SMART ATTENDANCE SYSTEM USING FACE RECOGNITION

A Mini Project Work

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

B SHIVA KUMAR	-	17H61A0467
BHARATH KUMAR KAIROJU	-	17H61A0470
HAZARI SAI JAGADEESH	-	17H61A0482

Under the guidance of

Dr.D.NARENDHAR SINGH

Associate Professor

Department of ECE

Department of Electronics and Communication Engineering

ANURAG GROUP OF INSTITUTIONS

AUTONOMOUS

SCHOOL OF ENGINEERING

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Venkatapur(V), Ghatkesar(M), Medchal-Malkajgiri Dist-500088

2020-2021

ANURAG GROUP OF INSTITUTIONS

AUTONOMOUS

SCHOOL OF ENGINEERING

(Affiliated to Jawaharlal Nehru Technological University, Hyderabad

Venkatapur(V),Ghatkesar(M), Medchal-Malkajgiri Dist-500088

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "SMART ATTENDANCE SYSTEM USING FACE RECOGNITION" being submitted by

B SHIVA KUMAR	17H61A0467
BHARATH KUMAR KAIROJU	17H61A0470
HAZARI SAI JAGADEESH	17H61A0482

in partial fulfillment for the award of the Degree of Bachelor of Technology in Electronics & Communication Engineering to the Jawaharlal Nehru Technological University, Hyderabad is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr.D.NARENDHAR SINGH
Associate Professor

Dr. S. SATHEES KUMARAN
Head of the Department
DEPT OF ECE

External Examiner

ACKNOWLEDGEMENT

This project is an acknowledgement to the inspiration, drive and technical assistance contributed by many individuals. This project would have never seen light of this day without the help and guidance we have received. We would like to express our gratitude to all the people behind the screen who helped us to transform an idea into a real application.

It's our privilege and pleasure to express our profound sense of gratitude to **Dr.D.Narendhar Singh, Associate Professor**, Department of ECE for his guidance throughout this dissertation work.

We express our sincere gratitude to **Dr.S.Sathees Kumaran, Head of Department,** Electronics and Communication Engineering for his precious suggestions for the successful completion of this project. He is also a great source of inspiration to our work.

We would like to express our deep sense of gratitude to **Dr.K.S.Rao**, **Director**, Anurag Group of Institutions for his tremendous support, encouragement and inspiration. Lastly, we thank almighty, our parents, friends for their constant encouragement without which this assignment would not be possible. We would like to thank all the other staff members, both teaching and non- teaching, which have extended their timely help and eased my work.

BY

B SHIVA KUMAR	**17H61A0467**
BHARATH KUMAR KAIROJU	**17H61A0470**
HAZARI SAI JAGADEESH	**17H61A0482**

DECLARATION

We hereby declare that the result embodied in this project report entitled “**Smart Attendance System using Face Recognition**” is carried out by us during the year 2019-2020 for the partial fulfilment of the award of **Bachelor of Technology** in **Electronics and Communication Engineering,** from ANURAG GROUP OF INSTITUTION. We have not submitted this project report to any other Universities / Institute for the award of any degree.

BY

B SHIVA KUMAR	**17H61A0467**	**signature**
BHARATH KUMAR KAIROJU	**17H61A0470**	**signature**
HAZARI SAI JAGADEESH	**17H61A0482**	**signature**

ABSTRACT

Uniqueness or individuality of an individual face is the representation of one's identity. In this project face of an individual is used for the purpose of attendance making automatically. Attendance of the student is very important for every college, universities and school. Conventional methodology for taking attendance is by calling the name or roll number of the student and the attendance is recorded. Time consumption for this purpose is an important point of concern. Assume that the duration for one subject is around 60 minutes or 1 hour & to record attendance takes 5 to 10 minutes. For every tutor this is consumption of time. To stay away from these losses, an automatic process is used in this project which is based on image processing. In this project face detection and face recognition is used. Face detection is used to locate the position of face region and face recognition is used for marking the understudy's attendance. The database of all the students in the class is stored and when the face of the individual student matches with one of the faces stored in the database then the attendance is recorded.

TABLE OF CONTENTS

Chapter 1

Introduction

1.1. Introduction

Attendance is prime important for both the teacher and student of an educational organization. So it is very important to keep record of the attendance. The problem arises when we think about the traditional process of taking attendance in class room.
Calling name or roll number of the student for attendance isnot only a problem of time consumption but also it needs energy. So an automatic attendance system can solve all above problems.

There are some automatic attendances making system which are currently used by much institution. One of such system is biometric technique and RFID system. Although it is automatic and a step ahead of traditional method it fails to meet thetime constraint. The student has to wait in queue for giving attendance, which is time taking.

This project introduces an involuntary attendance marking system, devoid of any kind of interference with the normal teaching procedure. The system can be also implemented during exam sessions or in other teaching activities where attendance is highly essential. This system eliminates classical student identification such as calling name of the student, orchecking respective identification cards of the student, which can not only interfere with the ongoing teaching process, but also can be stressful for students during examination sessions. 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.

1.2. 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 actuallyclassify shape, size, contour and the texture of the object in order to analyze the information. The analyzed 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 isimpossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computerswith 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 facerecognition were then conducted continuously until today (Ashley DuVal, 2012).

1.3. Problem Statement

Traditional student attendance marking technique is oftenfacing a lot of trouble. The face recognition student attendancesystem 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 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 theproblem 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 knownand 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 mustbe 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.

1.4. Aims and Objectives

The objective of this project is to develop face recognition 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

1.5. Scope of the Project

We are setting up to design a system comprising of two modules. The first module (face detector) is a mobile component, whichis basically a camera application that captures student faces and stores them in a file using computer vision face detection algorithms and face extraction techniques. The second module is a desktop application that does face recognition of thecaptured images (faces) in the file, marks the students register andthen stores the results in a database for future analysis.

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.

System Type	Advantage	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

Table 2.1: Advantages & Disadvantages of Different Biometric System [1]

2.2. Digital Image Processing

Digital Image Processing is the processing of images which are digital in nature by a digital computer [2]. Digital image processing techniques are motivated by three majorapplications mainly:

- Improvement of pictorial information for human perception
- Image processing for autonomous machine application
- Efficient storage and transmission.

[1] Arun Katara et al., 2017
3 Anil K Jain, Lin Hong, Sharath Pankanti, and Ruud Bolle, Biometric Identification. IEEE, 2004

3.1. Image Representation in a Digital Computer

An image is a 2-Dimensional light intensity function

$\mathbf{f}(\mathbf{x},y) = \mathbf{r}(\mathbf{x},y) \times \mathbf{i}(\mathbf{x},y)$ - (2.0)

Where,r (x, y) is the reflectivity of the surface of the corresponding image point. i (x,y) Represents the intensity of the incident light. A digital image f(x, y) is discretized both in spatial co-ordinates by grids and in brightness by quantization [3]. Effectively, the image can be represented as a matrix whose row, column indices specify a point in the image and the element value identifies gray level value at thatpoint. These elements are referred to as pixels or pels.

Typically following image processing applications, the image sizewhichisusedis 256×256, elements, 640×480 pels or 1024×1024 pixels. Quantization of these matrix pixels is done at 8 bits for black and white images and 24 bits for colored images (because of the three color planes Red, Green and Blue each at 8 bits)[4].

3.2. Steps in Digital Image Processing

Digital image processing involves the following basic tasks:

- Image Acquisition - An imaging sensor and the capability to digitize the signal produced by the sensor.
- Preprocessing – Enhances the image quality, filtering, contrast enhancement etc.
- Segmentation – Partitions an input image into constituent parts of objects.
- Description/feature Selection – extracts the description of image objects suitable for further computer processing.
- Recognition and Interpretation – Assigning a label to the object based on the information provided by its descriptor. Interpretation assigns meaning to a set of labelled objects.
- Knowledge Base – This helps for efficient processingas well as inter module cooperation.

3 N. Tom, Face Detection, Near Infinity - Podcasts, 2007 4 T. Kanade, Computer recognition of human faces. Basel [etc.]: Birkhäuser, 1977

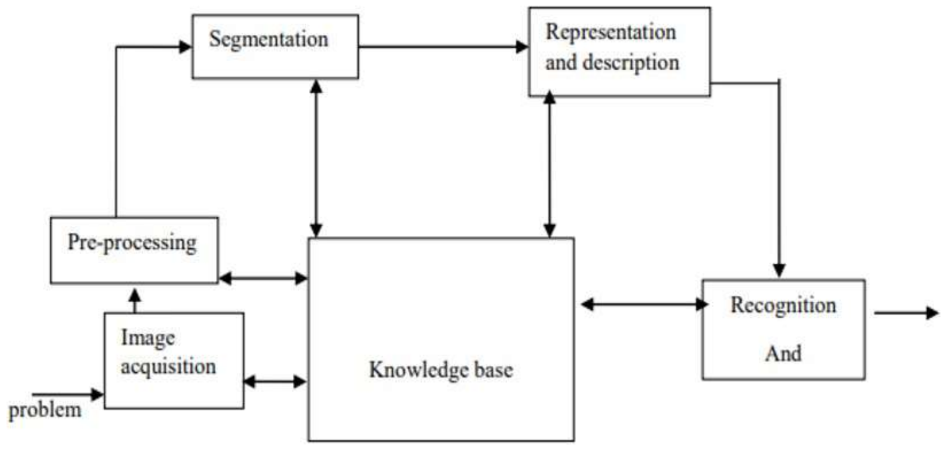


Fig: A diagram showing the steps in digital image processing

3.3. Definition of Terms and History

Face Detection

Face detection is the process of identifying and locating all the present faces in a single image or video regardless oftheir position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content [5].

Face Recognition

Face Recognition is a visual pattern recognition problem,where the face, represented as a three dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images [6].

Face Recognition is therefore simply the task of identifyingan already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is [7].

[5] A. L. Rekha and H. K. Chethan, "Automated Attendance System using face Recognition through Video Surveillance," Int. J. Technol. Res. Eng., vol. 1, no. 11, pp. 1327–1330, 2014.
[6] I. Kim, J. H. Shim, and J. Yang, "Face detection," Face Detect. Proj. EE368 Stanf. Univ., vol. 28, 2003.
[7] E. Shervin, "OpenCV Computer Vision," 03-Oct-2010

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 theconcept 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

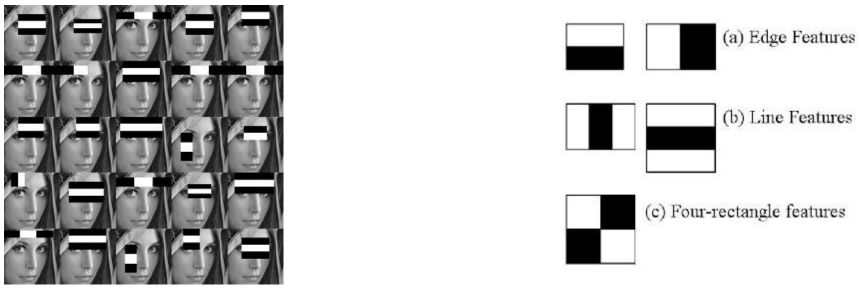
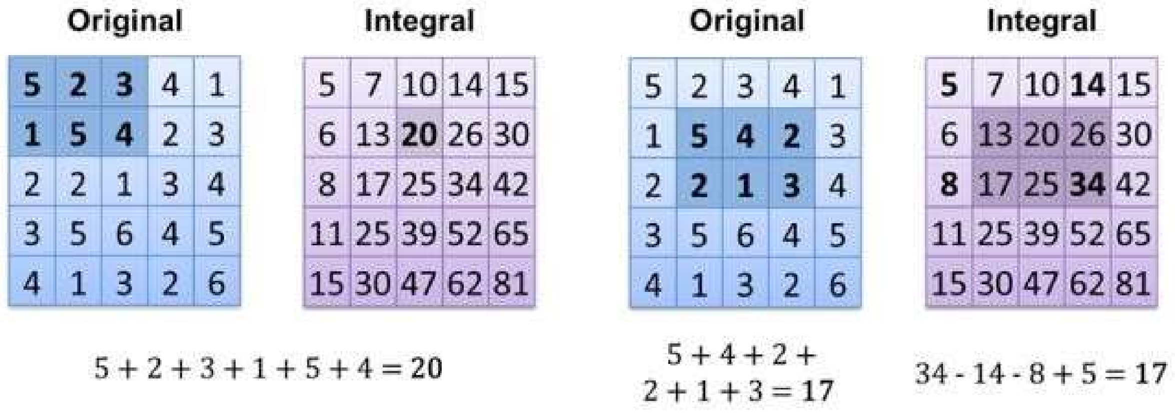


Fig: Haar Feature

Viola-Jones algorithm analyses a given image using Haar features consisting of multiple rectangles (Mekha Joseph et al., 2016).
In the fig 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).



Original

5	**2**	**3**	4	1
1	**5**	**4**	2	3
2	2	1	3	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	14	15
6	13	**20**	26	30
8	17	25	34	42
11	25	39	52	65
15	30	47	62	81

5 + 2 + 3 + 1 + 5 + 4 = 20

Original

5	2	3	4	1
1	**5**	**4**	**2**	3
2	**2**	**1**	**3**	4
3	5	6	4	5
4	1	3	2	6

Integral

5	7	10	**14**	15
6	13	20	26	30
8	17	25	**34**	42
11	25	39	52	65
15	30	47	62	81

5 + 4 + 2 +
2 + 1 + 3 = 17

34 - 14 - 8 + 5 = 17

Fig: Integral of Image

The value of integrating image in a specific location is thesum 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 arecumulative.
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) [11]. 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.

Local Binary Pattern Histogram:

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.

It was first described in 1994(LBP) and has since been found to be a powerful feature for texture classification. It has further determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images witha simple data vector.

LBPH algorithm work step by step:

LBPH algorithm work in 5 steps.

1. **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 horizontaldirection. 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 verticaldirection. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
2. **Training the Algorithm:** First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. Weneed to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.
3. **Applying the LBP operation:** The first computational stepof 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 parametersradius and neighbors.

The image below shows this procedure:

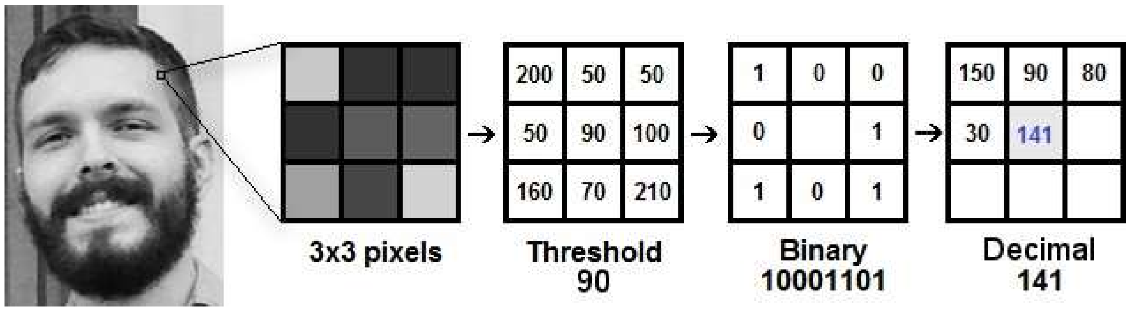


Fig: LBP Operation

Based on the image above, let’s 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 containingthe intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix tobe used as the threshold.
- This value will be used to define the new values from the8 neighbors.
- For each neighbor of the central value (threshold), weset 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 concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, we convert this binary value to a decimal valueand 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 anew image which represents better the characteristics of the original image.

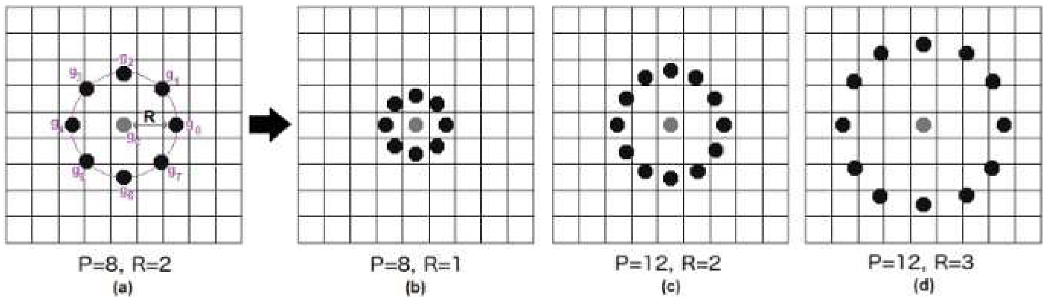


Fig: The LBP operation Radius Change

It can be done by using bilinear 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.

4. **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:

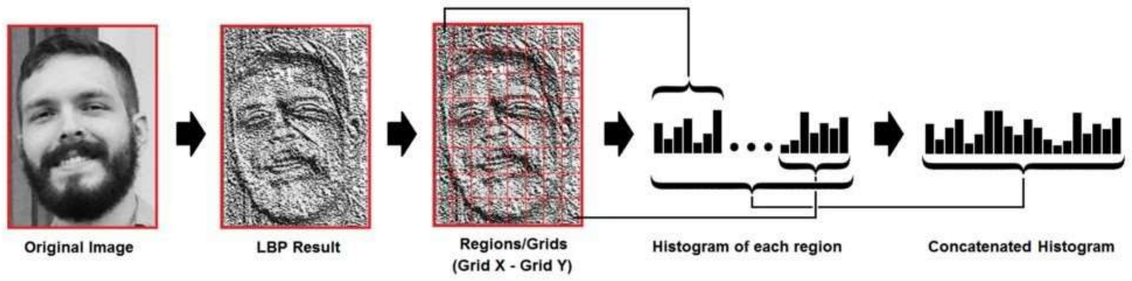


Fig: Extracting The Histogram

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.

5. **Performing the face recognition:** In this step, the algorithm is already trained. Each histogram createdis used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.
   - So to find the image that matches the input image wejust need to compare two histograms and return the image with the closest histogram.
   - We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the **Euclidean distance** (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

   - So the algorithm output is the ID from the image withthe closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
   - We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Chapter 3

Design Requirements

3.1. Design Requirements

We used some tools to build the HFR system. Without the help of these tools it would not be possible to make it done. Here we will discuss about the most important one.

3.1.1. Software Implementation

1. **OpenCV:** We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given bellow:
   - **Derivation**: Gradient / laplacian computing, contours delimitation
   - **Hough transforms:** lines, segments, circles, and geometrical shapes detection
   - **Histograms**: computing, equalization, and object localization with back projection algorithm
   - **Segmentation**: thresholding, distance transform,foreground / background detection, watershed segmentation
   - **Filtering**: linear and nonlinear filters, morphological operations
   - **Cascade detectors**: detection of face, eye, car plates
   - **Interest points**: detection and matching
   - **Video processing:** optical flow, background subtraction, camshaft (object tracking)
   - **Photography**: panoramas realization, high definition imaging (HDR), image inpainting.

2. **Python IDE:** There are lots of IDEs for python. Some of themare PyCharm, Thonny, Ninja, Spyder etc. Pycharm and Spyder both are very excellent and free but we used Pycharm as it feature- rich than spyder.

Chapter 4

Methodology

4.1. Introduction

Face detection involves separating image windows into two classes; one containing faces (turning the background(clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise.An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any faceor faces within that image as some bounding box with (x, y, width, height).After taking the picture the system will compare the equality of the pictures in its database and give the most related result

4.2. Flow Chart

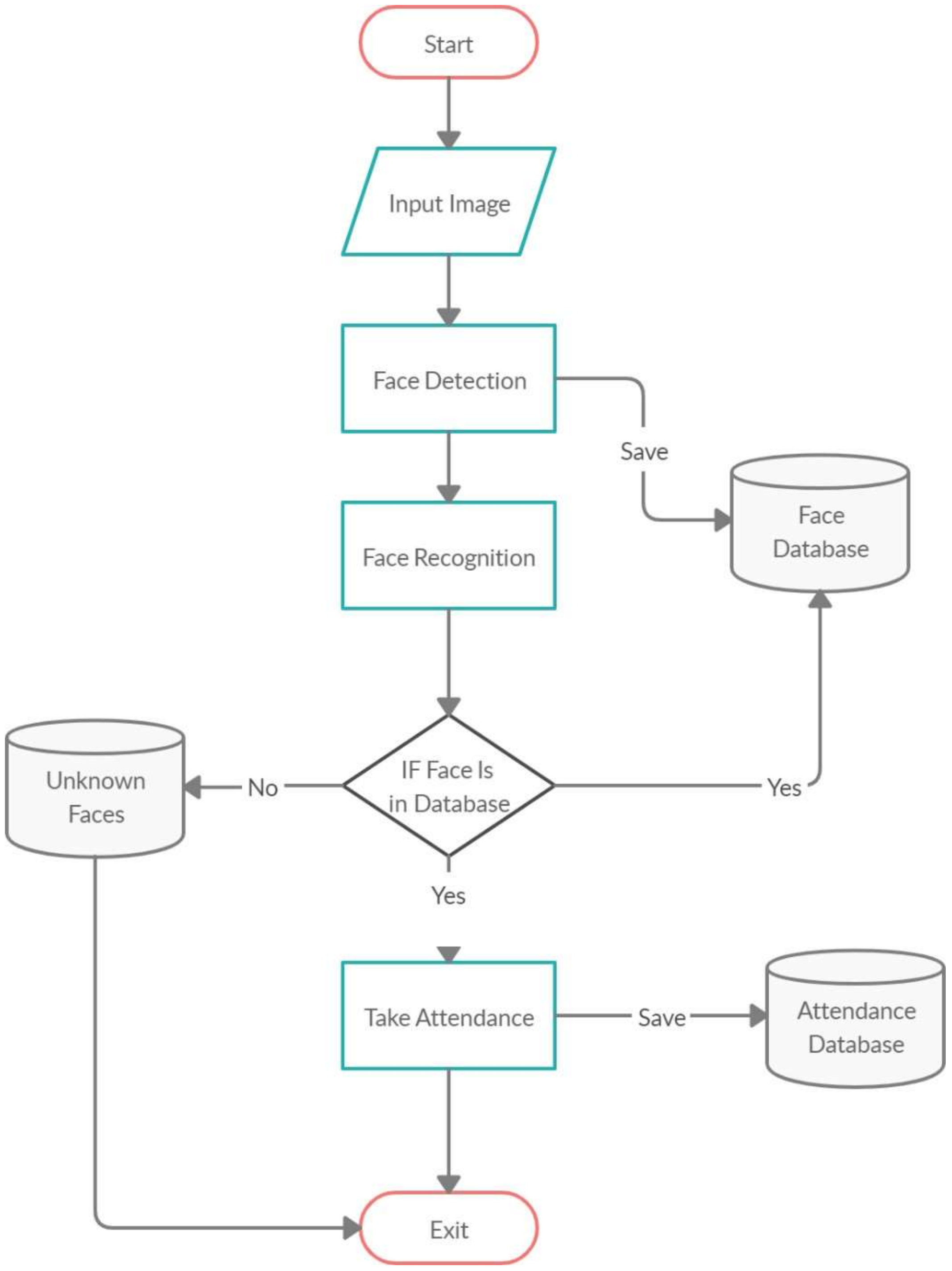


Fig: Flow Chart

4.3. Model Implementation

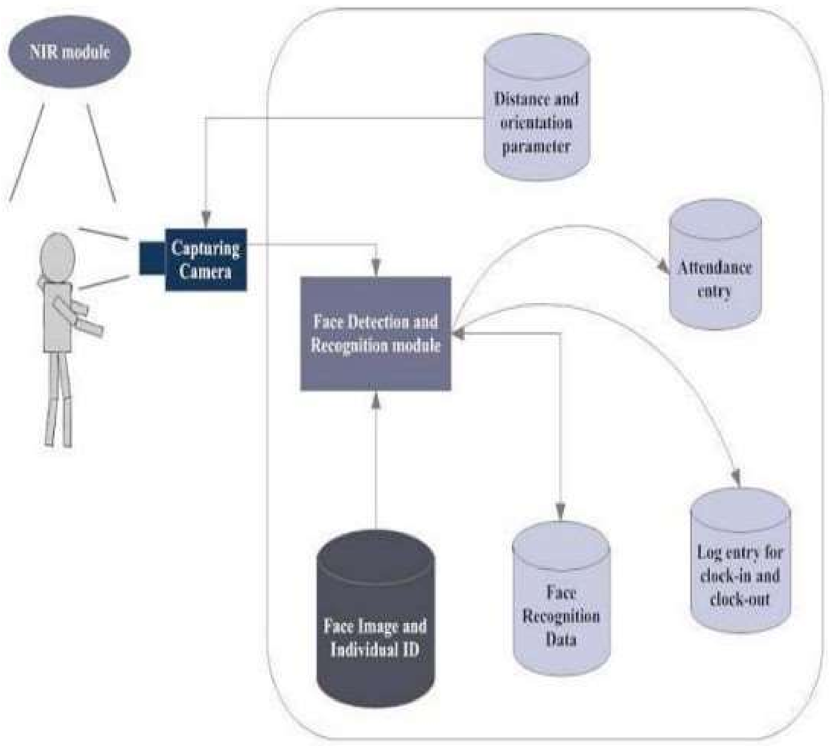

NIR module
Distance and orientation parameter
Capturing Camera
Attendance entry
Face Detection and Recognition module
Log entry for clock-in and clock-out
Face Recognition Data
Face Image and Individual ID

Fig: Model Implement

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the Eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The Eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbor.

4.4. Summary

In this long yet useful chapter we managed to cover the entire structure of how the system has been developed and how it functions to give the best outcome.

Chapter 5

Results & discussion

5.1. Introduction

This chapter of the report contains the results that weachieved throughout the course of using this system.

5.2. Experimental Results

The step of the experiments process are given below:

Face Detection:

Start capturing images through web camera of the client side: Begin:

- Pre-process the captured image and extract face image
- Calculate the Eigen value of the captured face imageand compared with Eigen values of existing faces in the database.
- If Eigen value does not matched with existing once save the new face image information to the face database (xmlfile).
- If Eigen value matched with existing one then recognition step will done.

End

Face Recognition:

Using PCA algorithm the following steps would be followed infor face recognition:

Begin:

- Find the face information of matched face image in fromthe database.
- Update the log table with corresponding face image and system time that makes completion of attendance for an individual students.

End

This section presents the results of the experiment conductedto capture the face into a grey scale image of 50x50 pixels.

Test data	Expected Result	Observed Result	Pass/ Fail
OpenCAM_CB()	Connects with the installed camera and starts playing.	Camera started.	pass
LoadHaar Classifier()	Loads the HaarClassifier Cascade files for frontal face	Gets ready for Extraction.	Pass
ExtractFace()	Initiates the Paul-Viola Face extracting Frame work.	Face extracted	Pass
Learn()	Start the PCA Algorithm	Updates the facedata.xml	Pass
Recognize()	It compares the input face with the saved faces.	Nearest face	Pass

Here is our data set sample.

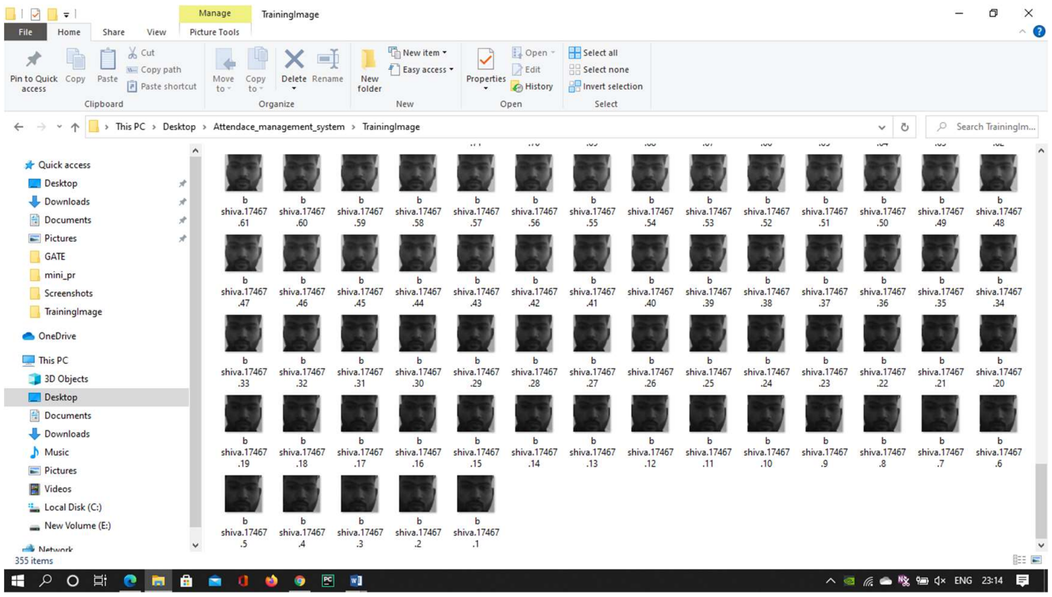
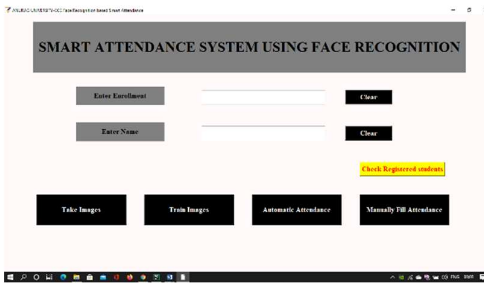
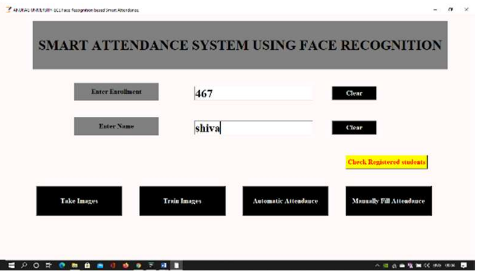

TrainingImage
This PC > Desktop > Attendace_management_system > TrainingImage
355 items

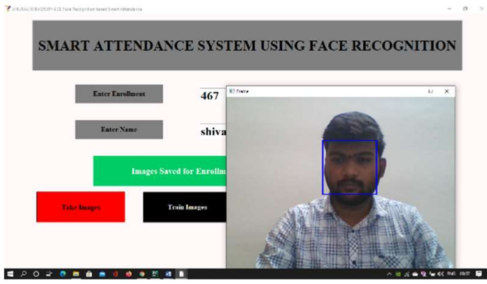
Fig: Dataset sample

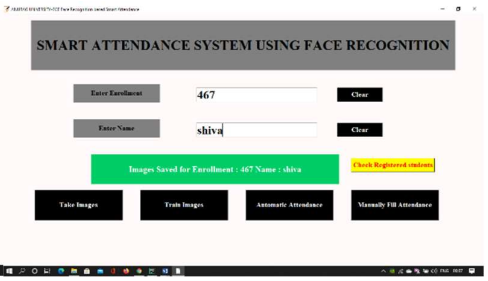
Sample Images:



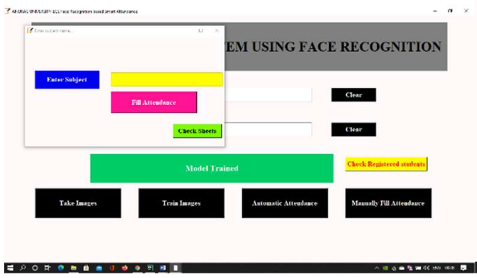


Enter Enrollment: 467
Enter Name: shiva

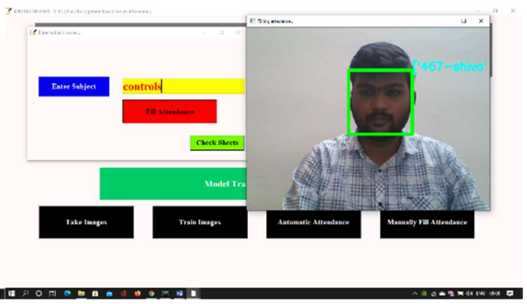




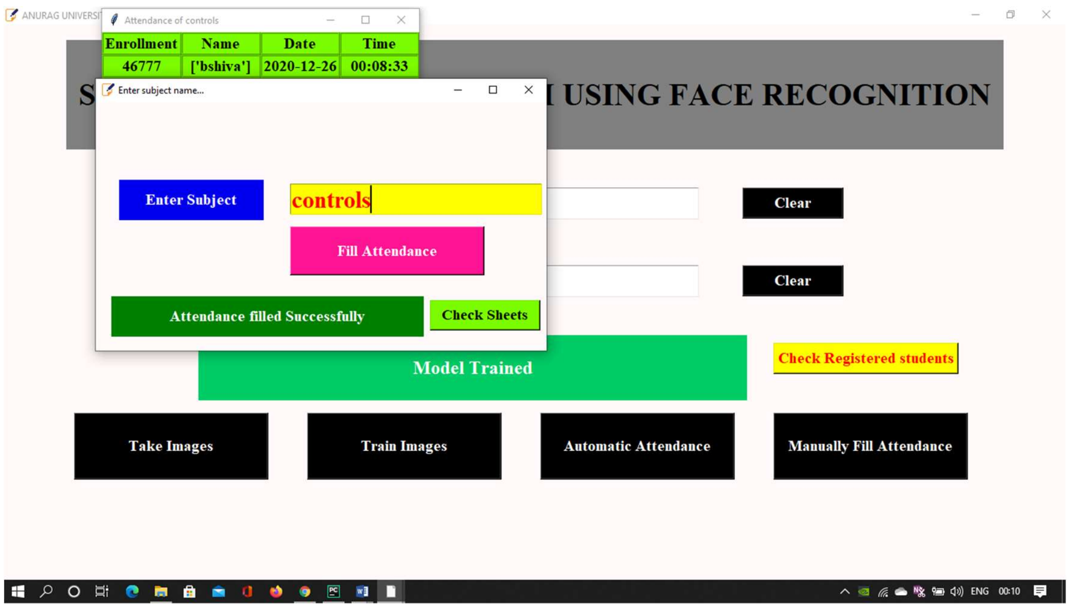
Enter Enrollment: 467
Enter Name: shiva
Images Saved for Enrollment : 467 Name : shiva



Model Trained



Enter Subject: controls



Enrollment	Name	Date	Time
46777	['bshiva']	2020-12-26	00:08:33

Enter Subject: controls
Attendance filled Successfully
Model Trained

Results Achieved

From initiation through conclusion of developing this systemthe following results has been achieved. They are as follows:

- The system can be administered by a non-IT technician.
- The system is market ready for commercial use.
- The system has the capacity to carry up to a thousand faces to recognize
- The system can serve as much people as they want within an organization.

5.3. Summary

This chapter has covered the different types of results thatwe have managed to obtain throughout the course of using this system.

Chapter 6

Future Scope

6.1. Introduction

This chapter discusses the future scope or the implementation of this robot. To incerease the scope of this device we can add some new features. As technology is becoming more advance it will be mendatory to change the sturctute some day with better replacement and sometimes based on customer requirements.

6.2. Future Scope of Work

There are so many future scope on this project. Some of them are

- Can improve security
- Can use Neural Network for high accuracy
- Can used in big factory or employee attendance
- Can build on fully web base system.

6.3. Summary

This chapter has described the possible future applications of the design. But there are a lot of possibilities with the designed device. The device may need some research for different applications, though the principle of the designed system will remain as it is.

Chapter 7

References

7. References

- Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. “Face description with local binary patterns: Application to face recognition.” IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037–2041.

- Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.” IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971–987.

- Ahonen, Timo, Abdenour Hadid, and Matti Pietikäinen. “Face recognition with local binary patterns.” Computer vision-eccv 2004 (2004): 469–481.

- LBPH OpenCV: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms

- Local Binary Patterns: http://www.scholarpedia.org/article/Local_Binary_Patterns

Appendix

All our code is written in Python language. First here is our project directory structure and files.

AMS.py

```
4     import tkinter as tk
      from tkinter import *
      import cv2
      import csv
      import os
      import numpy as np
      from PIL import Image,ImageTk
      import pandas as pd
      import datetime
      import time

      #####Window is our Main frame of system
      window = tk.Tk()
      window.title("ANURAG UNIVERSITY-ECE Face Recognition based Smart
      Attendance")

      window.geometry('1280x720')
      window.configure(background='snow')
      top = Tk()

      ####GUI for manually fill attendance

      def manually_fill():
          global sb
          sb = tk.Tk()
          sb.iconbitmap('AMS.ico')
          sb.title("Enter subject name...")
          sb.geometry('580x320')
          sb.configure(background='snow')

          def err_screen_for_subject():

              def ec_delete():
                  ec.destroy()
              global ec
              ec = tk.Tk()
              ec.geometry('300x100')
              ec.iconbitmap('AMS.ico')
              ec.title('Warning!!')
              ec.configure(background='snow')
              Label(ec, text='Please enter your subject name!!!', fg='red', bg='white',
     font=('times', 16, ' bold ')).pack()
              Button(ec, text='OK', command=ec_delete, fg="black", bg="lawn green",
      width=9, height=1, activebackground="Red",
                     font=('times', 15, ' bold ')).place(x=90, y=50)

          def fill_attendance():
              ts = time.time()
              Date = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
              timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
```

```
        Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        Hour, Minute, Second = timeStamp.split(":")
        ####Creatting csv of attendance

        ##Create table for Attendance
        date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
        global subb
        subb=SUB_ENTRY.get()
        DB_table_name = str(subb + "_" + Date + "_Time_" + Hour + "_" + Minute + "_"
+ Second)

        import pymysql.connections

        ###Connect to the database
        try:
            global cursor
            connection = pymysql.connect(host='localhost', user='root', password='',
db='manually_fill_attendance')
            cursor = connection.cursor()
        except Exception as e:
            print(e)

        sql = "CREATE TABLE " + DB_table_name + """
                        (ID INT NOT NULL AUTO_INCREMENT,
                         ENROLLMENT varchar(100) NOT NULL,
                         NAME VARCHAR(50) NOT NULL,
                         DATE VARCHAR(20) NOT NULL,
                         TIME VARCHAR(20) NOT NULL,
                             PRIMARY KEY (ID)
                             );
                        """


        try:
            cursor.execute(sql)  ##for create a table
        except Exception as ex:
            print(ex)  #

        if subb=='':
            err_screen_for_subject()
        else:
            sb.destroy()
            MFW = tk.Tk()
            MFW.iconbitmap('AMS.ico')
            MFW.title("Manually attendance of "+ str(subb))
            MFW.geometry('880x470')
            MFW.configure(background='snow')

            def del_errsc2():
                errsc2.destroy()

            def err_screen1():
                global errsc2
                errsc2 = tk.Tk()
                errsc2.geometry('330x100')
                errsc2.iconbitmap('AMS.ico')
                errsc2.title('Warning!!')
                errsc2.configure(background='snow')
                Label(errsc2, text='Please enter Student & Enrollment!!!', fg='red',
bg='white',
                font=('times', 16, ' bold ')).pack()
            Button(errsc2, text='OK', command=del_errsc2, fg="black", bg="lawn
green", width=9, height=1,
                   activebackground="Red", font=('times', 15, ' bold ')).place(x=90, y=50)

        def testVal(inStr, acttyp):
            if acttyp == '1':  # insert
                if not inStr.isdigit():
                    return False
            return True

        ENR = tk.Label(MFW, text="Enter Enrollment", width=15, height=2,
fg="white", bg="blue2",
                       font=('times', 15, ' bold '))
        ENR.place(x=30, y=100)

        STU_NAME = tk.Label(MFW, text="Enter Student name", width=15, height=2,
fg="white", bg="blue2",
                            font=('times', 15, ' bold '))
        STU_NAME.place(x=30, y=200)

        global ENR_ENTRY
        ENR_ENTRY = tk.Entry(MFW, width=20,validate='key', bg="yellow",
fg="red", font=('times', 23, ' bold '))
        ENR_ENTRY['validatecommand'] = (ENR_ENTRY.register(testVal), '%P',
'%d')
        ENR_ENTRY.place(x=290, y=105)

        def remove_enr():
            ENR_ENTRY.delete(first=0, last=22)

        STUDENT_ENTRY = tk.Entry(MFW, width=20, bg="yellow", fg="red",
font=('times', 23, ' bold '))
        STUDENT_ENTRY.place(x=290, y=205)

        def remove_student():
            STUDENT_ENTRY.delete(first=0, last=22)

        ####get important variable
        def enter_data_DB():
            ENROLLMENT = ENR_ENTRY.get()
            STUDENT = STUDENT_ENTRY.get()
            if ENROLLMENT=='':
                err_screen1()
            elif STUDENT=='':
                err_screen1()
            else:
                time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                Hour, Minute, Second = time.split(":")
                Insert_data = "INSERT INTO " + DB_table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
                VALUES = (str(ENROLLMENT), str(STUDENT), str(Date), str(time))
                try:
                    cursor.execute(Insert_data, VALUES)
                except Exception as e:
                    print(e)
                ENR_ENTRY.delete(first=0, last=22)
                STUDENT_ENTRY.delete(first=0, last=22)

        def create_csv():
            import csv
            cursor.execute("select * from " + DB_table_name + ";")
csv_name=r'C:\Users\india\Desktop\Attendace_management_system/Attendance/Ma
nually Attendance/'+DB_table_name+'.csv'
            with open(csv_name, "w") as csv_file:
                csv_writer = csv.writer(csv_file)
                csv_writer.writerow([i[0] for i in cursor.description])  # write headers
                csv_writer.writerows(cursor)
                O="CSV created Successfully"
                Notifi.configure(text=O, bg="Green", fg="white", width=33, font=('times',
19, 'bold'))
                Notifi.place(x=180, y=380)
            import csv
            import tkinter
            root = tkinter.Tk()
            root.title("Attendance of " + subb)
            root.configure(background='snow')
            with open(csv_name, newline="") as file:
                reader = csv.reader(file)
                r = 0

                for col in reader:
                    c = 0
                    for row in col:
                        # i've added some styling
                        label = tkinter.Label(root, width=13, height=1, fg="black",
font=('times', 13, ' bold '),
                                                  bg="lawn green", text=row, relief=tkinter.RIDGE)
                        label.grid(row=r, column=c)
                        c += 1
                    r += 1
            root.mainloop()

        Notifi = tk.Label(MFW, text="CSV created Successfully", bg="Green",
fg="white", width=33,
                            height=2, font=('times', 19, 'bold'))


        c1ear_enroll = tk.Button(MFW, text="Clear", command=remove_enr,
fg="white", bg="black", width=10,
                                  height=1,
                                  activebackground="Red", font=('times', 15, ' bold '))
        c1ear_enroll.place(x=690, y=100)

        c1ear_student = tk.Button(MFW, text="Clear", command=remove_student,
fg="white", bg="black", width=10,
                                   height=1,
                                   activebackground="Red", font=('times', 15, ' bold '))
        c1ear_student.place(x=690, y=200)

        DATA_SUB = tk.Button(MFW, text="Enter Data",command=enter_data_DB,
fg="black", bg="lime green", width=20,
                            height=2,
                            activebackground="Red", font=('times', 15, ' bold '))
        DATA_SUB.place(x=170, y=300)

        MAKE_CSV = tk.Button(MFW, text="Convert to CSV",command=create_csv,
fg="black", bg="red", width=20,
                            height=2,
                    activebackground="Red", font=('times', 15, ' bold '))
        MAKE_CSV.place(x=570, y=300)

        def attf():
            import subprocess
            subprocess.Popen(r'explorer
/select,"C:\Users\india\Desktop\Attendace_management_system\Attendance\Manuall
y Attendance\-------Check atttendance-------"')

        attf = tk.Button(MFW,  text="Check Sheets",command=attf,fg="black"
,bg="lawn green"  ,width=12  ,height=1 ,activebackground = "Red" ,font=('times', 14, '
bold '))
        attf.place(x=730, y=410)

        MFW.mainloop()


    SUB = tk.Label(sb, text="Enter Subject", width=15, height=2, fg="white",
bg="blue2", font=('times', 15, ' bold '))
    SUB.place(x=30, y=100)

    global SUB_ENTRY

    SUB_ENTRY = tk.Entry(sb, width=20, bg="yellow", fg="red", font=('times', 23, '
bold '))
    SUB_ENTRY.place(x=250, y=105)

    fill_manual_attendance = tk.Button(sb, text="Fill
Attendance",command=fill_attendance, fg="white", bg="deep pink", width=20,
height=2,
                    activebackground="Red", font=('times', 15, ' bold '))
    fill_manual_attendance.place(x=250, y=160)
    sb.mainloop()

##For clear textbox
def clear():
    txt.delete(first=0, last=22)

def clear1():
    txt2.delete(first=0, last=22)
def del_sc1():
    sc1.destroy()
def err_screen():
    global sc1
    sc1 = tk.Tk()
    sc1.geometry('300x100')
    sc1.iconbitmap('AMS.ico')
    sc1.title('Warning!!')
    sc1.configure(background='snow')
    Label(sc1,text='Enrollment & Name required!!!',fg='red',bg='white',font=('times',
16, ' bold ')).pack()
    Button(sc1,text='OK',command=del_sc1,fg="black"  ,bg="lawn green"  ,width=9
,height=1, activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)

##Error screen2
def del_sc2():
    sc2.destroy()
def err_screen1():
    global sc2
    sc2 = tk.Tk()
    sc2.geometry('300x100')
    sc2.iconbitmap('AMS.ico')
    sc2.title('Warning!!')
    sc2.configure(background='snow')
    Label(sc2,text='Please enter your subject name!!!',fg='red',bg='white',font=('times',
16, ' bold ')).pack()
    Button(sc2,text='OK',command=del_sc2,fg="black"  ,bg="lawn green"  ,width=9
,height=1, activebackground = "Red" ,font=('times', 15, ' bold ')).place(x=90,y= 50)

###For take images for datasets
def take_img():
    l1 = txt.get()
    l2 = txt2.get()
    if l1 == '':
        err_screen()
    elif l2 == '':
        err_screen()
    else:
        try:
            cam = cv2.VideoCapture(0)
            detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
            Enrollment = txt.get()
            Name = txt2.get()
            sampleNum = 0
            while (True):
                ret, img = cam.read()
                gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
                faces = detector.detectMultiScale(gray, 1.3, 5)
                for (x, y, w, h) in faces:
                    cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)
                    # incrementing sample number
                    sampleNum = sampleNum + 1
                    # saving the captured face in the dataset folder
                    cv2.imwrite("TrainingImage/ " + Name + "." + Enrollment + '.' +
str(sampleNum) + ".jpg",
                                gray[y:y + h, x:x + w])
                    cv2.imshow('Frame', img)
                # wait for 100 miliseconds
                if cv2.waitKey(1) & 0xFF == ord('q'):
                    break
                # break if the sample number is morethan 100
                elif sampleNum > 70:
                    break
            cam.release()
            cv2.destroyAllWindows()
            ts = time.time()
            Date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
            Time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
            row = [Enrollment, Name, Date, Time]
            with open('StudentDetails\StudentDetails.csv', 'a+') as csvFile:
                writer = csv.writer(csvFile, delimiter=',')
                writer.writerow(row)
                csvFile.close()
            res = "Images Saved for Enrollment : " + Enrollment + " Name : " + Name
            Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times',
18, 'bold'))
            Notification.place(x=250, y=400)
        except FileExistsError as F:
            f = 'Student Data already exists'
            Notification.configure(text=f, bg="Red", width=21)
```

```
      Notification.place(x=450, y=400)


###for choose subject and fill attendance
def subjectchoose():
    def Fillattendances():
        sub=tx.get()
        now = time.time()  ###For calculate seconds of video
        future = now + 20
        if time.time() < future:
            if sub == '':
                err_screen1()
            else:
                recognizer = cv2.face.LBPHFaceRecognizer_create()  #
cv2.createLBPHFaceRecognizer()
                try:
                    recognizer.read("TrainingImageLabel\Trainner.yml")
                except:
                    e = 'Model not found,Please train model'
                    Notifica.configure(text=e, bg="red", fg="black", width=33, font=('times',
15, 'bold'))
                    Notifica.place(x=20, y=250)

                harcascadePath = "haarcascade_frontalface_default.xml"
                faceCascade = cv2.CascadeClassifier(harcascadePath)
                df = pd.read_csv("StudentDetails\StudentDetails.csv")
                cam = cv2.VideoCapture(0)
                font = cv2.FONT_HERSHEY_SIMPLEX
                col_names = ['Enrollment', 'Name', 'Date', 'Time']
                attendance = pd.DataFrame(columns=col_names)
                while True:
                    ret, im = cam.read()
                    gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
                    faces = faceCascade.detectMultiScale(gray, 1.2, 5)
                    for (x, y, w, h) in faces:
                        global Id

                        Id, conf = recognizer.predict(gray[y:y + h, x:x + w])
                        if (conf <70):
                            print(conf)
                            global Subject
                            global aa
                            global date
                            global timeStamp
                            Subject = tx.get()
                            ts = time.time()
                            date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                            timeStamp =
datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                            aa = df.loc[df['Enrollment'] == Id]['Name'].values
                            global tt
                            tt = str(Id) + "-" + aa
                            En = '15624031' + str(Id)
                            attendance.loc[len(attendance)] = [Id, aa, date, timeStamp]
                            cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
                            cv2.putText(im, str(tt), (x + h, y), font, 1, (255, 255, 0,), 4)

                        else:
                            Id = 'Unknown'
                            tt = str(Id)
                            cv2.rectangle(im, (x, y), (x + w, y + h), (0, 25, 255), 7)
                            cv2.putText(im, str(tt), (x + h, y), font, 1, (0, 25, 255), 4)
                    if time.time() > future:
                        break

                    attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
                    cv2.imshow('Filling attedance..', im)
                    key = cv2.waitKey(30) & 0xff
                    if key == 27:
                        break

                ts = time.time()
                date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
                timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
                Hour, Minute, Second = timeStamp.split(":")
                fileName = "Attendance/" + Subject + "_" + date + "_" + Hour + "-" +
Minute + "-" + Second + ".csv"
                attendance = attendance.drop_duplicates(['Enrollment'], keep='first')
                print(attendance)
                attendance.to_csv(fileName, index=False)

                ##Create table for Attendance
                date_for_DB = datetime.datetime.fromtimestamp(ts).strftime('%Y_%m_%d')
                DB_Table_name = str( Subject + "_" + date_for_DB + "_Time_" + Hour +
"_" + Minute + "_" + Second)
                import pymysql.connections

                ###Connect to the database
                try:
                    global cursor
                    connection = pymysql.connect(host='localhost', user='root', password='',
db='Face_reco_fill')
                    cursor = connection.cursor()
                except Exception as e:
                    print(e)

                sql = "CREATE TABLE " + DB_Table_name + """
                (ID INT NOT NULL AUTO_INCREMENT,
                 ENROLLMENT varchar(100) NOT NULL,
                 NAME VARCHAR(50) NOT NULL,
                 DATE VARCHAR(20) NOT NULL,
                 TIME VARCHAR(20) NOT NULL,
                     PRIMARY KEY (ID)
                     );
                """
                ####Now enter attendance in Database
                insert_data =  "INSERT INTO " + DB_Table_name + "
(ID,ENROLLMENT,NAME,DATE,TIME) VALUES (0, %s, %s, %s,%s)"
                VALUES = (str(Id), str(aa), str(date), str(timeStamp))
                try:
                    cursor.execute(sql)  ##for create a table
                    cursor.execute(insert_data, VALUES)##For insert data into table
                except Exception as ex:
                    print(ex)  #

                M = 'Attendance filled Successfully'
                Notifica.configure(text=M, bg="Green", fg="white", width=33, font=('times',
15, 'bold'))
                Notifica.place(x=20, y=250)
                cam.release()
                cv2.destroyAllWindows()

                import csv
                import tkinter
                root = tkinter.Tk()
                root.title("Attendance of " + Subject)
                root.configure(background='snow')
                cs = r'C:\Users\india\Desktop\Attendace_management_system/' + fileName
                with open(cs, newline="") as file:
                    reader = csv.reader(file)
                    r = 0

                    for col in reader:
                        c = 0
                        for row in col:
                            # i've added some styling
                            label = tkinter.Label(root, width=8, height=1, fg="black",
font=('times', 15, ' bold '),
                                                  bg="lawn green", text=row, relief=tkinter.RIDGE)
                            label.grid(row=r, column=c)
                            c += 1
                        r += 1
                root.mainloop()
                print(attendance)

    ###windo is frame for subject chooser
    windo = tk.Tk()
    windo.iconbitmap('AMS.ico')
    windo.title("Enter subject name...")
    windo.geometry('580x320')
    windo.configure(background='snow')
    Notifica = tk.Label(windo, text="Attendance filled Successfully", bg="Green",
fg="white", width=33,
                        height=2, font=('times', 15, 'bold'))

    def Attf():
        import subprocess
        subprocess.Popen(r'explorer
/select,"C:\Users\india\Desktop\Attendace_management_system\Attendance\-------
Check atttendance-------"')

    attf = tk.Button(windo,  text="Check Sheets",command=Attf,fg="black"  ,bg="lawn
green"  ,width=12  ,height=1 ,activebackground = "Red" ,font=('times', 14, ' bold '))
    attf.place(x=430, y=255)

    sub = tk.Label(windo, text="Enter Subject", width=15, height=2, fg="white",
bg="blue2", font=('times', 15, ' bold '))
    sub.place(x=30, y=100)

    tx = tk.Entry(windo, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    tx.place(x=250, y=105)

    fill_a = tk.Button(windo, text="Fill Attendance",
fg="white",command=Fillattendances, bg="deep pink", width=20, height=2,
                      activebackground="Red", font=('times', 15, ' bold '))
    fill_a.place(x=250, y=160)
    windo.mainloop()

def admin_panel():
    win = tk.Tk()
    win.iconbitmap('AMS.ico')
    win.title("LogIn")
    win.geometry('880x420')
    win.configure(background='snow')

    def log_in():
        username = un_entr.get()
        password = pw_entr.get()

        if username == 'AUECEB1' :
            if password == '12345':
                win.destroy()
                import csv
                import tkinter
                root = tkinter.Tk()
                root.title("Student Details")
                root.configure(background='snow')

                cs =
r'C:\Users\india\Desktop\Attendace_management_system/StudentDetails/StudentDet
ails.csv'
                with open(cs, newline="") as file:
                    reader = csv.reader(file)
                    r = 0

                    for col in reader:
                        c = 0
                        for row in col:
                            # i've added some styling
                            label = tkinter.Label(root, width=8, height=1, fg="black",
font=('times', 15, ' bold '),
                                                  bg="lawn green", text=row, relief=tkinter.RIDGE)
                            label.grid(row=r, column=c)
                            c += 1
                        r += 1
                root.mainloop()
            else:
                valid = 'Incorrect ID or Password'
                Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19,
'bold'))
                Nt.place(x=120, y=350)

        else:
            valid ='Incorrect ID or Password'
            Nt.configure(text=valid, bg="red", fg="black", width=38, font=('times', 19,
'bold'))
            Nt.place(x=120, y=350)


    Nt = tk.Label(win, text="Attendance filled Successfully", bg="Green", fg="white",
width=40,
                  height=2, font=('times', 19, 'bold'))
    # Nt.place(x=120, y=350)

    un = tk.Label(win, text="Enter username", width=15, height=2, fg="white",
bg="blue2",
                   font=('times', 15, ' bold '))
    un.place(x=30, y=50)
    pw = tk.Label(win, text="Enter password", width=15, height=2, fg="white",
bg="blue2",
                  font=('times', 15, ' bold '))
    pw.place(x=30, y=150)

    def c00():
        un_entr.delete(first=0, last=22)

    un_entr = tk.Entry(win, width=20, bg="yellow", fg="red", font=('times', 23, ' bold '))
    un_entr.place(x=290, y=55)

    def c11():
        pw_entr.delete(first=0, last=22)

    pw_entr = tk.Entry(win, width=20,show="*", bg="yellow", fg="red", font=('times',
23, ' bold '))
    pw_entr.place(x=290, y=155)

    c0 = tk.Button(win, text="Clear", command=c00, fg="black", bg="deep pink",
width=10, height=1,
                            activebackground="Red", font=('times', 15, ' bold '))
    c0.place(x=690, y=55)

    c1 = tk.Button(win, text="Clear", command=c11, fg="black", bg="deep pink",
width=10, height=1,
                   activebackground="Red", font=('times', 15, ' bold '))
    c1.place(x=690, y=155)

    Login = tk.Button(win, text="LogIn", fg="black", bg="lime green", width=20,
                       height=2,
                       activebackground="Red",command=log_in, font=('times', 15, ' bold '))
    Login.place(x=290, y=250)
    win.mainloop()


###For train the model
def trainimg():
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    global detector
    detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    try:
        global faces,Id
        faces, Id = getImagesAndLabels("TrainingImage")
    except Exception as e:
        l='please make "TrainingImage" folder & put Images'
        Notification.configure(text=l, bg="SpringGreen3", width=50, font=('times', 18,
'bold'))
        Notification.place(x=350, y=400)

    recognizer.train(faces, np.array(Id))
    try:
        recognizer.save("TrainingImageLabel\Trainner.yml")
    except Exception as e:
        q='Please make "TrainingImageLabel" folder'
        Notification.configure(text=q, bg="SpringGreen3", width=50, font=('times', 18,
'bold'))
        Notification.place(x=350, y=400)

    res = "Model Trained"  # +",".join(str(f) for f in Id)
    Notification.configure(text=res, bg="SpringGreen3", width=50, font=('times', 18,
'bold'))
    Notification.place(x=250, y=400)

def getImagesAndLabels(path):
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    # create empth face list
    faceSamples = []
    # create empty ID list
    Ids = []
    # now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        # loading the image and converting it to gray scale
        pilImage = Image.open(imagePath).convert('L')
        # Now we are converting the PIL image into numpy array
        imageNp = np.array(pilImage, 'uint8')
        # getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces = detector.detectMultiScale(imageNp)
        # If a face is there then append that in the list as well as Id of it
        for (x, y, w, h) in faces:
            faceSamples.append(imageNp[y:y + h, x:x + w])
            Ids.append(Id)
    return faceSamples, Ids

window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
window.iconbitmap('AMS.ico')

def on_closing():
    from tkinter import messagebox
    if messagebox.askokcancel("Quit", "Do you want to quit?"):
        window.destroy()
window.protocol("WM_DELETE_WINDOW", on_closing)

message = tk.Label(window, text="SMART ATTENDANCE SYSTEM USING FACE
RECOGNITION", bg="grey", fg="black", width=50,
                   height=3, font=('times', 30, ' bold '))

message.place(x=80, y=20)

Notification = tk.Label(window, text="All things good", bg="Green", fg="white",
width=15,
                      height=3, font=('times', 17, 'bold'))

lbl = tk.Label(window, text="Enter Enrollment", width=20, height=2, fg="black",
bg="grey", font=('times', 15, ' bold '))
lbl.place(x=200, y=200)

def testVal(inStr,acttyp):
    if acttyp == '1': #insert
        if not inStr.isdigit():
            return False
    return True

txt = tk.Entry(window, validate="key", width=20, bg="white", fg="black",
font=('times', 25, ' bold '))
txt['validatecommand'] = (txt.register(testVal),'%P','%d')
txt.place(x=550, y=210)
```

```
lbl2 = tk.Label(window, text="Enter Name", width=20, fg="black", bg="grey",
height=2, font=('times', 15, ' bold '))
lbl2.place(x=200, y=300)

txt2 = tk.Entry(window, width=20, bg="white", fg="black", font=('times', 25, ' bold '))
txt2.place(x=550, y=310)

clearButton = tk.Button(window, text="Clear",command=clear,fg="white"
,bg="black"  ,width=10  ,height=1 ,activebackground = "Red" ,font=('times', 15, ' bold
'))
clearButton.place(x=950, y=210)

clearButton1 = tk.Button(window, text="Clear",command=clear1,fg="white"
,bg="black"  ,width=10 ,height=1, activebackground = "Red" ,font=('times', 15, ' bold
'))
clearButton1.place(x=950, y=310)

AP = tk.Button(window, text="Check Registered
students",command=admin_panel,fg="red"  ,bg="yellow"  ,width=19 ,height=1,
activebackground = "Red" ,font=('times', 15, ' bold '))
AP.place(x=990, y=410)

takeImg = tk.Button(window, text="Take Images",command=take_img,fg="white"
,bg="black"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold
'))
takeImg.place(x=90, y=500)

trainImg = tk.Button(window, text="Train Images",fg="white",command=trainimg
,bg="black"  ,width=20  ,height=3, activebackground = "Red" ,font=('times', 15, ' bold
'))
trainImg.place(x=390, y=500)

FA = tk.Button(window, text="Automatic
Attendance",fg="white",command=subjectchoose  ,bg="black"  ,width=20  ,height=3,
activebackground = "Red" ,font=('times', 15, ' bold '))
FA.place(x=690, y=500)

quitWindow = tk.Button(window, text="Manually Fill Attendance",
command=manually_fill  ,fg="white"  ,bg="black"  ,width=20  ,height=3,
activebackground = "Red" ,font=('times', 15, ' bold '))
quitWindow.place(x=990, y=500)

window.mainloop()
```

training.py

```
import cv2,os
import numpy as np
from PIL import Image
#
# recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer=cv2.face.createFisherFaceRecognizer_create()
detector= cv2.CascadeClassifier("haarcascade_frontalface_default.xml")

def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #create empth face list
    faceSamples=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
        #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image

        Id = int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces=detector.detectMultiScale(imageNp)
        #If a face is there then append that in the list as well as Id of it
        for (x,y,w,h) in faces:
            faceSamples.append(imageNp[y:y+h,x:x+w])
            Ids.append(Id)
    return faceSamples,Ids

faces,Ids = getImagesAndLabels('TrainingImage')
recognizer.train(faces, np.array(Ids))
recognizer.save('TrainingImageLabel/trainner.yml')
```

testing.py

```
import cv2
import numpy as np

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('TrainingImageLabel/trainner.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath)
font = cv2.FONT_HERSHEY_SIMPLEX

cam = cv2.VideoCapture(0)
while True:
    ret, im =cam.read()
    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray, 1.2,5)
    for(x,y,w,h) in faces:
        Id, conf = recognizer.predict(gray[y:y+h,x:x+w])

        # # else:
        # #     Id="Unknown"
        # cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)
        cv2.rectangle(im, (x, y), (x + w, y + h), (0, 260, 0), 7)
        cv2.putText(im, str(Id), (x,y-40),font, 2, (255,255,255), 3)

        # cv2.putText(im, str(Id), (x + h, y), font, 1, (0, 260, 0), 2)
    cv2.imshow('im',im)
    if cv2.waitKey(10) & 0xFF==ord('q'):
        break
cam.release()
cv2.destroyAllWindows()
```