FACE RECOGNITION ATTENDANCE SYSTEM USING RASPBERRY PI

by

ABHISH KUMAR D 2015105502
SAKTHI SRI P 2015105528
SABARINATH M 2015105558
VAISHALI B 2015105566

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

ANNA UNIVERSITY, CHENNAI – 25

APRIL 2019

BONAFIDE CERTIFICATE

Certified that this project report titled ... **FACE RECOGNITION ATTENDANCE SYSTEM USING RASPBERRY PI** is the *bonafide* work of **ABHISH KUMAR D (2015105502)**, **SAKTHI SRI P (2015105528)**, **SABARINATH M (2015105558)** and **VAISHALI B (2015105566)** who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Electronics and communication Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

Place: Chennai **Prof. Dr. M. Shanmugapriya**
Date: Professor
Department of Electronics and communication Engineering
Anna University, Chennai – 25

COUNTERSIGNED

Head of the Department,
Department of Electronics and communication Engineering,
Anna University Chennai,
Chennai – 600025

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, **Prof. Dr. M. Shanmugapriya** for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us. Without her immense support through every step of the way, we could never have it to this extent.

We are extremely grateful to **Dr. Muttan**, Head of the Department of Electronics and Communication Engineering, Anna University, Chennai 25, for extending the facilities of the Department towards our project and for her unstinting support.

We express our thanks to the panel of reviewers **Dr. M. A. Bhagyaveni, Dr. J. Kamala, Dr. K. Gunaseelan** and **Dr. Ewins Pon Pushpa** for their valuable suggestions and critical reviews throughout the course of our project.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Abhish Kumar D **Sakthi Sri P** **Sabarinath M** **Vaishali B**

ABSTRACT

Attendance for the students is an important task in class. When done manually it generally wastes a lot of productive time of the class. As it the repetitive, Student authentication can be automated using various methods available in the market like bio-metric attendance. This proposed solution for the current problem is through automation of attendance system using face recognition. This project describes the method of detecting and recognizing the face in real-time. Raspberry Pi 3 model B is used for computation in the detection and recognition modules. This project describes an efficient algorithm using open source image processing framework known as OpenCV.

This system is build by five modules Face Detection, Face Preprocessing, Face Training, Face Recognition and Attendance Database. The face database is collected to recognize the faces of the students. The system is initially trained with the students faces. The system uses user friendly User interface to maximize the user experience while both training and testing.

This project uses (LBP histograms) Local Binary Patterns Histograms for face recognition and uses MySQL to update the database. The system will automatically update the students presence in the class to the students database.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

LBPH : Local Binary Patterns Histograms

GUI : Graphical User Interface

RFID : Radio Frequency Identification

LBP : Local Binary Pattern

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Real time Face recognition system for attendance applications using Open CV library, uses Local Binary Patterns Histograms (LBPH). When the recognition of the student is completed the attendance is automatically updated to the database with name, date and time. A web application is used for viewing the status of the same which shows the attendance report of the class on a particular date. This model also considers possible threats like spoofing and this is avoided by using the eye blink detector algorithm to recognize the viewer avoiding security threats.

1.2 OBJECTIVES

To develop a prototype that will facilitate classroom control and attendance by face detection and recognition of students faces using Raspberry Pi and a webcam.

- The system should be able to detect students frontal faces in a class room within 30
- The system should be able to automatically reveal the number of students present on a GUI
- Recognise student stored on a database of faces by matching them to images on a database with an accuracy within30
- The system should be able to match detected students faces

cropped from an image to those on a database on the system.

- The system should be able to process an image within 10 minutes to be able to achieve the objective of recognition by the end of a lecture. i.e. 5 names per hour per lecture
- The algorithm implemented for the systems functionality will achieve system accuracy within 20
- The system designed will be user friendly with a Graphical User Inter phase that will serve as an access to the functionalities of the system

1.3 PROBLEM STATEMENT

Currently, industries, organisations are using personal identification strategies such as RFID, Iris recognition, Fingerprint identification for taking attendance. Face recognition has several applications in attendance management systems and security systems. In this work, a system is implemented that takes attendance for students during lecture, employees in industries etc. using face detection and recognition technology. A time period is set for taking attendance and the database is automatically uploaded into the web server through the internet connectivity. This process is done without any human intervention.In the system a raspberry Pi installed with openCV library ad a Raspberry Pi Camera module is connected for facial detection and Recognition. The data is stored in the memory card connected to Raspberry Pi and it can be accessed through the internet. The results show that a continuous observation increases accuracy and maximises the output.

1.4 SWOT ANALYSIS

1.4.1 Strengths

Automation simplifies time tracking. Attendance system using facial recognition technology can accurately report attendance, absence, and overtime with an identification process that is fast as well as accurate.It helps in labor cost savings and time fraud. It tightens security. It eases integration as it can be easily programmed into time and attendance system.

1.4.2 Weakness

Image quality affects how well facial recognition algorithms work.The image quality of a scanning video is quite low compared with that of the original camera. When a face-detection algorithm finds a face in an image or in a still from a video capture, the relative size of that face compared with the enrolled image size affects how well the face will be recognized.The facial angle of the target's face influences the recognition score.

1.4.3 Opportunities

This project can be used for many other applications where face recognition can be used for authentication. Raspberry Pi usage helps in minimizing the cost of the product and the usability as it can be connected to any device to take the attendance.

1.4.4 Threats

Data uncertainty in face recognition due to various environmental factors and spoofing face recognition with 3D- masks are two major threats to gain illegitimate access. A single image of face has got high uncertainty in representing the face since the face image varies with

illumination, facial expression and pose.

1.5 OVERVIEW OF THESIS

The rest of the thesis is organized as follows: Chapter 2 discusses the related researches that have been done before. It also discusses about the advantages and disadvantages of every approach.Chapter 3 gives the requirements analysis of the system. It explains the hardware and software requirements, constraints and assumptions made in the implementation of the system. Chapter 4 explains the overall system architecture and the design of various modules along with their pseudo code.Chapter 5 gives the results obtained from each module, describing the algorithms used. Here, we also discuss the performance measures when compared with previous approach. Chapter 6 concludes the thesis and gives an overview of its criticisms. It also states the various extensions that can be made to the system to make it function more effectively.

CHAPTER 2

RELATED WORK

2.1 PREVIOUS APPROACHES

Feraud et al (2000) discuss face detection as To detect a face in an image means to find its position in the image plane and its size or scale. The detection of a face in a digital image is a prerequisite to any further processing face recognition or any face processing software.

Face detection algorithms focused mainly on the frontal part of the human face (Srinivasan, Golomb and Martinez, 2016).However, in recent years, Cynganek, suggest that newer algorithms take into consideration different perspectives for face detection. Researchers have used such systems but the most challenge that has been faced is to make a system detect faces irrespective of different illumination conditions.

Yang et al (2002) classifies face detection methodologies into four major categories: Knowledge-based, Feature invariant, Template matching and appearance-based approaches.

Knowledge Based Method that uses human knowledge or human coding to model facial feature based on nature of the human face such as two eyes, mouth and the nose. This is very easy to apply the rules but very difficult to detect in various background depending on the pose and illumination. Low detection accuracy with small burden of calculation and short detection time.Feature-Based-Methods that uses algorithms to look for structural features regardless of pose, viewpoint or lighting conditions to find faces. Template Matching Methods;uses standard facial

patterns stored for use to correlate an input image with the stored pattern to compute for detection. Appearance Base Methods; uses a set of training sets of images to learn the templates and capture the representative of facial appearance. Most image-based approaches use window-scanning techniques for face detection.Window scanning algorithm searches for possible face locations at all scales.

Hjelms and Low, (2003) classifies face detection methodologies into two major categories.Image-based approaches, which is further subcategorized into Linear subspace methods, Neural networks and statistical approaches.Image Based Approaches; Most of the recent feature-based attempts in the same study by Hjelmsand Low, (2003) have improved the ability to cope with variations, but are still limited to head,shoulder and part of frontal faces. Furthermore,this method ignores the basic knowledge of the face in general and uses face patterns from a given set of images. This is mostly known as the training stage in the detection method.

Lenc and Krl (2014 pp.759-769) classify face recognition into various approaches; Correlation Method, compares two images by computing the correlation between them, with the images handled as one-dimensional vectors of intensity values. The images are normalized to have zero mean and unit variance with the nearest neighbour classifier used in the image directly. With these considerations stated, the light source intensity and characteristics of the camera are suppressed. The limitations of this method are; Large amount of memory storage needed, the corresponding points in the image space may not be tightly clustered and it is computationally expensive

Eigenfaces, This method considers the whole image as a vector. With this method, performance depends on alignment of the images with approximately the same pose. The change in lighting conditions, scale,

pose and other dissimilarities decreases the recognition rate rapidly

Fisher faces the most widely and effectively used methods for recognition of faces. This method depends on the method of appearance. Linear or fisher discriminant analysis for face recognition established in the year 1930 by Fisher. It one of successful methods that are used for face recognition procedure Belhumer et al. authenticated the method called LDA. This LDA method used for the finding of set of centre images that maximizes the ratio of the outside the class scatter and within the class scatter. This method has some drawbacks that the session the distribute medium will be perpetually alone ever since pixels of number image more than that pictures that are maximized for detection error rate so that if any alteration is posed and brightness if there changes that is inside the pictures that are same.

Local Binary Patterns; first used in texture as texture descriptor, the operator uses the value of the central pixel to threshold a local image region. The pixels are labelled either as 0 or 1 depending on whether the value is lower or greater than the threshold. Linna et al. (2015) in their study, proposed a system (Online Face Recognition System) that is based on LBP and Facial Landmarks, which uses nearest neighbor classifier in LBP histogram matching. They experimented the system on the videos of Honda/UCSD video database and achieved recognition rates of 64.0

2.2 CONCLUSION

The above section presents the various approaches to the given problem. It it clear that simple algorithms like Correlation Method, Eigenfaces,Fisher faces do not optimize energy, though they are currently put to use, they are not efficient.Local Binary Patterns based approaches can overcome this drawback.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 HARDWARE AND SOFTWARE REQUIREMENTS

The hardware and software requirements for face detection includes :

- Raspberry Pi 3B model
- Webcam
- SD card of minimum 8 GB
- Ethernet Cable
- A good internet connection

In-order to perform training we need the following

- A laptop with decent specs
- Webcam
- A good internet connection
- Webserver Apache that serve the GUI

CHAPTER 4

SYSTEM DESIGN

4.1 PROPOSED SYSTEM

The proposed system is used for taking attendance by using face recognition and managing the attendance in suitable environments such as colleges and offices. Raspberry Pi Camera Module V2 attached to Raspberry Pi3 and it is placed where the people enter the class. Camera Module is used to capture video from which images of human faces is extracted.

Then face recognition takes place and it automatically verifies with the existing database through library files present in OpenCV. Face Recognition is generally more advanced and efficient than other systems.

Another algorithm for Blink Detection is also implemented to avoid fraudulent cases wherein a static photo can be used to enter attendance and not by the actual person.

4.2 ARCHITECTURE DIAGRAM

In this section, we propose our system design of Face Recognition Attendance System using Raspberry Pi in Fig 4.1. Our framework consists of three major components: 1) External Hardware 2) Data set Creator and Trainer and 3) Graphical User Interface. Over these system components, the entire algorithm runs in three phases: Data set Creation, Training, Verification and Identification.

The data set creation phase captures images from the video, detect and crop faces if any and update the database. The images are sampled at frame rates corresponding to the camera's specifications to avoid uncertainties in image details. The Training phase implements the chosen algorithm to extract elements of interest (the elements which will be compared) from the detected faces. The verification and Identification phase involves comparison of the detected and trained data to identify the person and hence update the attendance database. The GUI provides an user interface to view the updated data over an Web Application.

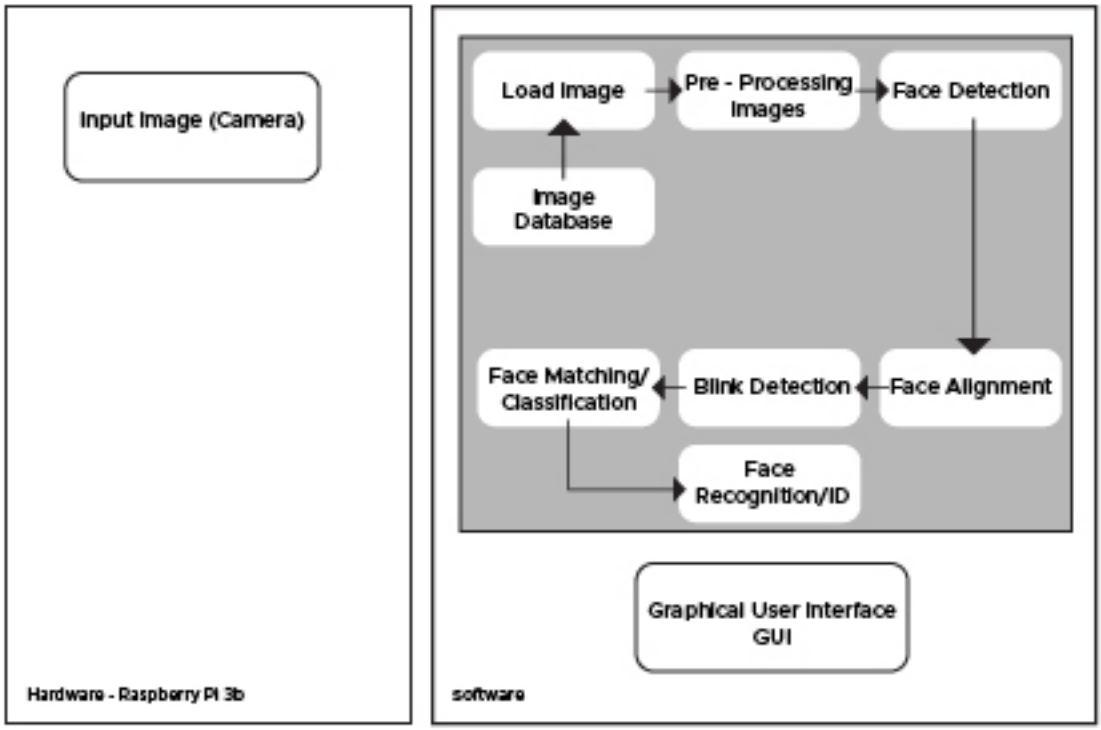


Figure 4.1 ARCHITECTURE DIAGRAM

4.3 DETAILED MODULE DESIGN

4.3.1 Image Database

The camera module is placed in a region where the people enter into college or office and video is taken within the distance less than 5 meters. A camera is used for taking video which contains many frames from which any one of the frames can be used for face recognition and marking the attendance. As a bio metric method has been chosen for implementation, it is crucial for enrollment of every individual whose attendance needs to be taken. Here face of every individual is captured and stored in a suitable database (sql) which includes the person's name and other credentials.

4.3.2 Pre - Processing Images using LBPH Algorithm

Module Description

Local Binary Pattern (LBP) is a simple yet very efficient

texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

Parameters: the LBPH uses 4 parameters:
Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

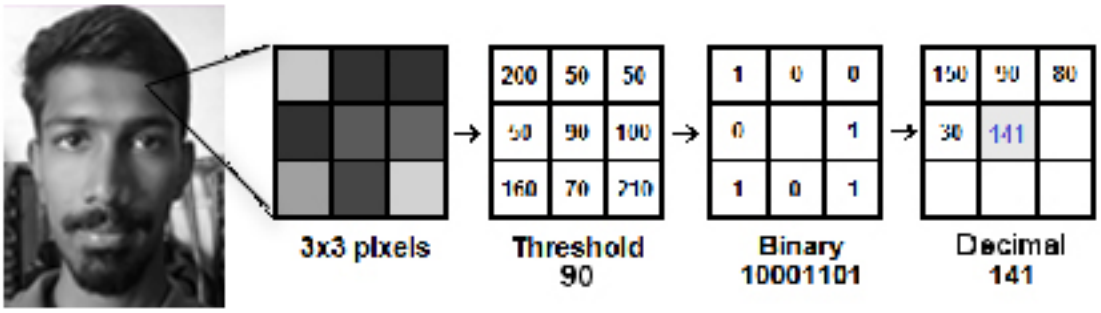


Figure 4.2 APPLYING THE LBPH OPERATION

Based on the image above, lets break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0 255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concate-

nate the binary values (e.g. clockwise direction), but the final result will be the same.

- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.
- At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

Note: The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.

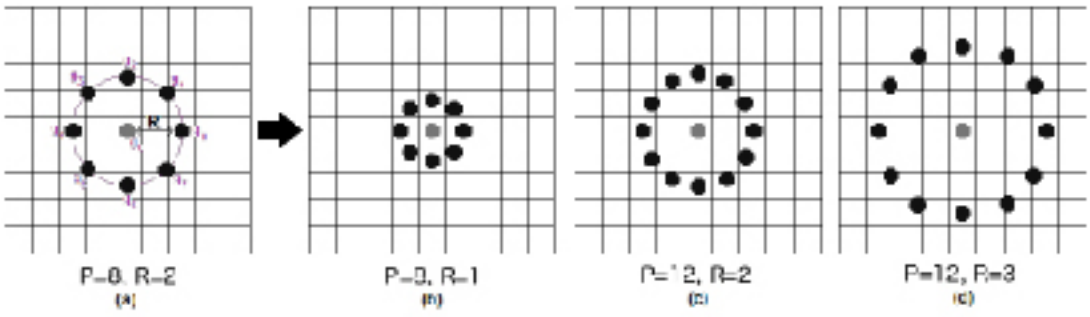


Figure 4.3 Circular LBP

It can be done by using bi linear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

Extracting the Histograms: Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:

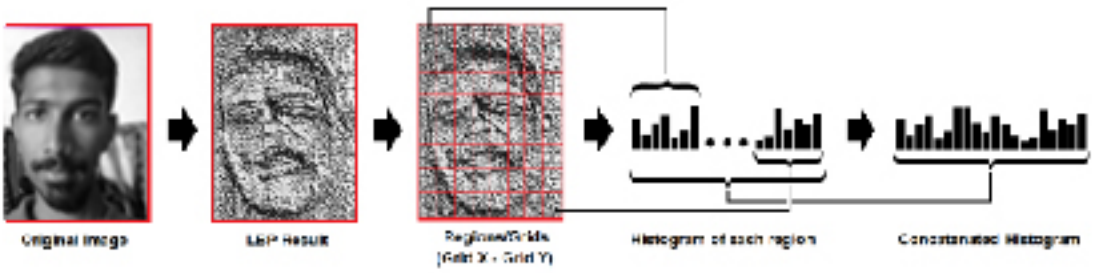


Figure 4.4 Extracting the Histograms

Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0 255) representing the occurrences of each pixel intensity.
- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image

The LBPH algorithm is pretty much it.

4.3.3 EYE MOVEMENT DETECTOR

Module Description

The traditional face detector algorithms fail to differentiate between a static image such as photos , and a regular video stream. OpenCV does not provide native methods to identify if the input feed is live or static. Unfortunately this particular shortcoming can become a security vulnerability in an attendance marking system. Therefore it is important to identify whether the input feed from the camera is not a static image of a person.

The above task is performed by the eye movement detector. This module makes use of the facial landmark detection algorithm. Unlike traditional image processing methods for computing blinks which typically involve some combination of: Eye localization. Thresholding to find the whites of the eyes. Determining if the white region of the eyes disappears for a period of time (indicating a blink). These methods do not work real time. The eye aspect ratio is instead a much more elegant solution that involves a very simple calculation based on the ratio of distances between facial landmarks of the eyes.

Each eye is represented by 6 (x, y)-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region.There is a relation between the width and the height of these coordinates and we can derive a relation between these coordinates.

$$EAR = \frac{||p2 - p6|| + ||p3 - p5||}{2||p1 - p4||}$$

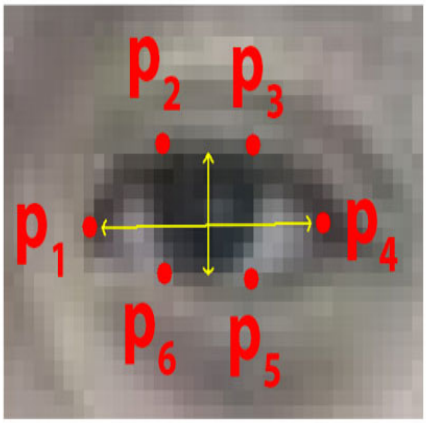


Figure 4.5 Coordinates of eyes

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only one set of horizontal points but two sets of vertical points.the eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place.Using this simple equation, we can avoid image processing techniques and simply rely on the ratio of eye landmark distances to determine if a person is blinking.

Thus this module helps us to differentiate a static image from a live person

INPUT: Feed from webcam

OUTPUT: Decision whether the input feed is live or static image

PSEUDOCODE/ALGORITHM

- earThreshold = 0.27, counter = 2, blink=0 //Parameters
- lStart, lEnd = facialLandmark(dutil.leftEye)
- rStart, rEnd = facialLandmark(dutil.rightEye)
- leftEye,RightEye = shape[lStart,lEnd], shape[rStart, rEnd]
- leftEAR = eyeAspectRatio(leftEye)
- rightEAR = eyeAspectRatio(rightEye)
- ear = (leftEAR + rightEAR) / 2.0
- ear ¡ earThreshold ? blink++ : resetEar
- if (blink) $>$ 0 return true

4.3.4 Face Detector

Module Description

Face detection method notices the segment faces by shaping spheres on image having pictures of learners. Haar classifiers are used for the recognition of the faces. This face detection algorithm are first used for changeability of images with different action and then it used for illumination conditions and this also applied for detecting the expressions on face in existent phase audio-visual. First this procedure used for pictures and functioned for classroom of image for finding of multiple faces in the image. Next step after discovery from images the faces are collected for identifying faces. This procedure winding is recycled to rise the speed of the algorithm. The image that is collected is then allotted to distinct drift because of purpose of knowledge reasons.

INPUT: Raw Images from the Camera.

OUTPUT: Detected set of faces.

Face detection must deal with several well known challenges. They are usually present in images captured in uncontrolled environments, such as surveillance video systems. These challenges are attributed to the following factors like :/new

Pose variation. The ideal scenario for face detection would be one in which only frontal images were involved. But, as stated, this is very unlikely in general uncontrolled conditions. Moreover, the performance of face detection algorithms drops severely when there are large pose variations. Its a major research issue. Pose variation can happen due to subjects movements or cameras angle.

Feature occlusion. The presence of elements like beards, glasses or hats introduces high variability. Faces can also be partially covered by objects or other faces.

Facial expression. Facial features also vary greatly because of dierent facial gestures.

Imaging conditions. Dierent cameras and ambient conditions can aect the quality of an image, aecting the appearance of a face.

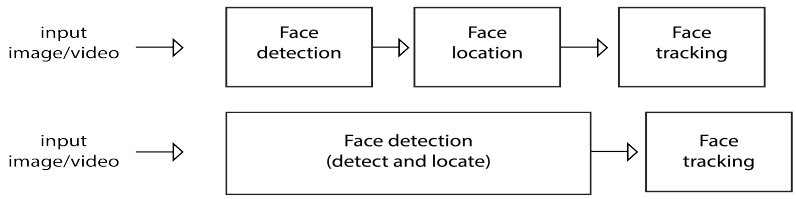


Figure 4.6 Face Detection Process

PSEUDOCODE/ALGORITHM

4.3.5 Attendance Manager

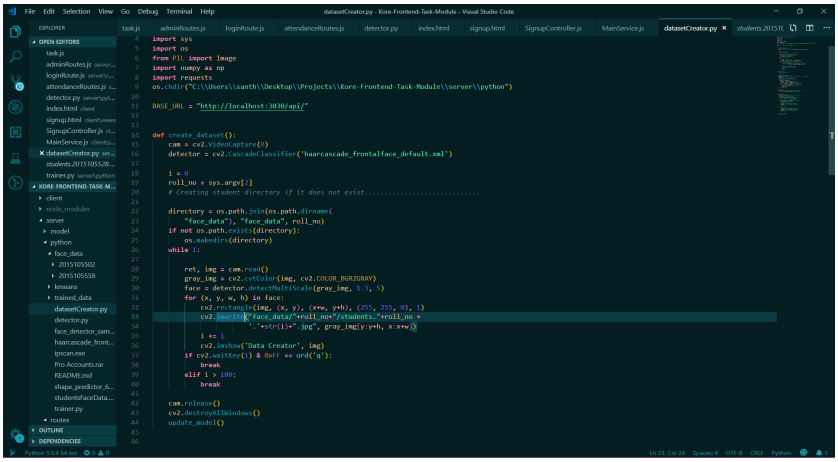
Module Description

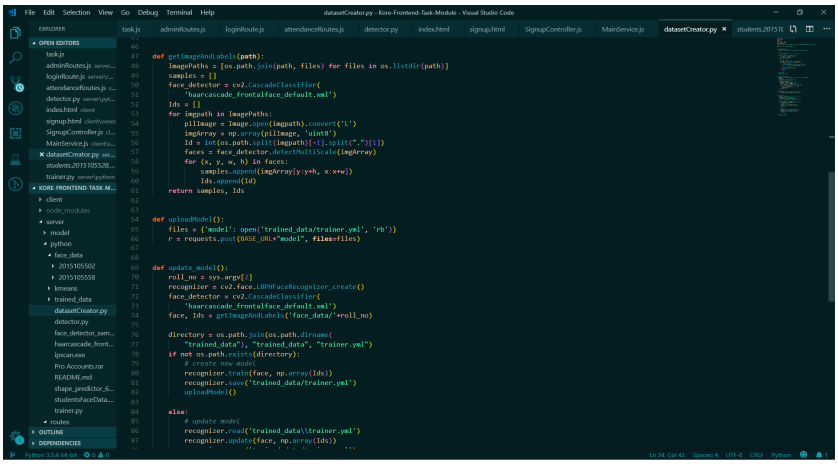
Once the appropriate face has been detected in the Pi, attendance has to marked against that specific student. To serve this purpose we make use of a web server and a relational database. After detecting the student's face and deciding if its real (using eye movement detector) an API call is made to the attendance manager via the internet. This module does not reside on the Pi itself, but on some remote system or server. This is to enhance the performance of the Pi by freeing up RAM space. The face detector detects face using a student's ID , this is sent to the attendance manager module along with the time of detection. The attendance manager depending upon the time, matches it with the available subject list to determine for which subject attendance is to be marked. Then it inserts this subject student and hour tuple into the relational database. The attendance manager is also responsible for supplying the marked attendance data to the admin. It does this by querying the appropriate data from the relational database and sending it to the admin panel on-demand.

INPUT: Student ID and time of detection.

OUTPUT: Marked attendance.

Pseudocode of the dataset creator





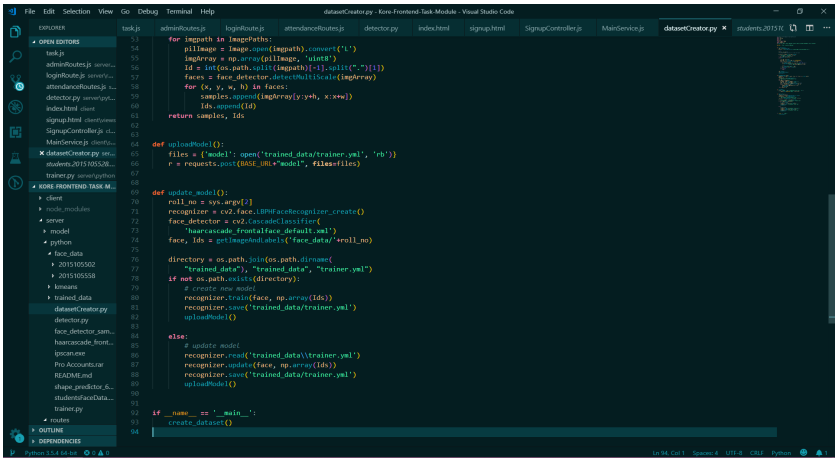
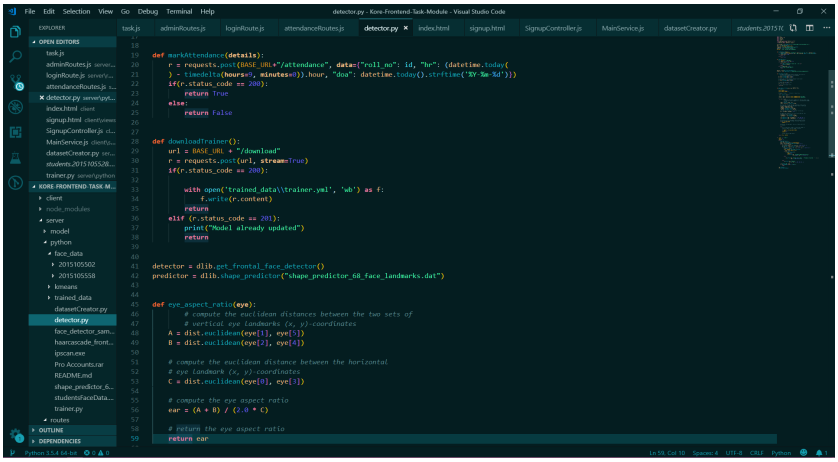
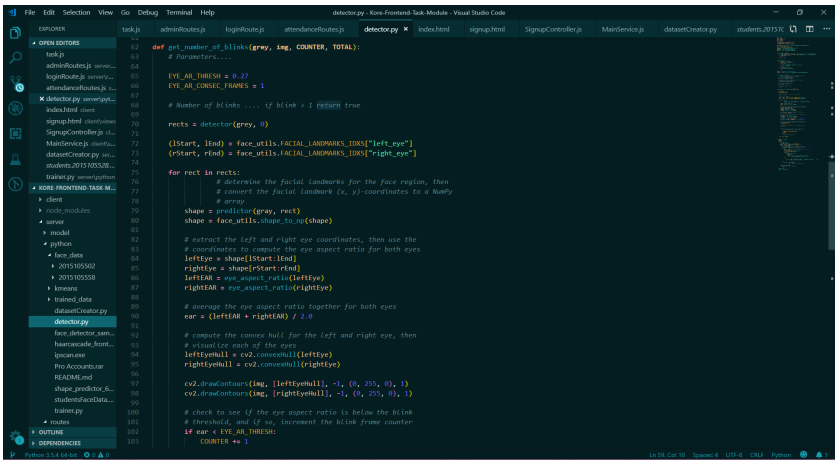


Figure 4.7 Dataset creation algorithm

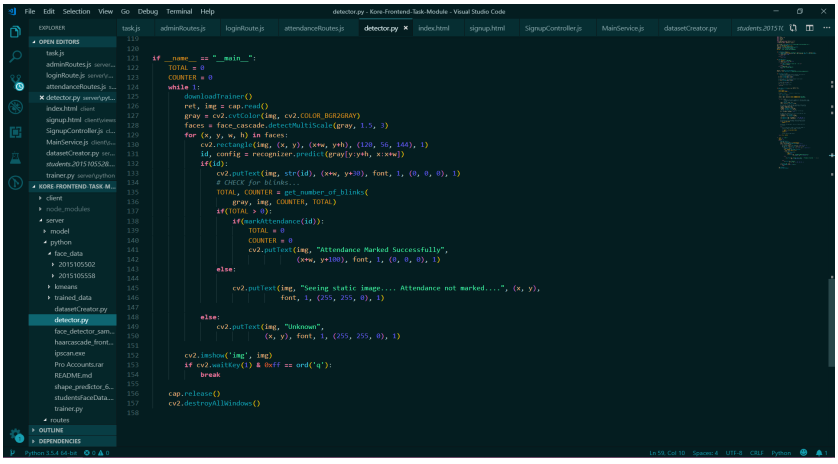
Pseudocode of the detector



```
def markAttendance(details):
    r = requests.post(BASE_URL+"/attendance", data={"roll_no": id, "hr": (datetime.today(
    ) - timedelta(hours=5, minutes=30)).hour, "doa": datetime.today().strftime('%Y-%m-%d')})
    if(r.status_code == 200):
        return True
    else:
        return False


def downloadTrainer():
    url = BASE_URL + "/download"
    r = requests.post(url, stream=True)
    if(r.status_code == 200):

        with open('trained_data\\trainer.yml', 'wb') as f:
            f.write(r.content)
        return
    elif (r.status_code == 201):
        print("Model already updated")
        return


detector = dlib.get_frontal_face_detector()
predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat")


def eye_aspect_ratio(eye):
        # compute the euclidean distances between the two sets of
        # vertical eye landmarks (x, y)-coordinates
    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    # compute the euclidean distance between the horizontal
    # eye landmark (x, y)-coordinates
    C = dist.euclidean(eye[0], eye[3])

    # compute the eye aspect ratio
    ear = (A + B) / (2.0 * C)

    # return the eye aspect ratio
    return ear
```



```
def get_number_of_blinks(grey, img, COUNTER, TOTAL):
    # Parameters....

    EYE_AR_THRESH = 0.27
    EYE_AR_CONSEC_FRAMES = 1

    # Number of blinks .... if blink > 1 return true

    rects = detector(grey, 0)

    (lStart, lEnd) = face_utils.FACIAL_LANDMARKS_IDXS["left_eye"]
    (rStart, rEnd) = face_utils.FACIAL_LANDMARKS_IDXS["right_eye"]

    for rect in rects:
                # determine the facial landmarks for the face region, then
                # convert the facial landmark (x, y)-coordinates to a NumPy
                # array
        shape = predictor(grey, rect)
        shape = face_utils.shape_to_np(shape)

        # extract the left and right eye coordinates, then use the
        # coordinates to compute the eye aspect ratio for both eyes
        leftEye = shape[lStart:lEnd]
        rightEye = shape[rStart:rEnd]
        leftEAR = eye_aspect_ratio(leftEye)
        rightEAR = eye_aspect_ratio(rightEye)

        # average the eye aspect ratio together for both eyes
        ear = (leftEAR + rightEAR) / 2.0

        # compute the convex hull for the left and right eye, then
        # visualize each of the eyes
        leftEyeHull = cv2.convexHull(leftEye)
        rightEyeHull = cv2.convexHull(rightEye)

        cv2.drawContours(img, [leftEyeHull], -1, (0, 255, 0), 1)
        cv2.drawContours(img, [rightEyeHull], -1, (0, 255, 0), 1)

        # check to see if the eye aspect ratio is below the blink
        # threshold, and if so, increment the blink frame counter
        if ear < EYE_AR_THRESH:
            COUNTER += 1
```



```
if __name__ == "__main__":
    TOTAL = 0
    COUNTER = 0
    while 1:
        downloadTrainer()
        ret, img = cap.read()
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(gray, 1.5, 3)
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x+w, y+h), (120, 56, 144), 1)
            id, config = recognizer.predict(gray[y:y+h, x:x+w])
            if(id):
                cv2.putText(img, str(id), (x+w, y+30), font, 1, (0, 0, 0), 1)
                # CHECK for blinks....
                TOTAL, COUNTER = get_number_of_blinks(
                    gray, img, COUNTER, TOTAL)
                if(TOTAL > 0):
                    if(markAttendance(id)):
                        TOTAL = 0
                        COUNTER = 0
                        cv2.putText(img, "Attendance Marked Successfully",
                                    (x+w, y+100), font, 1, (0, 0, 0), 1)
                    else:

                        cv2.putText(img, "Seeing static image.... Attendance not marked....", (x, y),
                                    font, 1, (255, 255, 0), 1)

                else:
                    cv2.putText(img, "Unknown",
                                (x, y), font, 1, (255, 255, 0), 1)

        cv2.imshow('img', img)
        if cv2.waitKey(1) & 0xff == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()
```

Figure 4.8 Face detector algorithm

CHAPTER 5

RESULTS

5.1 PERFORMANCE OF THE SYSTEM

The performance of the system in correspond to the user's interaction with the system and the surrounding environment for different conditions and cases are discussed below

5.1.1 Case 1

Whenever an admin wishes to add a new student into the database, he uses the following GUI. The output obtained at the end is also shown.

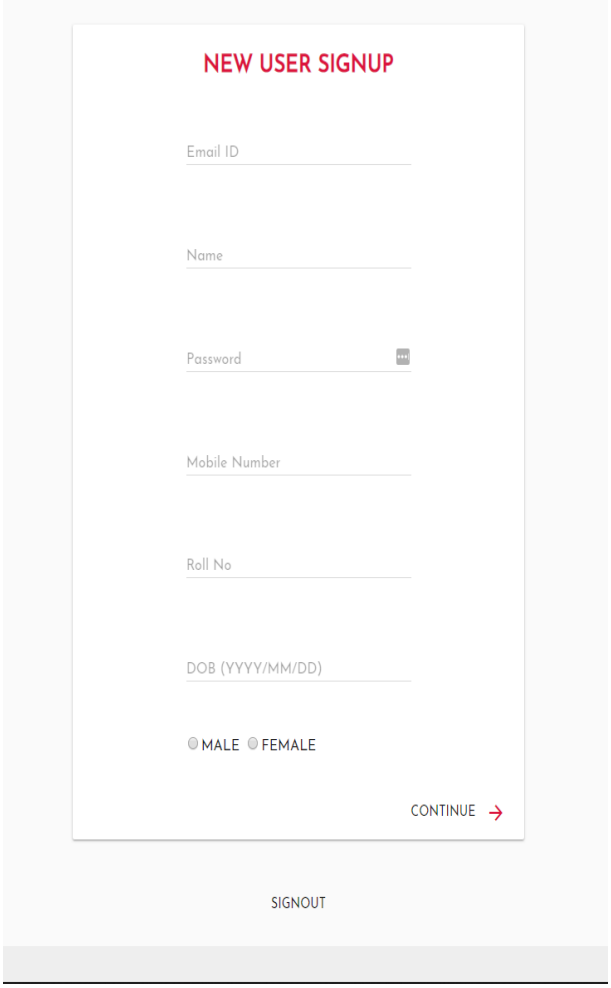


Figure 5.1 Add a new student to database

5.1.2 Case 2

If an Admin wishes to add student who's ID already exists in the database, the following error message is received

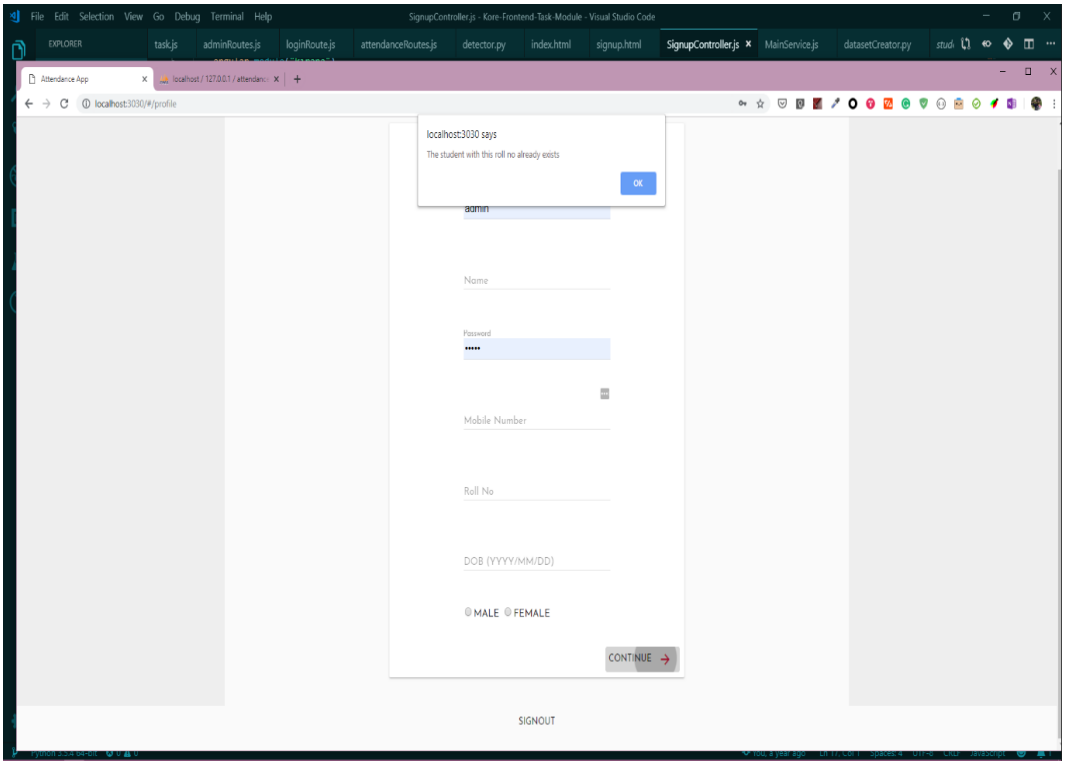


Figure 5.2 Error message of repetation of the dataset

5.1.3 Case 3

Output obtained in the face detector under normal light conditions. When multiple faces are included in the live feed, the detector marks present for all these identified faces. This is shown by the below screen shot.

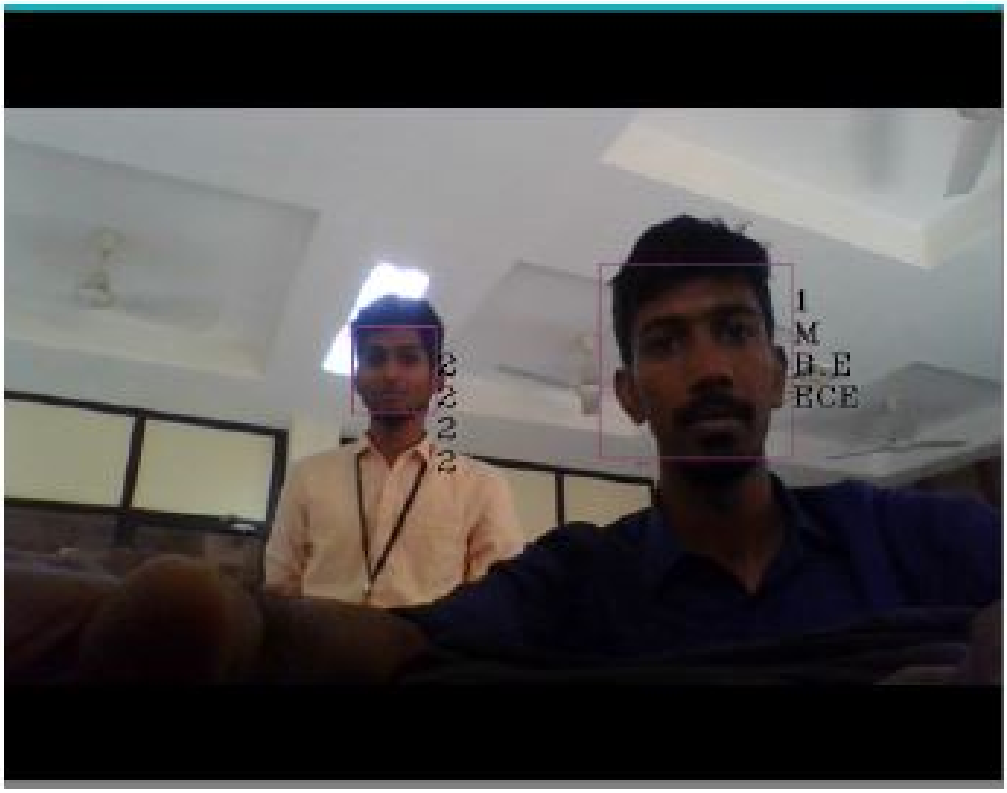


Figure 5.3 Multiple face detection

5.1.4 Case 4

When a static image is given as input instead of live feed of a person, the detector does not identify the face but shows a warning message as shown below

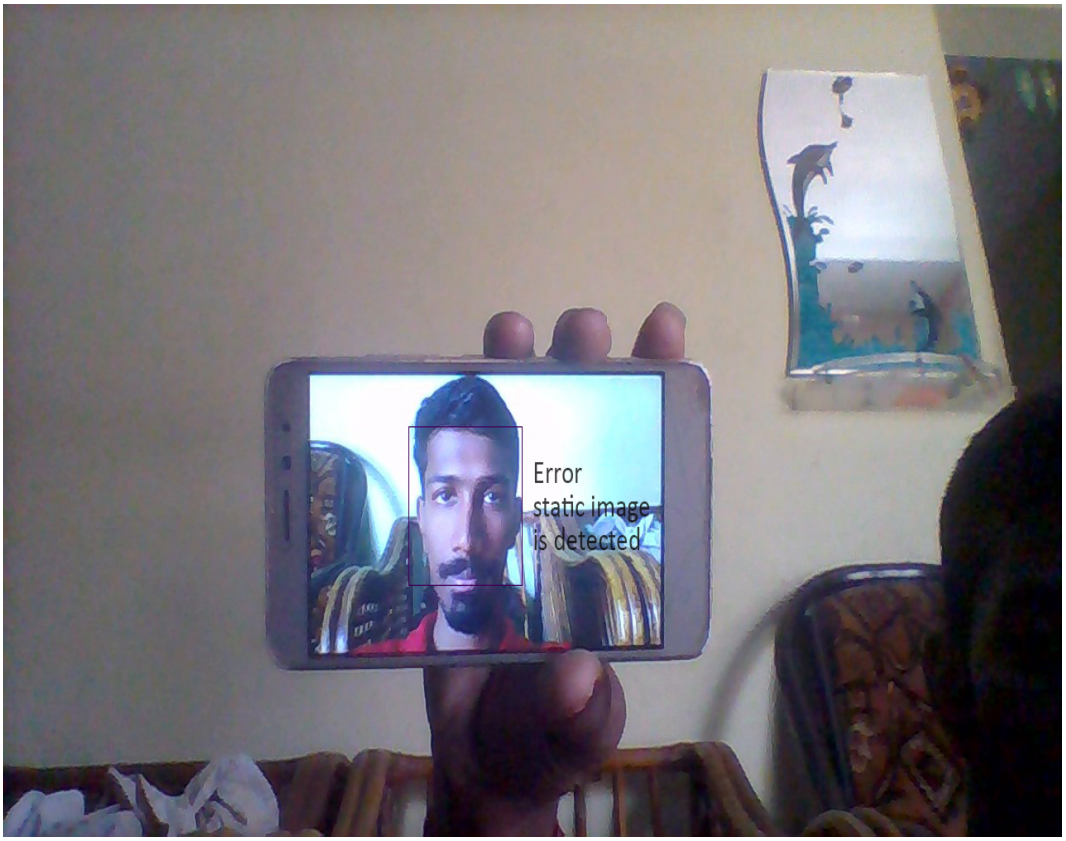


Figure 5.4 Warning message for the static image

5.1.5 Case 5

When the person in the live feed blinks his eyes, the detector records the eye movement as shown below

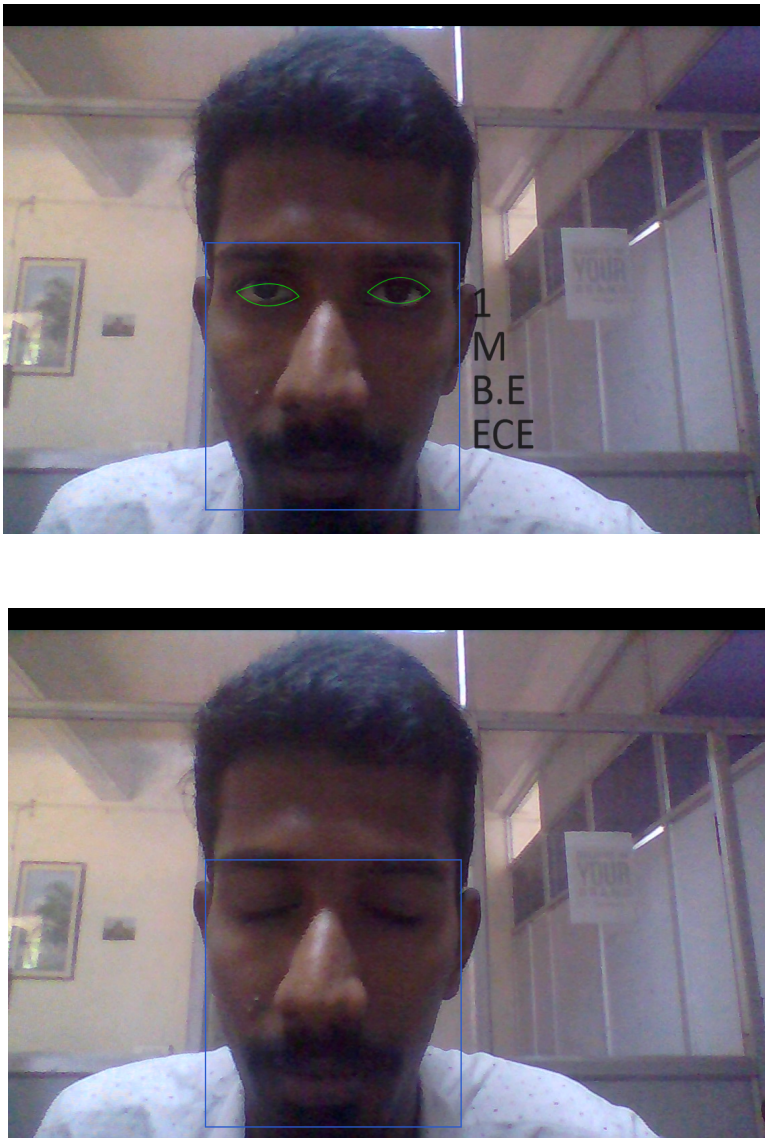


Figure 5.5 Detection of the eye movement

5.1.6 Case 6

When all environmental and hardware conditions are suitable, under normal circumstances attendance is marked by the attendance manager as shown below

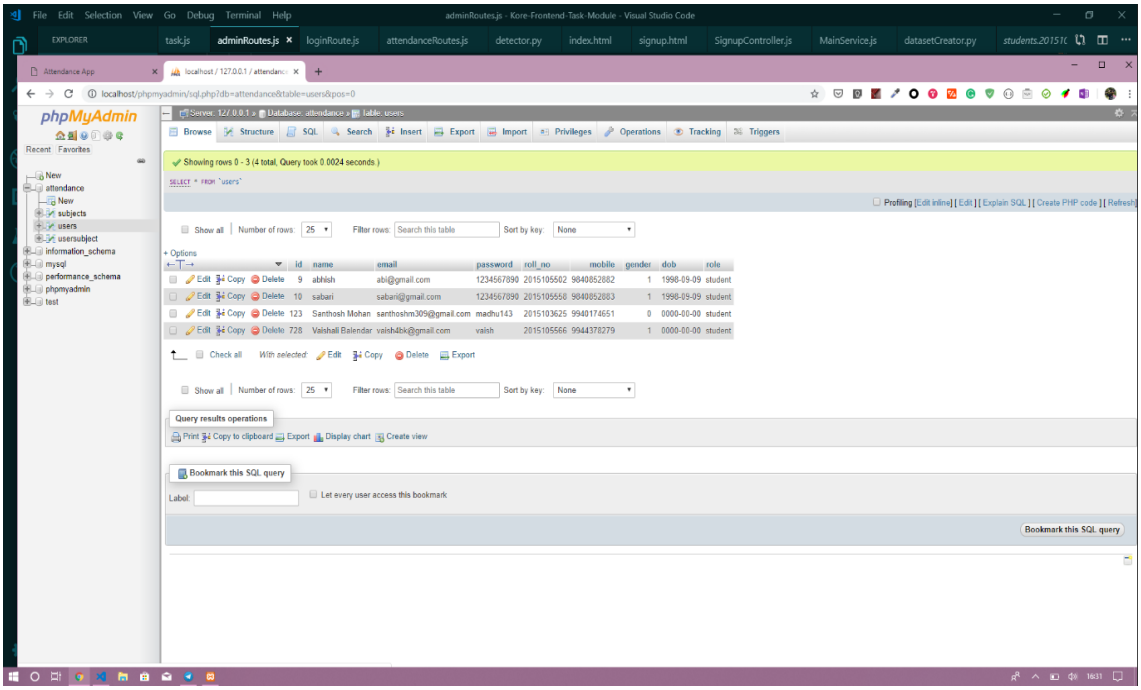


Figure 5.6 Database of the attendance manager

5.2 PERFORMANCE METRICS

5.2.1 System performance with respect to the number of samples in data set creation

The below graph shows how the system performance is affected by the number of samples stored in the data set. The LBPH algorithm has been used to detect the faces and train them. The graph shows, as the number of samples reduces the time taken to detect the faces increases hence the performance decreases. On the other hand increasing the number of samples in the data set makes the detection process quicker and improves the performance

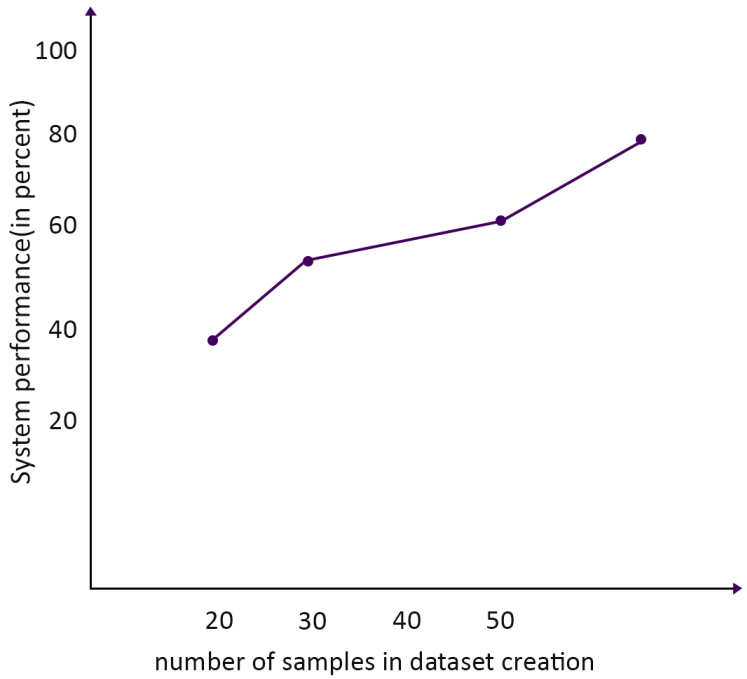


Figure 5.7 System performance vs number of samples in data set creation

5.2.2 System performance with respect to the number of persons in class

The below graph shows how the class population affects the system performance. If a class has less number of people say 30 then to detect a person's face from the class, the system compares the image with 30 person's(30* 100 samples) trained data set. If the class population is high then, for detecting a person's face the system has to compare with higher number of student's data set. So, lower the population faster is the detection process.

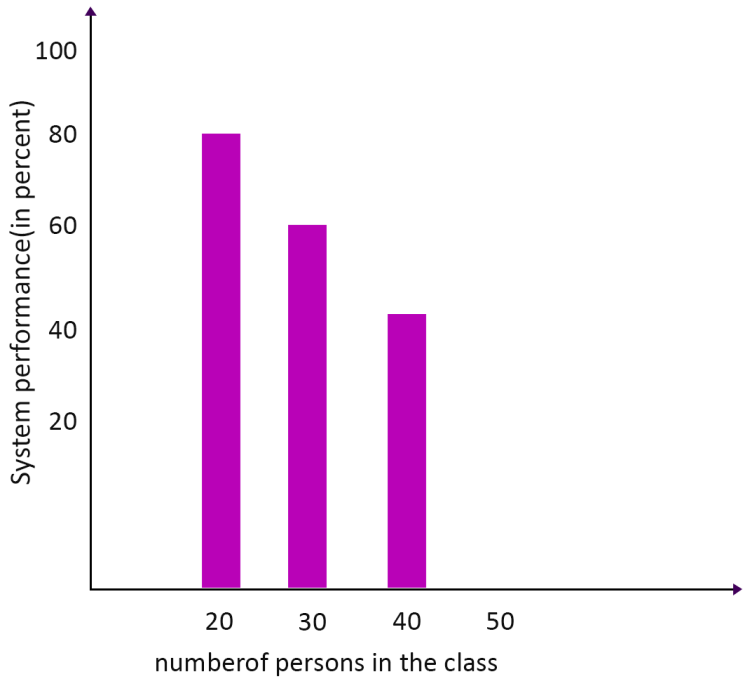


Figure 5.8 System performance vs number of persons in class

CHAPTER 6

CONCLUSIONS

6.1 SUMMARY

The face recognition was done using LBPH and raspberry pi platform. To reduce the false-positives drastically and increase the efficiency,in this research, we are using haar like features and for recognition of face we are using LBPH (local binary pattern histogram).This reference design can be used for authentication in banks ,and other public places.Thus for a safety purpose in real time we designed a face recognition system in minimum expenses using raspberry pi, open cv and lbp algorithm. LBPH is one of the easiest face recognition algorithms. It can represent local features in the images. It is possible to get great results (mainly in a controlled environment). It is robust against monotonic gray scale transformations. It is provided by the OpenCV library (Open Source Computer Vision Library).

6.2 FUTURE WORK

The future work is based on LBP but a bit improved on Local-Color-Vector Binary Pattern (LCVBP). The color images are taken and a improved bit of nine bit code is considered while as in LBP only eight bits are considered and in LBP there were only 256 labels but because of improved bit the bins or labels are increased to 512. The Gaussian distribution for multi variable is considered and multi blocks of lbp comes into picture and output which is matched with the data base will be in color. In future, this proposed approach will be more beneficial for security agencies to identify criminals, whose have criminal record in database. It will help to recognize any unknown or known person in surveillance area at low resolutions due to long distance of camera and observed subject.

REFERENCES

1. U. Eze peter; Owerri Nigeria; C.K.A. JoeUzuegbu; Laz Uzoechi; F.K. 2013. Opera biometric based attendance system with remote real time monitoring tertary institutions in developing countries emerging and sustainable technologies for power and ICT in a developing society NIGERCON, IEEE.

2. Adam Schmidt, Andrzej Kasinski, The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection, Computer Recognition Systems 2016.