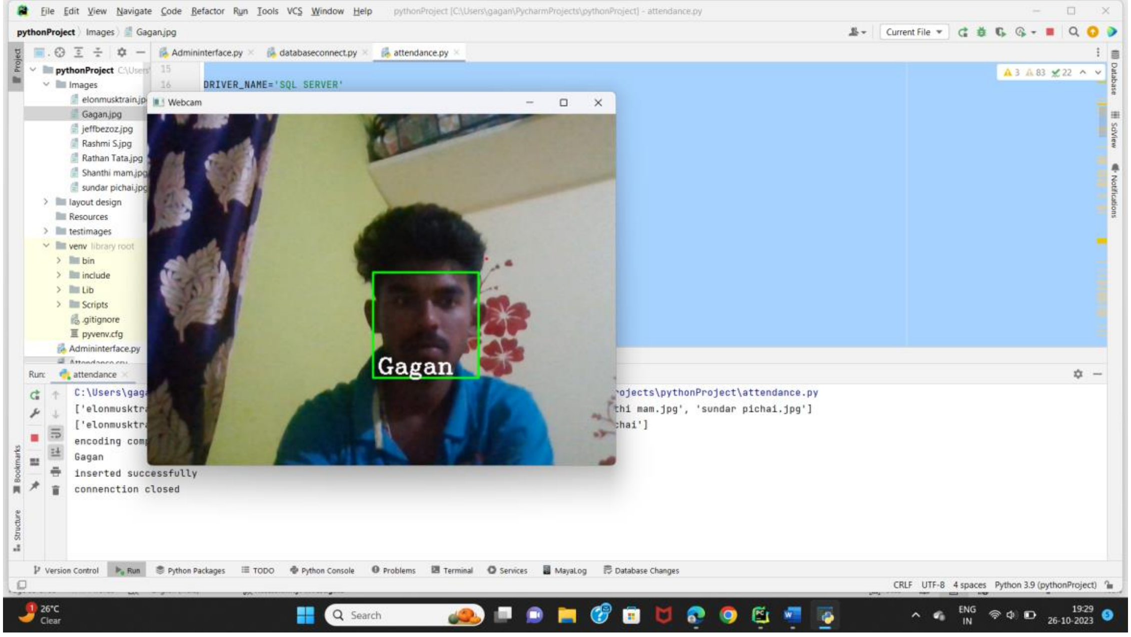
```
def open_database_table(root):
    try:
        # Establish database connection
        connection_string = """
        DRIVER={SQL SERVER};
        SERVER=GAGAN\SQLEXPRESS;
        DATABASE=gagan;
        Trust_connection=yes;
        """
        con = odbc.connect(connection_string)
        cursor = con.cursor()

        # Fetch data from the database
        query = "SELECT * FROM abc"
        cursor.execute(query)
        data = cursor.fetchall()

        # Clear the existing widgets in the root window
        for widget in root.winfo_children():
            widget.destroy()
        username_label = tk.Label(root, text="AI FACE RECOGNITION AND 
ATTENDANCE SYSTEM")
        username_label.pack()
        # Create a Treeview widget for the database table
        tree = ttk.Treeview(root)
        tree["columns"] = ("SerialNumber", "Name", "Time", "Date")
        tree.heading("SerialNumber", text="Serial Number")
        tree.heading("Name", text="Name")
        tree.heading("Time", text="Time")
        tree.heading("Date", text="Date")
        tree.column("#0", width=0)  # Hide the first column
        tree.column("SerialNumber", width=100, anchor="center")
        tree.column("Name", width=150, anchor="center")
        tree.column("Time", width=100, anchor="center")
        tree.column("Date", width=100, anchor="center")
        tree.tag_configure("evenrow", background="#F0F0F0")
        tree.tag_configure("oddrow", background="#D9D9D9")
        tree.pack(expand=True, fill=tk.BOTH)

        # Add data to the table
        for index, record in enumerate(data, start=1):
            serial_number = index
            if index % 2 == 0:
                tree.insert("", tk.END, values=(serial_number,) + 
record, tags=("evenrow",))
            else:
                tree.insert("", tk.END, values=(serial_number,) + 
record, tags=("oddrow",))

        # Create a refresh button to reload the table data
        refresh_button = tk.Button(root, text="Refresh Database", 
command=lambda: open_database_table(root))
        refresh_button.pack(pady=10)

        # Create a button to go back to the main application
        back_button = tk.Button(root, text="Back to Main Application", 
command=lambda: initialize_main_application(root))
        back_button.pack(pady=10)
    except Exception as e:
        messagebox.showerror("Error", f"An error occurred while 
opening the database table: {e}")


def login(root):
    def validate_login():
        username = username_entry.get()
        password = password_entry.get()

        # Validate the username and password
        if username == "gagan" and password == "gagansr3":
            # Clear the login form
            login_frame.destroy()

            # Initialize the main application
            initialize_main_application(root)
        else:
            messagebox.showerror("Login Failed", "Invalid username or 
password. Please try again.")

    def exit_application():
        root.destroy()

    username_label = tk.Label(root, text="AI FACE RECOGNITION AND 
ATTENDANCE SYSTEM")
    username_label.pack()
    # Create a frame for the login form
    login_frame = tk.Frame(root)
    login_frame.pack(pady=50)

    # Create labels and entries for username and password
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame, text="ADMIN LOGIN")
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_label = tk.Label(login_frame)#for spacing
    username_label.pack()
    username_label = tk.Label(login_frame, text="Username:")
    username_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    username_entry = tk.Entry(login_frame)
    username_entry.pack()

    username_label = tk.Label(login_frame) # for spacing
    username_label.pack()
    password_label = tk.Label(login_frame, text="Password:")
    password_label.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()
    password_entry = tk.Entry(login_frame, show="+")
    password_entry.pack()
    username_label = tk.Label(login_frame)  # for spacing
    username_label.pack()

    # Create a login button
    login_button = tk.Button(login_frame, text="Login",
command=validate_login)
    login_button.pack(pady=10)

    # Create an exit button
    exit_button = tk.Button(login_frame, text="Exit",
command=exit_application)
    exit_button.pack(pady=10)


def initialize_main_application(root):
    # Clear the existing widgets in the root window
    for widget in root.winfo_children():
        widget.destroy()

    # Create a FileUploader instance
    file_uploader = FileUploader(root)

    # Create a button to open the database table
    database_table_button = tk.Button(root, text="Open Database
Table", command=lambda: open_database_table(root))
    database_table_button.pack(pady=10)


# Create the main application window
root = tk.Tk()
root.withdraw()  # Hide the main window initially

# Create the loading dialog
loading_dialog = LoadingDialog(root)

# Run the Tkinter event loop
root.mainloop()
import csv
import sys
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
import pyttsx3
import pypyodbc as odbc

temp=1

abc=[100]
import databaseconnect

DRIVER_NAME='SQL SERVER'
SERVER_NAME='GAGAN\SQLEXPRESS'
DATABASE_NAME='gagan'
```

```
connection_string=f"""
DRIVER={{{DRIVER_NAME}}};
SERVER={SERVER_NAME};
DATABASE={DATABASE_NAME};
Trust_connection=yes;
"""

#speech language setting
engine=pyttsx3.init()
#directory path

path='Images'
images = []
classNames = []
Mylist = os.listdir(path)
print(Mylist)

for cl in Mylist:
    curimg=cv2.imread(f'{path}/{cl}')
    images.append(curimg)
    classNames.append(os.path.splitext(cl)[0])
print(classNames)

def findencodings(images):
    encodelist=[]
    for img in images:
        img=cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode=face_recognition.face_encodings(img)[0]
        encodelist.append(encode)
    return encodelist

sampletime=datetime.now()
mem=sampletime.strftime('%H:%M:%S')
if mem=='00:00:00':
    abc=[]
def markattendance(name):
        if name not in abc:
            now=datetime.now()
            dtstring=now.strftime('%H:%M:%S')
            date = now.strftime('%d/%m/%y')
            abc.append(name)
            engine.say(name+'Your face is confirmed and attendance is
marked Successfully')
            engine.runAndWait()
            engine.stop()
            #database file open
            time = now.strftime('%H:%M:%S')
            date = now.strftime('%d/%m/%y')

            records = [name, time, date]

            DRIVER_NAME = 'SQL SERVER'
            SERVER_NAME = 'GAGAN\SQLEXPRESS'
            DATABASE_NAME = 'gagan'

            connection_string = f"""
            DRIVER={{{DRIVER_NAME}}};
            SERVER={SERVER_NAME};
            DATABASE={DATABASE_NAME};
            Trust_connection=yes;
            """

            try:
                con = odbc.connect(connection_string)
            except Exception as e:
                print(e)
                print("task is terminated")
                sys.exit()
            else:
                curser = con.cursor()
            insert = """
            INSERT INTO abc
            VALUES (?,?,?)
            """
            try:
                curser.execute(insert, records)
            except Exception as e:
                curser.rollback()
                print(e)
                print("gtransaction rolled back")
            else:
                print("inserted successfully")
                curser.commit()
                var = curser.closed
            finally:
                if con.connected == 1:
                    print("connenction closed")

encodelistforknown=findencodings(images)
print('encoding complete')

cap= cv2.VideoCapture(0)

while True:
    success, img= cap.read()
    imgs = cv2.resize(img,(0,0),None,0.25,0.25)
    imgs = cv2.cvtColor(imgs,cv2.COLOR_BGR2RGB)

    facecurframe= face_recognition.face_locations(imgs)
    encodescurframe=face_recognition.face_encodings(imgs,
facecurframe)

    for encodeFace,faceloc in zip(encodescurframe, facecurframe):
        matches= face_recognition.compare_faces(encodelistforknown,
encodeFace)
        facedis=face_recognition.face_distance(encodelistforknown,
encodeFace)
        #print(facedis)
        matchindex=np.argmin(facedis)
        #print(matchindex)

        if matches[matchindex]:
            name=classNames[matchindex]
            print(name)
            y1,x2,y2,x1= faceloc
            y1, x2, y2, x1 =y1*4, x2*4, y2*4, x1*4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0),2)
            cv2.putText(img,name,(x1+6,y2-
6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
            markattendance(name)
        else:
            engine.say("Your face not confirmed wait to recheck 
again")
            engine.runAndWait()
            engine.stop()
            exec (open("attendance.py").read())


        cv2.imshow('Webcam',img)
        cv2.waitKey(1)

import pypyodbc as odbc
import sys
from datetime import datetime
from attendance import name

nme=name
now=datetime.now()
time=now.strftime('%H:%M:%S')
date=now.strftime('%d/%m/%y')

records=[nme,time,date]

DRIVER_NAME='SQL SERVER'
SERVER_NAME='GAGAN\SQLEXPRESS'
DATABASE_NAME='gagan'

connection_string=f"""
DRIVER={{{DRIVER_NAME}}};
SERVER={SERVER_NAME};
DATABASE={DATABASE_NAME};
Trust_connection=yes;
"""

try:
    con=odbc.connect(connection_string)
except Exception as e:
    print(e)
    print("task is terminated")
    sys.exit()
else:
    curser=con.cursor()
insert="""
INSERT INTO abc
VALUES (?,?,?)
"""
try:
    curser.execute(insert, records)
except Exception as e:
    curser.rollback()
    print(e)
    print("gtransaction rolled back")
else:
    print("inserted successfully")
    curser.commit()
    var = curser.closed
finally:
    if con.connected==1:
        print("connenction closed")
exec(open("attendance.py").read())
sys.exit()
```

CHAPTER 9
SCREENSHOTS

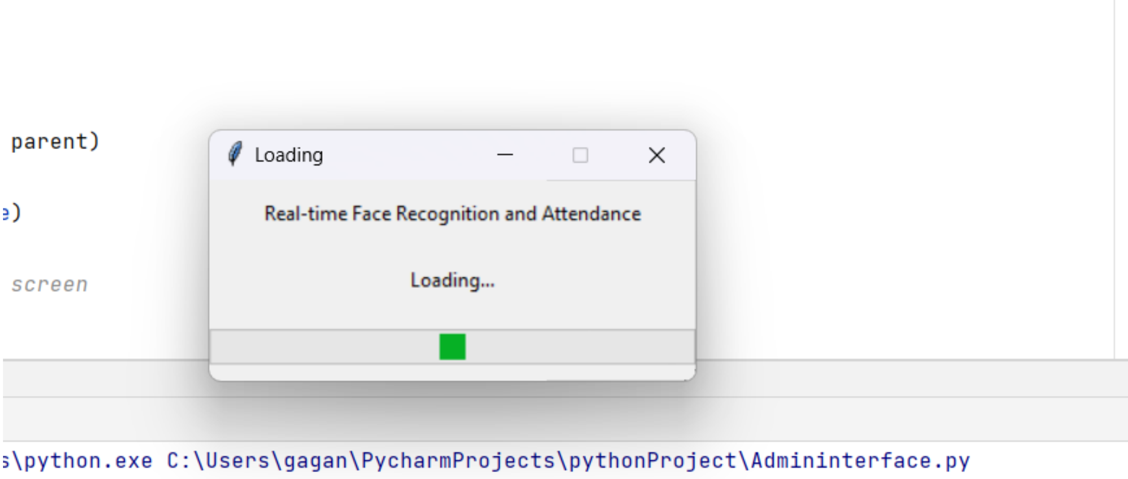


```
C:\Users\gagan\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\gagan\PycharmProjects\pythonProject\attendance.py
['elonmusktrain.jpg', 'Gagan.jpg', 'jeffbezoz.jpg', 'Rashmi S.jpg', 'Rathan Tata.jpg', 'Shanthi mam.jpg', 'sundar pichai.jpg']
['elonmusktrain', 'Gagan', 'jeffbezoz', 'Rashmi S', 'Rathan Tata', 'Shanthi mam', 'sundar pichai']
encoding complete
Gagan
inserted successfully
connection closed
Gagan
Gagan
Gagan
Gagan
```

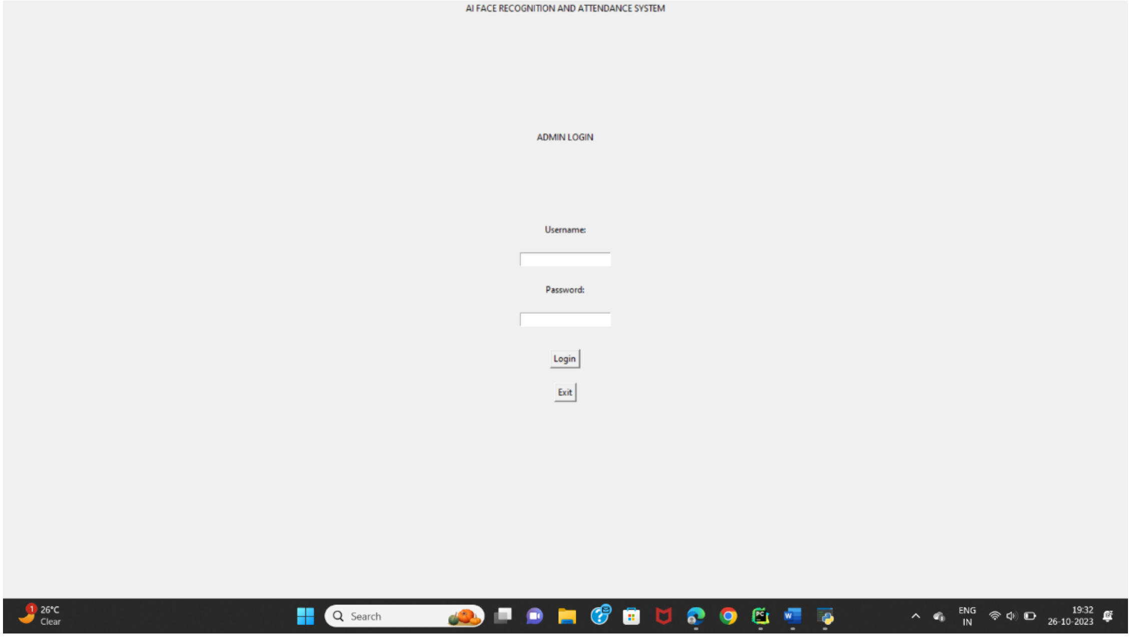


```
DRIVER_NAME='SQL SERVER'
```

```
C:\Users\gaga               ojects\pythonProject\attendance.py
['elonmusktr               thi mam.jpg', 'sundar pichai.jpg']
['elonmusktr               chai']
encoding com
Gagan
inserted successfully
connection closed
```



```
s\python.exe C:\Users\gagan\PycharmProjects\pythonProject\Admininterface.py
```

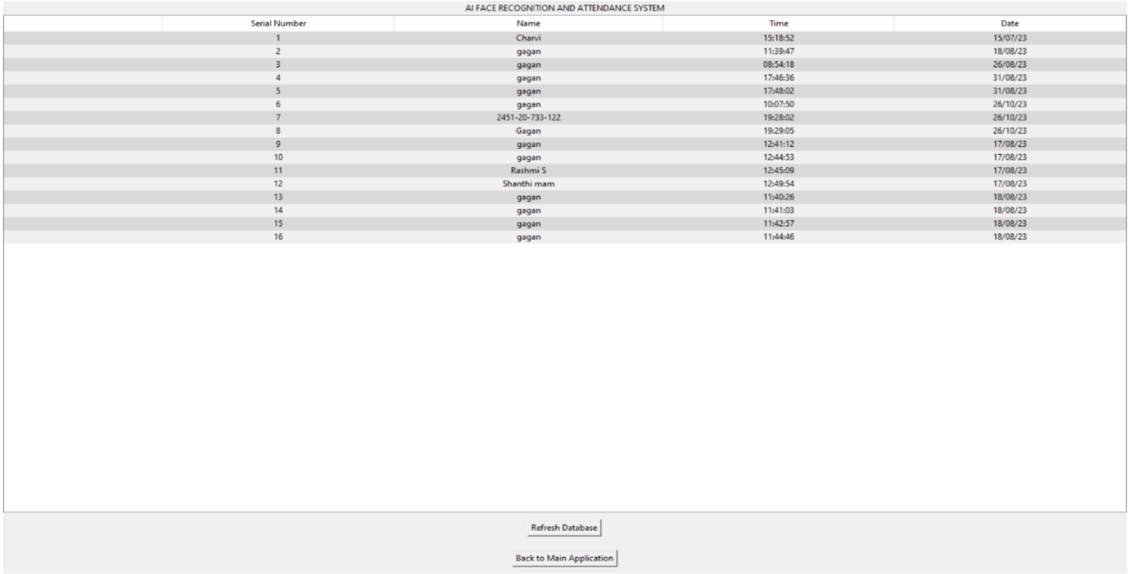


AI FACE RECOGNITION AND ATTENDANCE SYSTEM

ADMIN LOGIN

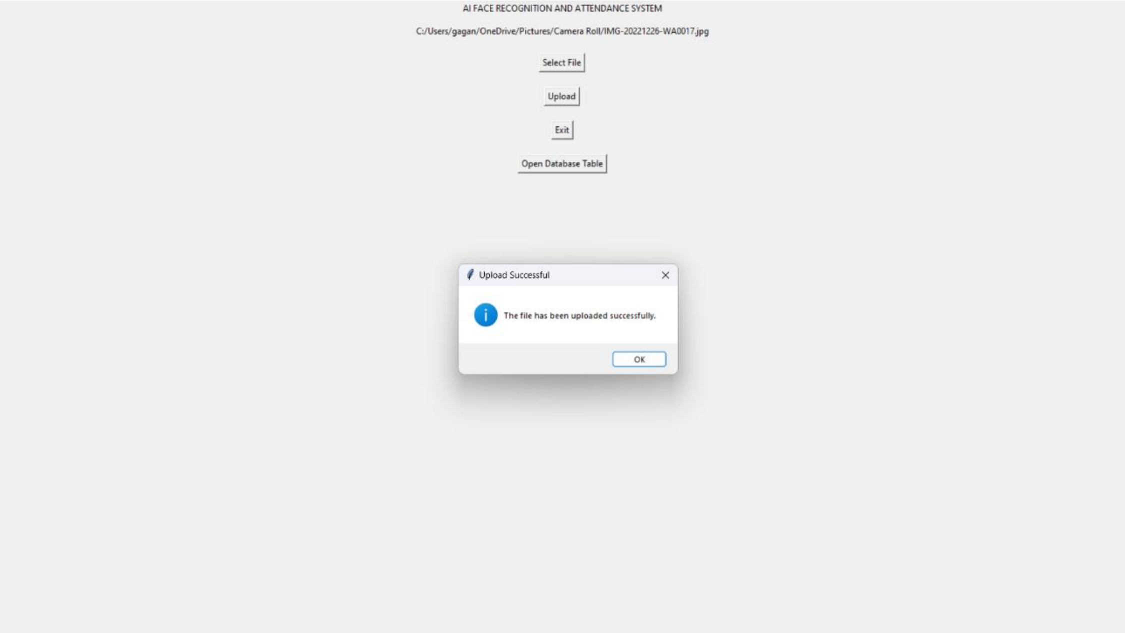
Username:

Password:



AI FACE RECOGNITION AND ATTENDANCE SYSTEM

	Serial Number	Name	Time	Date
	1	Charvi	15:18:52	15/07/23
	2	gagan	11:39:47	18/08/23
	3	gagan	08:54:18	26/08/23
	4	gagan	17:46:36	31/08/23
	5	gagan	17:48:02	31/08/23
	6	gagan	10:07:50	26/10/23
	7	2451-20-733-122	19:28:02	26/10/23
	8	Gagan	19:29:05	26/10/23
	9	gagan	12:41:12	17/08/23
	10	gagan	12:44:53	17/08/23
	11	Rashmi S	12:45:09	17/08/23
	12	Shanthi mam	12:49:54	17/08/23
	13	gagan	11:40:26	18/08/23
	14	gagan	11:41:03	18/08/23
	15	gagan	11:42:57	18/08/23
	16	gagan	11:44:46	18/08/23

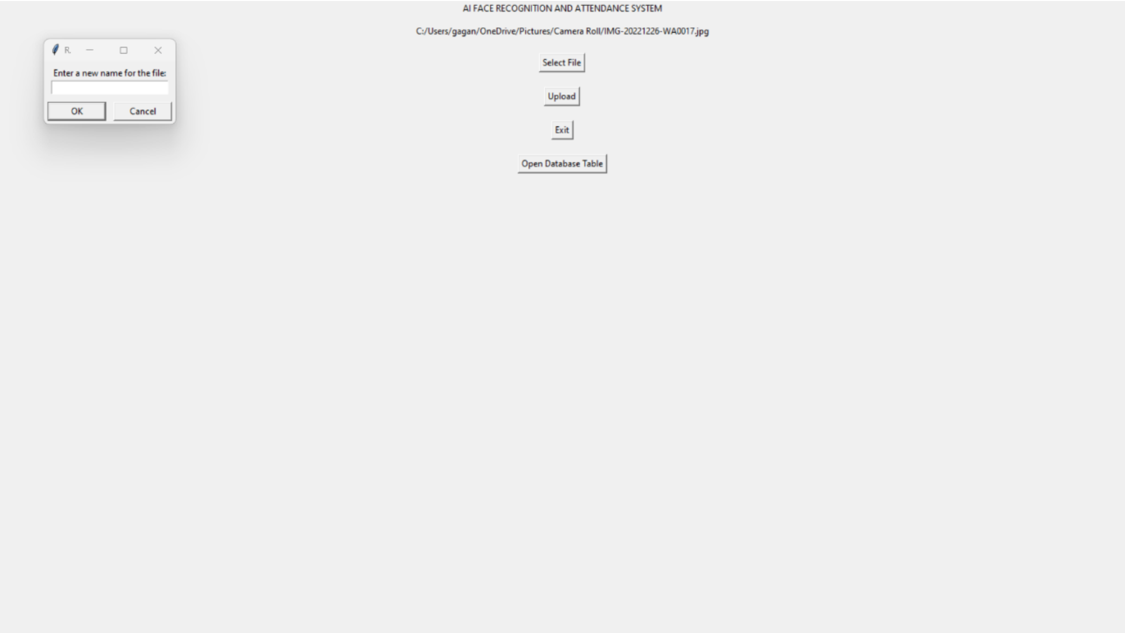


AI FACE RECOGNITION AND ATTENDANCE SYSTEM

C:/Users/gagan/OneDrive/Pictures/Camera Roll/IMG-20221226-WA0017.jpg

Upload Successful

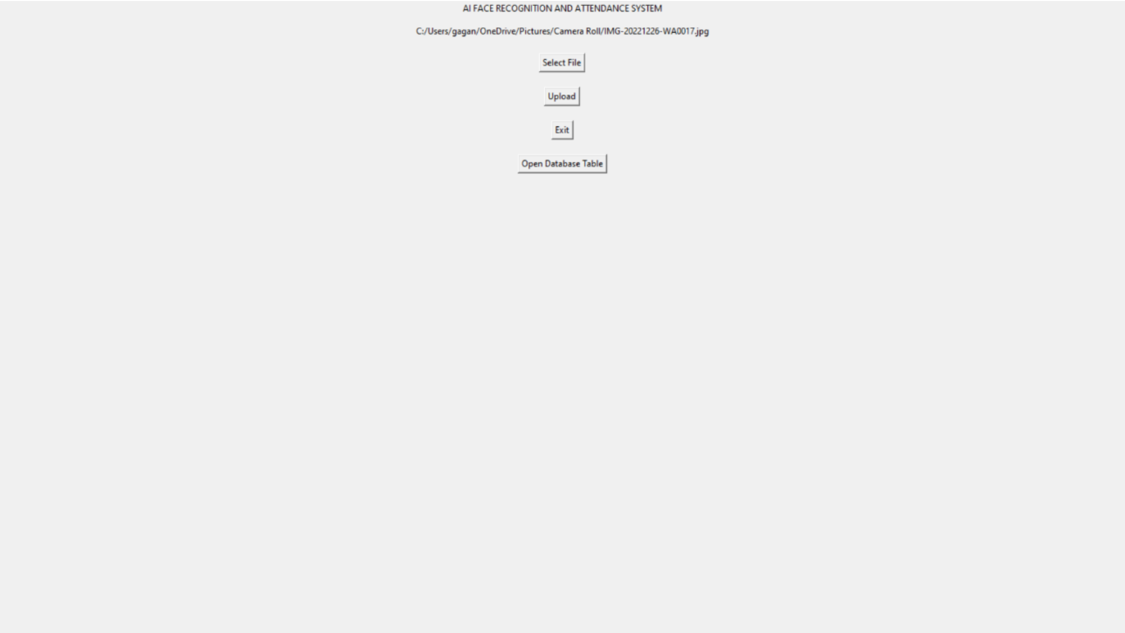
The file has been uploaded successfully.



AI FACE RECOGNITION AND ATTENDANCE SYSTEM

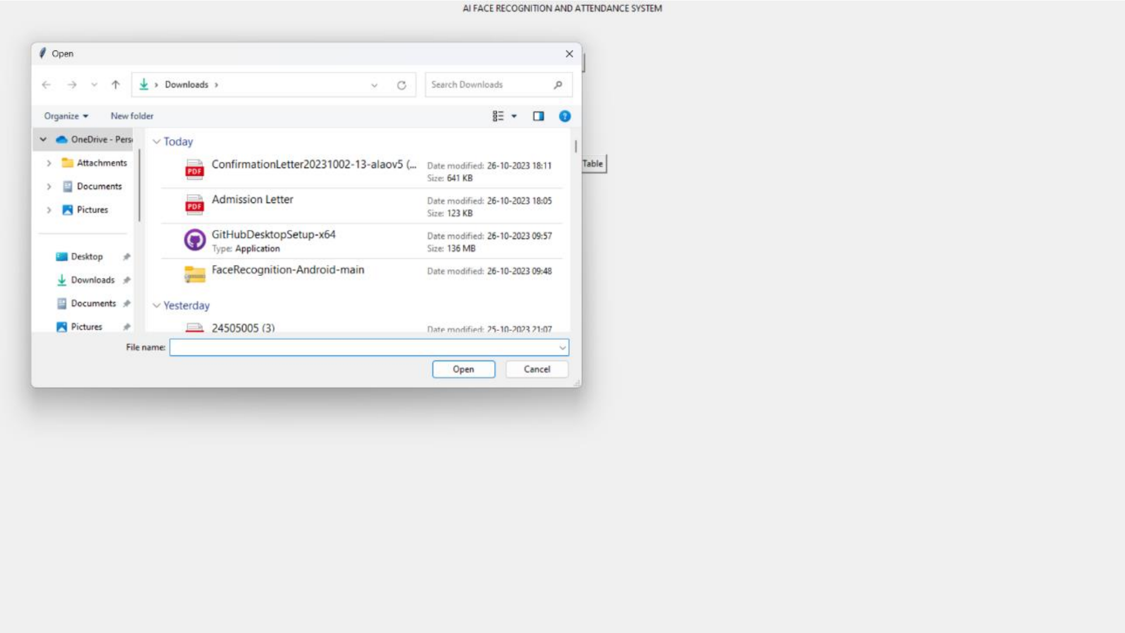
C:/Users/gagan/OneDrive/Pictures/Camera Roll/IMG-20221226-WA0017.jpg

Enter a new name for the file:



AI FACE RECOGNITION AND ATTENDANCE SYSTEM

C:/Users/gagan/OneDrive/Pictures/Camera Roll/IMG-20221226-WA0017.jpg



AI FACE RECOGNITION AND ATTENDANCE SYSTEM

Open

Downloads ›

Today

ConfirmationLetter20231002-13-alaov5 (... Date modified: 26-10-2023 18:11 Size: 641 KB

Admission Letter Date modified: 26-10-2023 18:05 Size: 123 KB

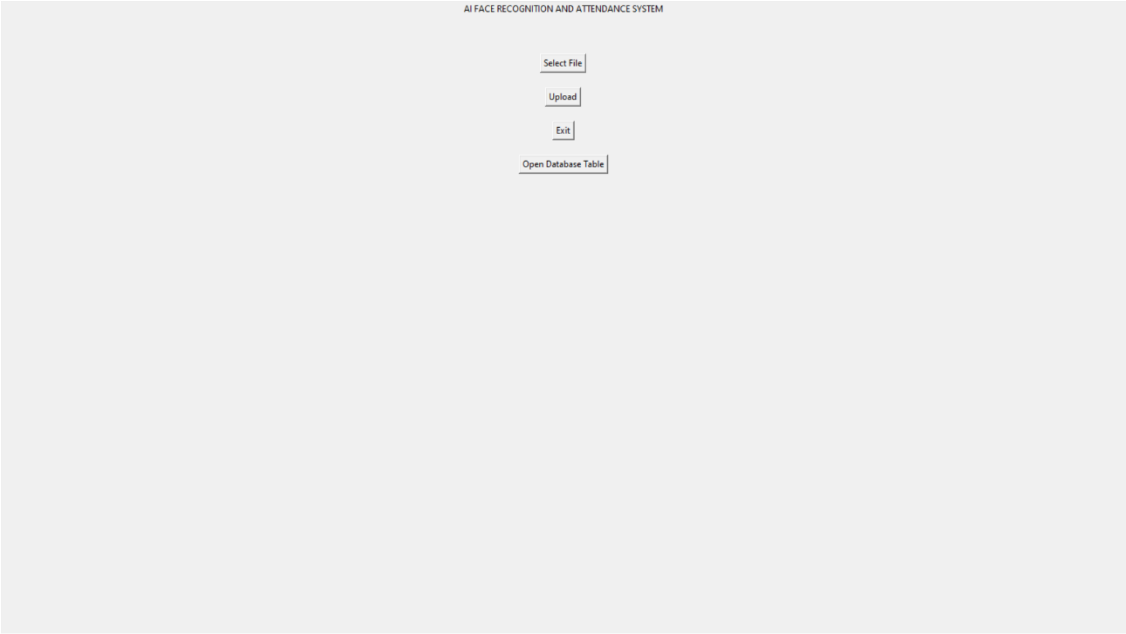
GitHubDesktopSetup-x64 Type: Application Date modified: 26-10-2023 09:57 Size: 136 MB

FaceRecognition-Android-main Date modified: 26-10-2023 09:48

Yesterday

24505005 (3) Date modified: 25-10-2023 21:07

File name:



AI FACE RECOGNITION AND ATTENDANCE SYSTEM

CHAPTER 6
REFERENCES

Ashley DuVal. (2012).*Face Recognition Software -History of Forensic Psychology*. [online] Available at: http://forensicpsych.umwblogs.org/research/criminal-justice/face-recognitionsoftware/ [Accessed 12 Apr. 2018].

Jesse Davis West. (2017). *History of Face Recognition - Facial recognition software*. [online] Available at: https://www.facefirst.com/blog/brief-history-of-facerecognition-software/ [Accessed 25 Mar. 2018].

Reichert, C. (2017). *Intel demos 5G facial-recognition payment technology | ZDNet*. [online] ZDNet. Available at: http://www.zdnet.com/article/intel-demos-5g-facial-recognition-paymenttechnology/ [Accessed 25 Mar. 2018].

Zhao, W., Chellappa, R., Phillips, P. and Rosenfeld, A. (2003). Face recognition. *ACM Computing Surveys*, 35(4), pp.399-458.

Pooja G.R, et al. (2010). An automated Attendance System Using Image Processing. *International Journal Of Advanced Networking & Applications*.

Wagh, P., Thakare, R., Chaudhari, J. and Patil, S. (2015). Attendance system based on face recognition using eigen face and PCA algorithms. *International Conference on Green Computing and Internet of Things*.

Arun Katara, Mr. Sudesh V. Kolhe, Mr. Amar P. Zilpe, Mr. Nikhil D. Bhele, Mr. Chetan J. Bele. (2017). “Attendance System Using Face Recognition and Class Monitoring System”, *International Journal on Recent and Innovation Trends in Computing and Communication*, V5 (2).

S.Aanjanadevi, Dr.V.Palanisamy,R.Anandha Jothi. (2017). A Study on Secure Online Voting System using Biometrics Face Detection and Recognition Algorithms. *International Journal for Modern Trends in Science and Technology,* V3(8).

Wei-Lun Chao. (2007). Face Recognition, GICE, National Taiwan University.

Akshara Jadhav. (2017) .Automated Attendance System Using Face Recognition. *International Research Journal of Engineering and Technology*.V4 (1).

P. Arun Mozhi Devan et al., (2017). Smart Attendance System Using Face Recognition. *Advances in Natural and Applied Sciences*. 11(7), Pages: 139-144

Rahul V. Patil and S. B.Bangar. (2017). Video Surveillance Based Attendance system. *IJARCCE*, 6(3), pp.708-713.

Mrunmayee Shirodkar. (2015). Automated Attendance Management System using Face Recognition. *International Journal of Computer Applications* and *International Conference and Workshop on Emerging Trends in Technology*.

Naveed Khan Balcoh. (2012). Algorithm for Efficient Attendance Management: Face Recognition based approach.*International Journal of Computer Science* Issues, V9 (4), No 1.

Varsha Gupta, Dipesh Sharma. (2014), "A Study of Various Face Detection Methods", *International Journal of Advanced Research in Computer and Communication Engineering)*, vol.3, no. 5.

P. Viola, M. J. Jones. (2004), "Robust Real-Time Face Detection", *International Journal of Computer Vision* 57(2), 137–154.

Mekha Joseph et al. (2016). Children's Transportation Safety System Using Real Time Face Recognition. *International Journal of Advanced Research in Computer and Communication Engineering* V5 (3).

Srushti Girhe et al. (2015). Computer Vision Based Semi-automatic Algorithm for face detection. *International Journal on Recent and Innovation Trends in Computing and Communication* V3(2).

Burak Ozen. (2017).Introduction to Boosting Methodology & Adaboost algorithm. [online] Available at: https://www.linkedin.com/pulse/introduction-boostingmethodology-adaboost-algorithm-burak-ozen [Accessed 12 Apr. 2018].

Chris McCormick. (2013). Adaboost Tutorial. [online] Available at: http://mccormickml.com/2013/12/13/adaboost-tutorial/ [Accessed 12 Apr. 2018].

Kihwan Kim. (2011).Rapid Object Detection using a Boosted Cascade of Simple Feature and Fast Face Detection via Morphology-Based Pre-processing.

Subhi Singh. (2015). Automatic Lecture Attendance System Using Face Reorganizationmatrix. *Academic International Online Journal of Engineering and Technology*.V3 (1).

Shireesha Chintalapati, M.V. Raghunadh. (2014), "Automated Attendance Management System Based On Face Recognition Algorithms", *IEEE International Conference on Computational Intelligence and Computing Research*.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. Upper Saddle River, N.J., Prentice Hall.

Kanan, C. and Cottrell, G. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition?. *PLoS ONE*, 7(1), p.e29740.

Pratiksha M. Patel (2016). Contrast Enhancement of Images and videos using Histogram Equalization. *International Journal on Recent and Innovation Trends in Computing and Communication*.V4 (11).

Sasi, N. and Jayasree, V. (2013). Contrast Limited Adaptive Histogram Equalization for Qualitative Enhancement of Myocardial Perfusion Images. *Engineering*, 05(10), pp.326-331.

Aliaa A. A. Youssif, and Atef Z. (2006).Comparative Study of Contrast Enhancement and Illumination Equalization Methods for Retinal Vasculature Segmentation. Proc. Cairo International Biomedical Engineering Conference.

A., I. and E.Z., F. (2016). Image Contrast Enhancement Techniques: A Comparative Study of Performance. *International Journal of Computer Applications*, 137(13), pp.43-48.

Bhuvaneshwari, K., Abirami, A. and Sripriya, N. (2017). Face Recognition Using PCA. *International Journal Of Engineering And Computer Science*, 6(4).

Abhishek Singh and Saurabh Kumar. (2012). *Face Recognition Using PCA and Eigen Face Approach*. [online] Available at: http://ethesis.nitrkl.ac.in/3814/1/Thesis.pdf [Accessed 10 Apr. 2018].

LC Paul and Abdulla Al Sumam. (2012). Face Recognition Using Principal Component Analysis Method. *IJARCET*, V1 (9).

D. Nithya (2015). Automated Class Attendance System based on Face Recognition using PCA Algorithm. *International Journal of Engineering Research and*, V4 (12).

Suman Kumar Bhattacharyya & Kumar Rahul. (2013), “Face Recognition by Linear Discriminant Analysis”, *International Journal of Communication Network Security*, V2(2), pp 31-35.

Ojala, T., Pietikainen, M. and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), pp.971-987.

Md. Abdur Rahim (2013), Face Recognition Using Local Binary Patterns. *Global Journal of Computer Science and Technology Graphics & Vision* V13 (4) Version 1.0.

Kasar, M., Bhattacharyya, D. and Kim, T. (2016). Face Recognition Using Neural Network: A Review. *International Journal of Security and Its Applications*, 10(3), pp.81-100.

Taigman, Y. et al., 2014. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *IEEE Conference on Computer Vision and Pattern Recognition*.

Divyarajsinh N Parmar et al. (2013).Face Recognition Methods and Applications. *Int.J.Computer Technology & Applications*,Vol 4 (1),84-86.

Deepesh Raj (2011), A Realtime Face Recognition system using PCA and various Distance Classifiers. *CS676 : Computer Vision and Image Processing*. Available at: http://home.iitk.ac.in/~draj/cs676/project/index.html [Accessed March 25, 2018].

Kamila, NK. (2015.). *Handbook of research on emerging perspectives in intelligent pattern recognition, analysis, and image processing*.

Lee, S. and Li, S. (2007). *Advances in Biometrics*. Berlin: Springer.

Kalyan Sourav Dash. (2014). *Face recognition using PCA - File Exchange - MATLAB Central*. [online] Mathworks.com. Available at: https://www.mathworks.com/matlabcentral/fileexchange/45750-face-recognitionusing-pca [Accessed 11 Apr. 2018].

Mathworks.com. (2018). *Detect objects using the Viola-Jones algorithm - MATLAB*. [online] Available at:https://www.mathworks.com/help/vision/ref/vision.cascadeobjectdetectorsystem-object.html [Accessed 22 Apr. 2018].

Cvc.cs.yale.edu. (1997). *Yale Face Database*. [online] Available at: http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html [Accessed 22 Apr. 2018].

Luxand.com. (2018). *Luxand - Face Recognition, Face Detection and Facial Feature Detection Technologies*. [online] Available at: https://www.luxand.com/ [Accessed 22 Apr. 2018].

Kihwan Kim. (2011). *Computer Vision Final Project:Viola-Jones & Morphologybased Face Detector*. [online] Available at: https://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm [Accessed 22 Apr. 2018].