Project Report
on

FACE RECOGNITION ATTENDANCE MANAGEMENT SYSTEM

Submitted by

Namagondla Tharuni-R171034

Under the guidance of

S.K.Shabana
Assistant Professor

Department of Computer Science and Engineering

Rajiv Gandhi University of Knowledge and Technologies(RGUKT), R.K.Valley,Kadapa,Andhra Pradesh

Declaration

We,hereby declares that this report entitled **“Face Recognition Attendance Management System”**, submitted by me under the guidance and supervision of **S.K.Shabana** is a bonafide work . We also declare that it has not been submitted previously in part or in full to this University or other University or Institution for the award of any degree or diploma.

We will be solely responsible if any kind of plagiarism is found.

Place : RK Valley

Namagondla Tharuni-R171034

Rajiv Gandhi University of Knowledge Technologies

RK Valley,Kadapa(Dist),Andhra Pradesh, 516330

CERTIFICATE

This is to certify that the project work titled "**FACE RECOGNITION ATTENDANCE MANAGEMENT SYSTEM**" is a bonafied project work submitted by **NAMAGONDLA THARUNI** with id **R171034** in the department of **COMPUTER SCIENCE AND ENGINEERING** in partial fulfillment of requirements for the award of degree of **Bachelor of Technology** for the year 2022-2023 carried out the work under the supervision.

GUIDE
(S.K.SHABANA)

Head Of The Department
(N SATYANANDARAM)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful companies of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success.

I am extremely grateful to our respected Directoe, **Prof. K.SANDHYA RANI** for fosteringan excellent academic climate in our institution.

I also express my sincere gratitude to our respected Head Of The Department **Mr. N.SATYANANDARAM** for his encouragement,overall guidance in viewing this project a good asset and effort in bringing out this project.

I would like to convey my sincere thanks to our guide at college **Ms. S.K.Shabana** for her guidance,encouragement,co-operation and kindess during the entire duration of the course and academics.

My sincere thanks to all the members who helped me directly and indirectly in the completion of project work.I express my profound gratitude to all our friends and family members for their encouragement.

INDEX

ABSTRACT

This Project involves building an attendance system which utilizes facial recognition to mark the presence of students.It covers areas such as facial detection,alignment and recognition.To cater to various use cases of the system such as registration of new users ,addition of photos to the training dataset,viewing attendance reports etc..It can be used in corporate offices, schools,and small organizations where security is essential.The proposed FACE RECOGNITION ATTENDANCE MANAGEMENT SYSTEM will capture the image and will be compared to the image stored in the database.The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records. Prsent strategies for taking attendance are tedious and time-consuming.Attendance records can be easily manipulated by manual recording.The propsed system builds in python.Face recognition is a part of Machine Learning.This system makes use of Haar-Cascade Classifiers,Local Binary Pattern Histogram(LBPH).Faces are detected and recognized from live streaming video of the classroom. Attendance records will be saved in the csv file.

Introduction

Nowadays Educational institutions are concerned about regularity of student attendance. This is mainly due to student's overall academic performance is affected by his or her attendance in the institute. Mainly there are two conventional methods of marking attendance which are calling out the roll call or by taking student sign on paper. They both were more time consuming and difficult. Hence, there is a requirement of computer-based student attendance management system which will assist the faculty for maintaining attendance record automatically. In this project we have implemented the automated attendance system using PYTHON. We have projected our ideas to implement "Facial Recognition Attendance Management System", in which it imbibes large applications. The application includes face identification, which saves time and eliminates chances of proxy attendance because of the face authorization Hence, this system can be implemented in a field where attendance plays an important role. The system is designed using PYTHON platform. The proposed system uses Haar Cascade classifiers and LBPH algorithm. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images and determines students who are present and absent.

The attendance record is maintained in the csv file which is updated automatically in the system. Problem Statement Attendances of every student are being maintained by every school, college and university. Empirical evidences have shown that there is a significant correlation between students attendances and their academic performances. Therefore, faculty has to maintain proper record for the attendance. The manual attendance record system is not efficient and requires more time to arrange record and to calculate the average attendance of each student. Hence there is a requirement of a system that will solve the problem of student record arrangement and student average attendance calculation. One alternative to make student attendance system automatic is provided by facial recognition. Face recognition can be applied for a wide variety of problems like image and film processing,human-computer interaction, criminal identification etc. This has motivated researchers to develop computational models to identify the faces, which are relatively simple and easy to implement. The existing system represents some face space with higher dimensionality and it is not effective too. The important fact which is considered is that although these face images have high dimensionality, in reality they span very low dimensional space. So instead of considering whole face space with high dimensionality, it is better to consider only a subspace with lower dimensionality to represent this face space.

Purpose

The purpose of this system is to build a attendance system which is based on face recognition techniques. Here face of an individual will be considered for marking attendance. Nowadays, face recognition is gaining more popularity and has been widely used. In this paper, we proposed a system which detects the faces of students from live streaming video of classroom and attendance will be marked if the detected face is found in the database. This new system will consume less time than compared to traditional methods.

Scope

Facial recognition is becoming more prominent in our society. It has made major progress in the field of security. It is a very effective tool that can help low enforcers to recognize criminals and software companies are leveraging the technology to help users access the technology.This technology can be further developed to be used in other avenues such as ATMs, accessing confidential files, or other sensitive materials.This project servers as a foundation for future projects based on facial detection and recognition. Using this system any corporate offices, school and organization.It can replace their traditional way of maintaining attendance of the employees and can also generate their availability(presence) report throughout the month.

Advantages:

- Security
- Low Cost
- Easy to Use
- Good Performace
- Easy Maintenance of attendance records
- Time Saving
- No Data Loss
- Easy and Fast Retrieval

Overall Description

Currently manual student attendance marking technique is often facing a lot issues and a very slow process. Teacher'sor faculty calling names of student from their data sheet and student responding to them. But this existing process becomes very complex in large classes that consists so many students. Many times, students also mark proxies by responding to fake name.

This makes disturbance in class and distracts the students during the exam times. Also, verifying the total students present by counting them after attendance, which takes a lot of time consuming. Apart from calling names attendance sheet is passed around classroom during lectures especially the classes consisting large number of students might find it hard to have attendance sheet being passed around the class.

Douglas Ahlers, Bernie DiDario, Michael Dobson, in 2006 gave the concept of attendance tracking system.This framework consists of identity tags, with wireless communication capabilities, for each attendee and the scanners for detecting the attendee's tags as they enter in that allocated room.

O.A. Idowu and O. Shoewu: Development of Attendance Management System by using Biometrics. Attendance is taken with the help of a finger print device and the records of attendance are stored in the database. Attendance is marked after successful identification.

Arun Katara et al.(2017) mentioned disadvantages of RFID(Ratio Frequency Identification) card system fingerprint system and iris recognition system.RFID card system is implemented duw 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 facer 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 avilable,but it is less accurate compafred to other methods.Hence,face recognition system is suggested to be implemented in the student attendance system.

Difference Between Face Detection and Face Recogniton:

Face detection answers the question,Where is the face? It identifes an object as a "face" and locates it in the input image.Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone.It can ttherefore be seen that face detections output(the detected face) is the input to the face recognizer and the face Recognition's output is the final decision i.e., face known or face unknown.

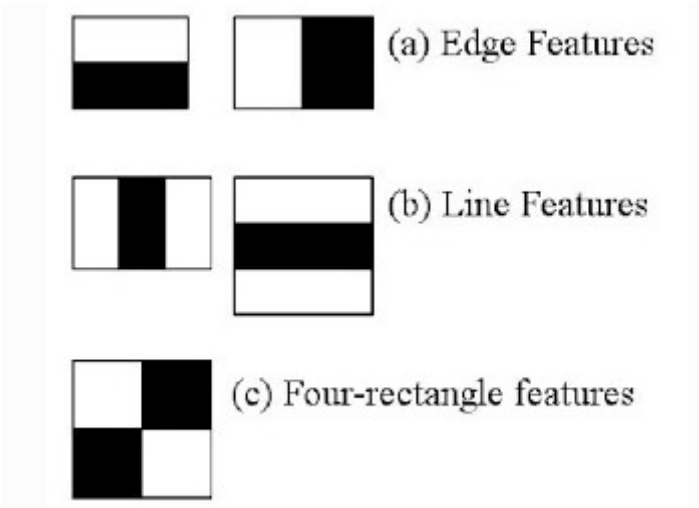
Face Detection using Haar Cascades:

A face Detector has to tell whether an image of arbitrary size contains a human face and if so,where it is.Face detection can be performed based on several clues.Skin color(for faces in color images and videos,motion(for faces in videos),facial/head shape,facial appearance or a combination of these paramters.Most face detection algorithms are appearance based without using ither clues.An input image is scanned at all possible locations and scales by a sub window.Face detection is posed as classifying the pattern in the sub window either as a face or a non-face.The face/nonface classifier is learned from face and nin-face training examples using

statistical learning methods.Most modern algorithms are based on the Viola ,Jones object detection frameword,which is based on Haar Cascades.

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

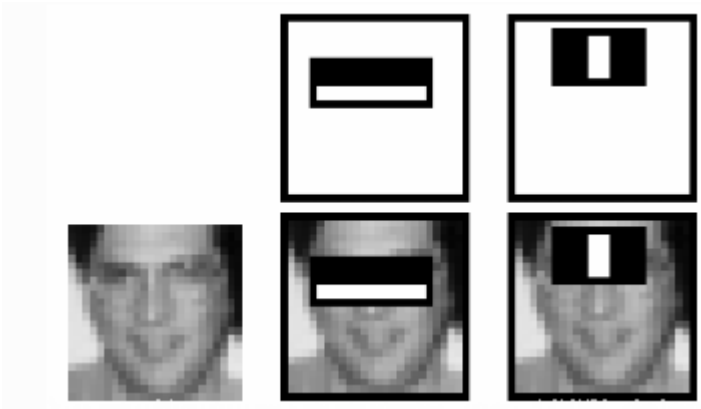
Here we will work with face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.



Now all possible sizes and locations of each kernel is used to calculate plenty of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. Nice, isn’t it? It makes things super-fast.

But among all these features we calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. So how do we select the best features out of 160000+ features? It is achieved by **Adaboost**.

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).



Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. (Imagine a reduction from 160000+ features to 6000 features. That is a big gain).So now you take an image. Take each 24x24 window. Apply 6000 features to it. Check if it is face or not. Wow.. Wow.. Isn't it a little inefficient and time consuming? Yes, it is. Authors have a good solution for that.In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot. Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region. How is the plan !!!Authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. (Two features in the above image is actually obtained as the best two features from Adaboost). According to authors, on an average, 10 features out of 6000+ are evaluated per sub-window.

Haar-cascade Detection in OpenCV:

OpenCV already contains many pre-trained classifiers for face, eyes, smile etc. Those XML files are stored in opencv/data/haarcascades/ folder. Let's create face and eye detector with OpenCV.First we need to load the required XML classifiers. Then load our input image (or video) in grayscale mode.

```
import numpy as np
import cv2

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade = cv2.CascadeClassifier('haarcascade_eye.xml')

img = cv2.imread('sachin.jpg')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can create a ROI for the face and apply eye detection on this ROI (since eyes are always on the face !!!).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
    img = cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = img[y:y+h, x:x+w]
    eyes = eye_cascade.detectMultiScale(roi_gray)
    for (ex,ey,ew,eh) in eyes:
        cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)

cv2.imshow('img',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Local Binary Pattern Histogram(LBPH):

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 been 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 with a simple data vector.

LBPH algorithm work step by step:

It works 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.

-Nighbors: the number of sample points to build the circular local binary pattern.Keep mind the more sample point yu include the higher the computrational cost.It is usually set to 8.

-GridX: the number of cells in the horizontal direction. The more cells,the finer the grid,the higher the dimentionality of the resulting feature vector.It usually set to 8.

-GridY: the number of cells in the horizontal direction. The more cells,the finer the grid,the higher the dimentionality of the resulting feature vector.It usually set to 8.

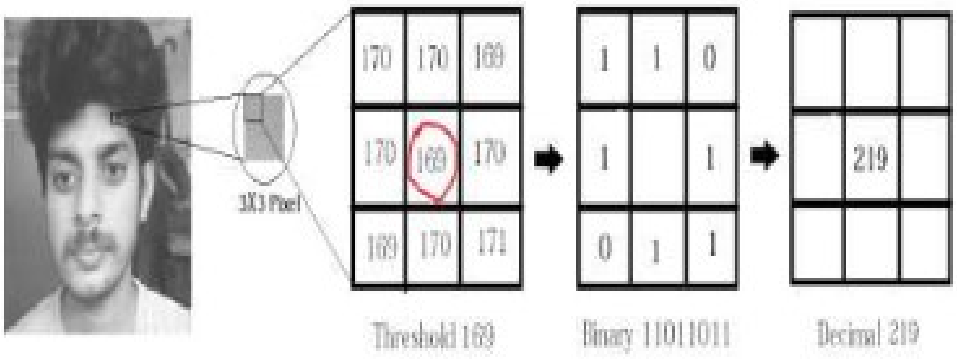
2.Training the Algorithm:

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. We need 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.

3.Applying the LBP operation:

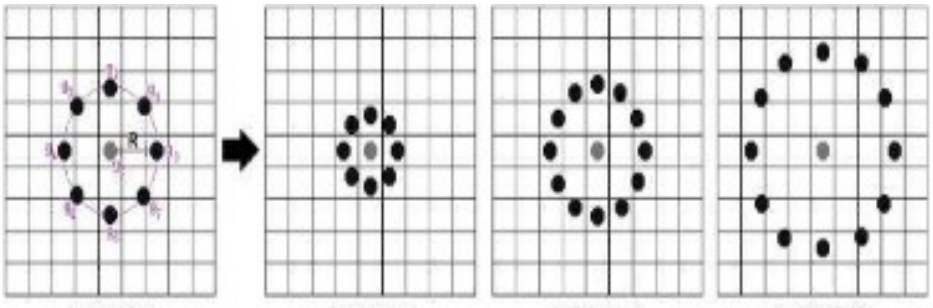
The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

The image below shows this procedure:



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 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 concatenate the binary values (e.g.
  clockwise direction), but the final result will be the same.
- Then,we convert this binary values 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.



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 GridX and GridY parameters to divide the image into multiple grids,as caan be seen in the following image.

Based on the image below, 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 occurences of each pixel intensity.
- Then,we need to concatenate each histogran to created a new and bigger histogram.Supposing we have 8X8 grids,we will have 8x8x256=16,384 characteristics of the image original image.

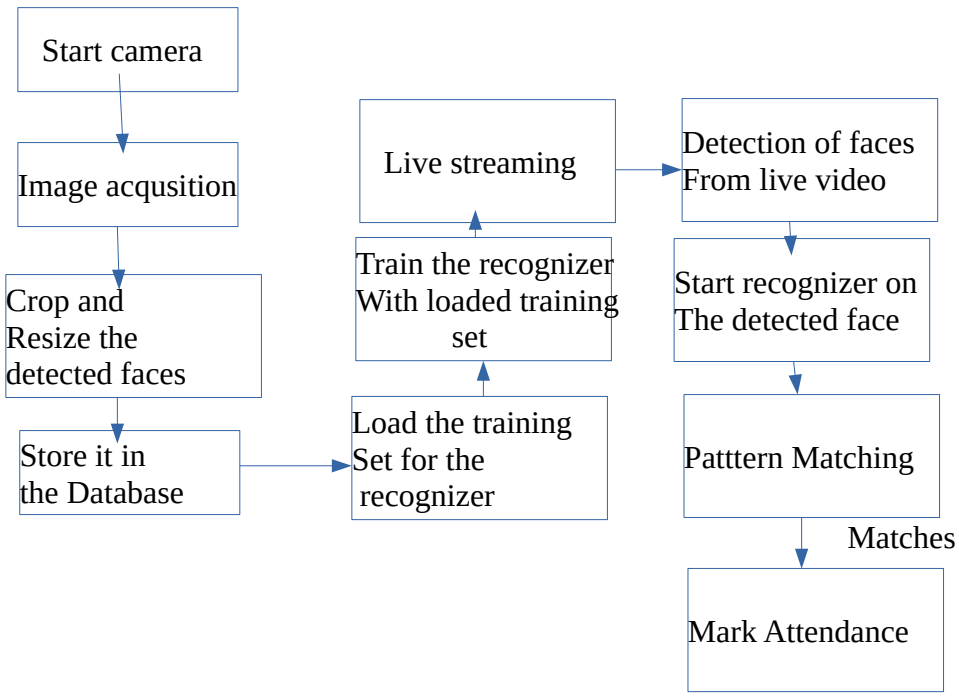
5.Performing the face recognition: In this step,the algorithm is already trained.Each histogram created is used to rpresent each image from the training dataset.So, given an input image,we perform the steps again for this new image and created a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return theimage with the closest histogrm
- We can use various approaches to compare the two histograms (calculate the distance between two histograms),for example Euclidean distance,chi-square,absolute value,etc.In this example,we can use **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 with the closest histogram.The algorithm should also return the calculated distance,which can be used as a 'confidence' measurement.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.

System Architecture



Typically this process can be divided into four stages,

1. Dataset Creation

Images of students are captured using a web cam. Multiple images of single student will be acquired with varied gestures and angles. These images undergo pre-processing. The images are cropped to obtain the Region of Interest (ROI) which will be further used in recognition process. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to gray scale images. And then these images will be saved as the names of respective student in a folder.

2. Face Detection

Face detection here is performed using Haar-Cascade Classifier with OpenCV. Haar Cascade algorithm needs to be trained to detect human faces before it can be used for facedetection. This is called feature extraction. The haar cascade training data used is an xmlfile- haarcascade_frontalface_default.

3. Face Recognition

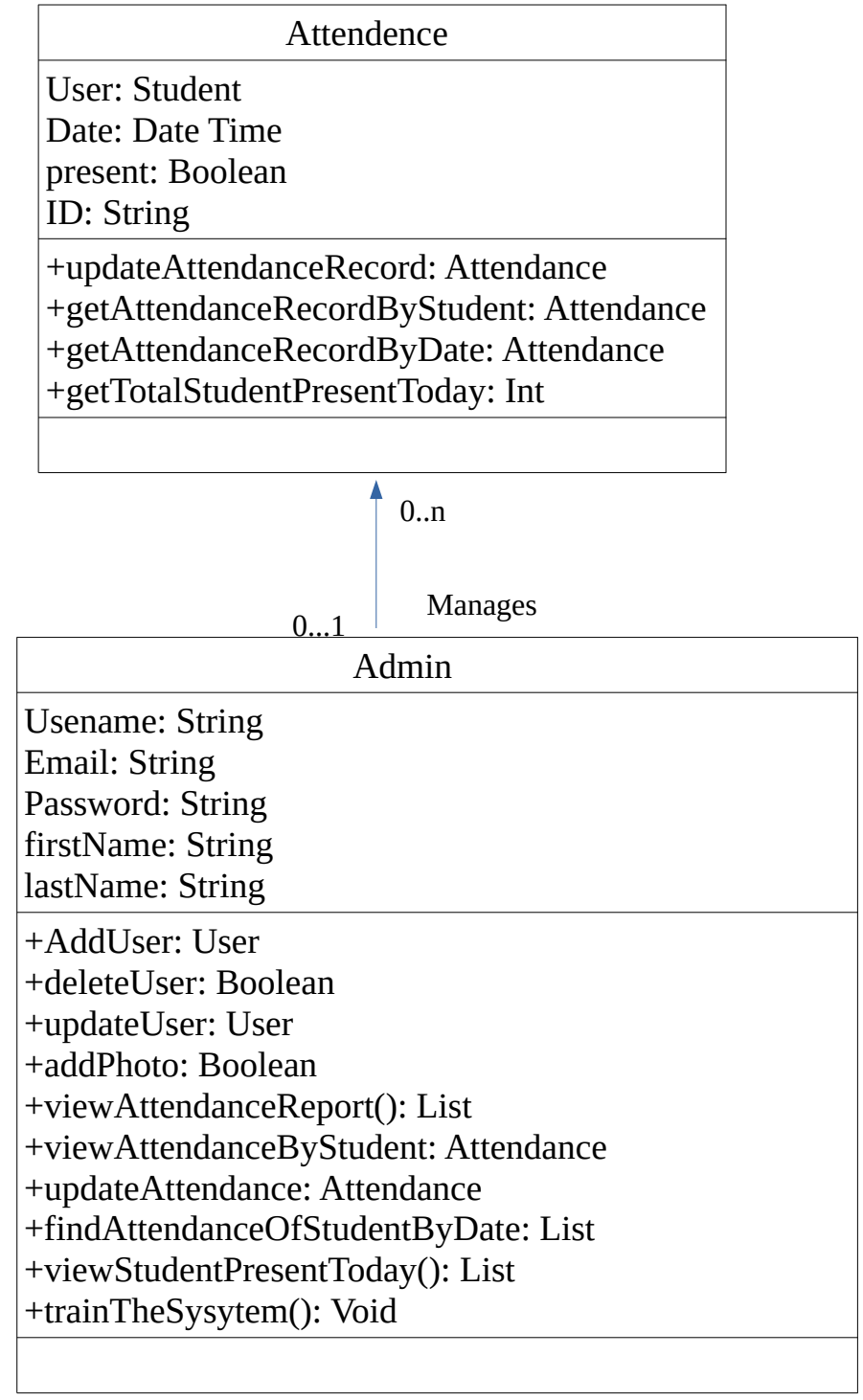
Face recognition process can be divided into three steps-prepare training data, train face recognizer, prediction. Here training data will be the images present in the dataset. They will be assigned with a integer label of the student it belongs to. These images are then used for face recognition. Face recognizer used in this system is Local Binary Pattern Histogram. Initially, the list of local binary patterns (LBP) of Histogram. Initially, the list of local binary patterns (LBP) of entire face is obtained.

4. Attendance Updation:

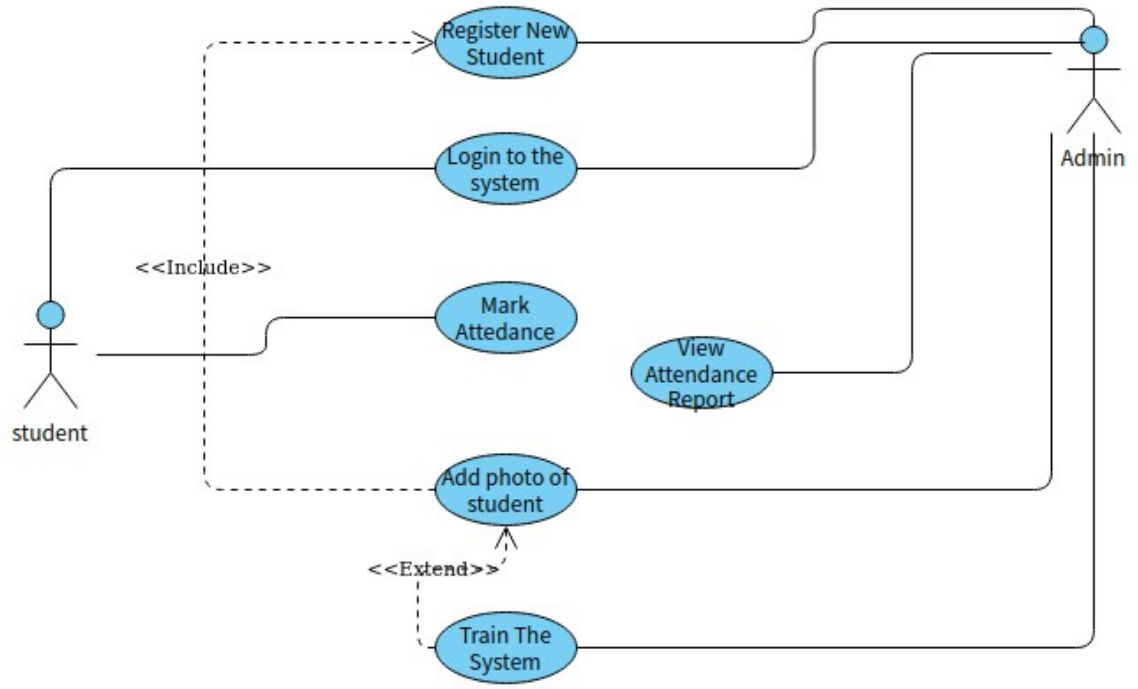
After face recognition process, the recognized faces will be marked as present in the excel sheet and the rest will be marked as absent and the list of absentees will be mailed to the respective faculties. Faculties will be updated with monthly attendance sheet at the end of every month.

System Design

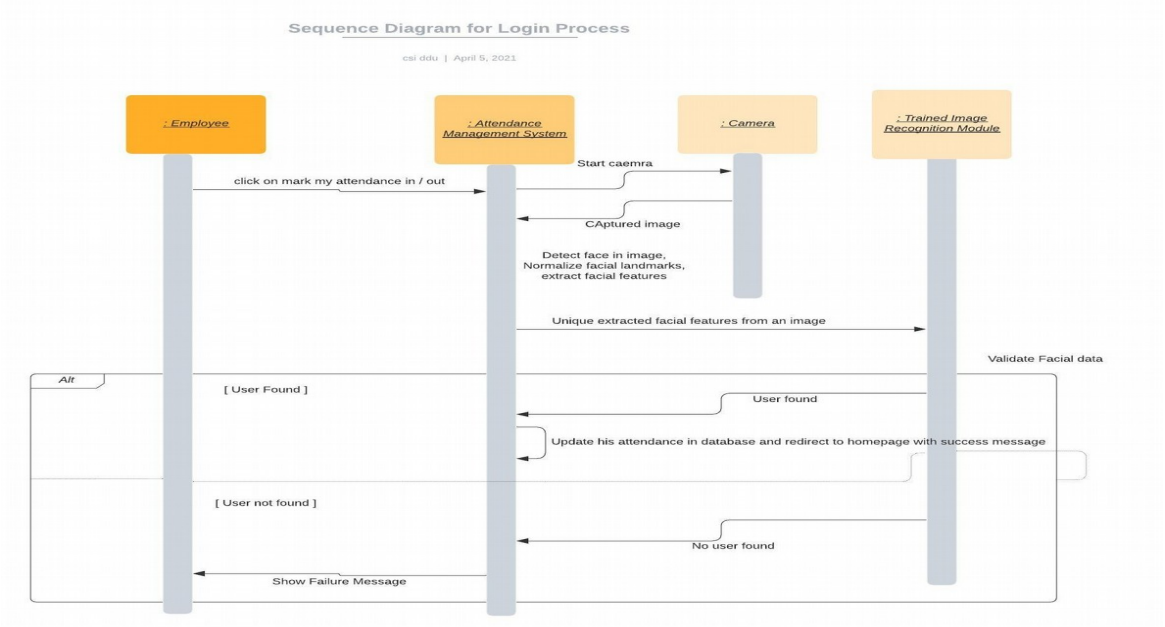
Class Diagram:



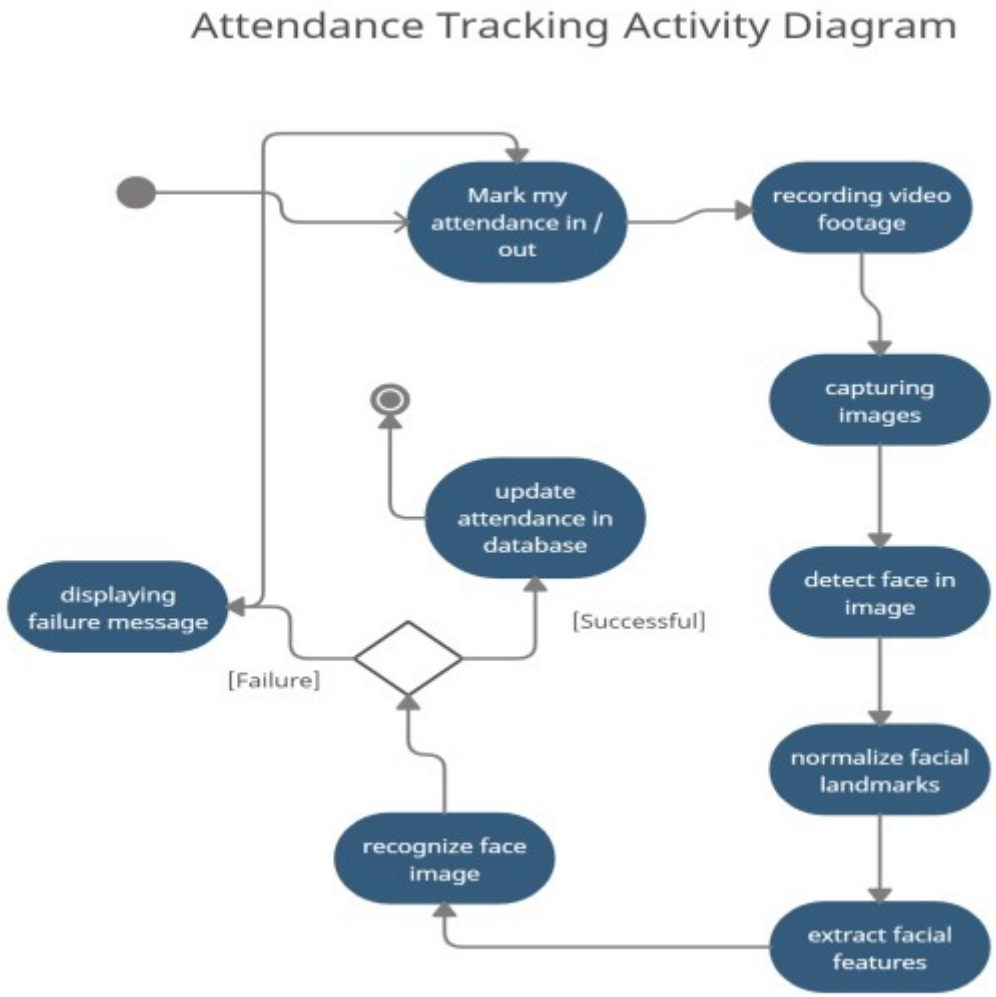
Usecase Diagram:



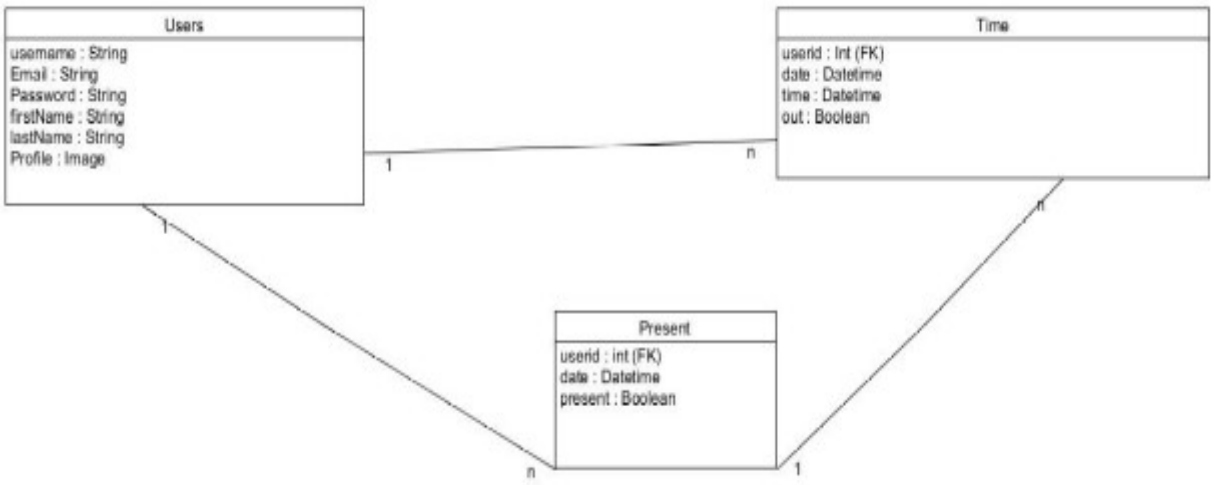
Sequence Diagram:



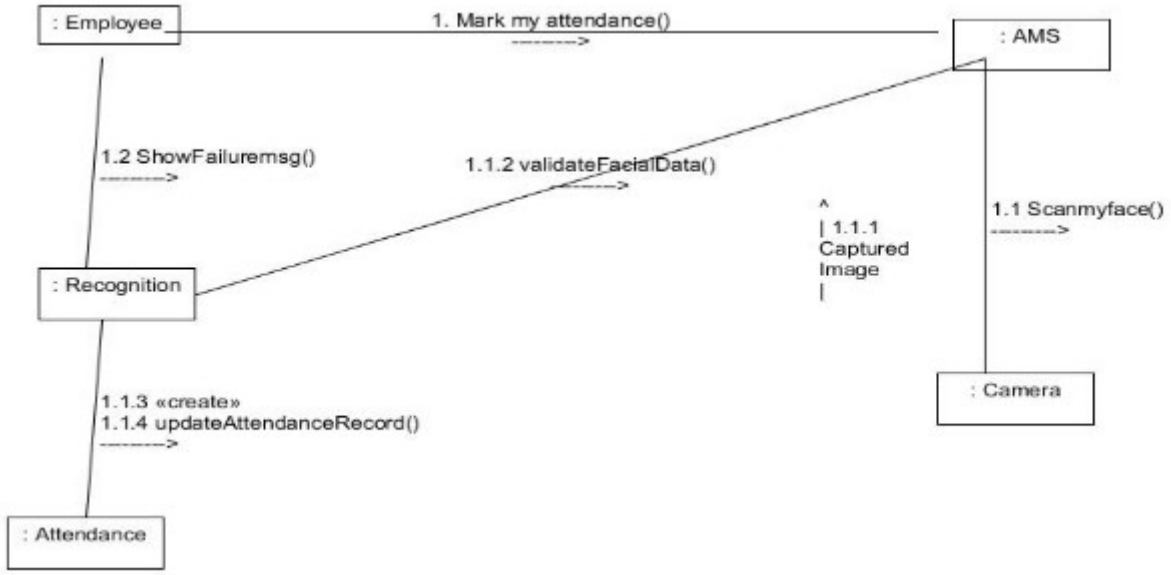
Activity Diagram:



ER Diagram:



Collaboration Diagram:



The features of the system are mainly divided into 3 modules.

Registration and Login Module:

This module mainly deals with the functionalities related to the registration of any new student to the organization, Log into the system and managing student's profile details. Using features provided by this module admin can register new student to the system and admin / student both can log into the system using their credentials.

Manage Attendance Details:

This module mainly deals with the features related to the student's attendance. Using this student can mark their presence, time-in and time-out in the system. Admin can see the availability report of each student =, student can see his/her attendance report along with some possible filters such as filter by student and filter by date.

Manage Employee Details:

This module mainly deals with the features related to the student's profile. Using this admin can add a photo of the newly registered student during registration. Admin can also command the system explicitly to train the model and system will make necessary calculation and will generate some data which will be used internally to identify each student uniquely.

Function prototypes which implement major functionality

- List<Attendance> viewMyAttendanceReport(int stdId);
- Int totalStudentRegistered();
- List<Attendance> getAttendanceRecordByStudent(int stdid);
- Boolean updateAttendanceRecord(int stdid, Attendance update);
- Boolean registerStudnet(Student new_student);
- Boolean addPhoto(int empId, string photo);

Requirement Specification

We have 2 types of users of the system.
1. Student/Employee
2. Admin

Following functionalities can be performed by the admin:

-Login
-Register new student to the system
-Add student photos to the training data set
-Train the model
-View attendance report of all students. Attendance can be filtered by id or name

Following functionalities can be performed by the student:

-Mark his/her time-in and time-out by scanning their face
-View attendance report of self

FUNCTIONAL REQUIREMENTS

-Manage Registration and Login

-Register new student

Description: Admin can register new
Input:Admin Details
Output: success message displaying the user has been created.

-Log-In to the system

Input: User credentials
Output: If the credentials are correct, user will be redirected to the dashboard of the system
Exception Flow: If the entered credentials are incorrect then user will be redirected to the login page again displaying an error message.

-Manage Attendance Details

-Mark your attendance

Input: User will scan his/her face using the external web camera.
Output:system will identify the user uniquely and will mark his/her attendance to the database.The same success message will be transmitted to the user.

-View my attendance report

Description: Student may often need to see his / her attendance record throughout the month or year. Using this feature one can see his / her attendance record till the date.
Input: User selection
Output: Statistical analytics of the particular employee who is currently logged into the system will be displayed.

-View students's attendance report

Description: This feature is for admin. Admin can monitor the availability of each studenr till the date. i.e., how many students are present today out of total students etc. can be monitored.
Input: user selection
Output: Attendance record of each student including how many students are present today out of total along with the availability graph.

- Manage Student Details

-Add photo of the student

Description: Admin only can access this feature. Admin can add a photo of a student during the registration process.

Input: Username of the student

Output: Success message record has been added.

Process: System will process an image and will generate necessary system data to identify student uniquely.

-Train the system

Input: user selection

Output: system will process all the available records of the students and will generate necessary system data to identify each student uniquely.

Functional Requirements:

Functional requirement refers to the functionalities that are applicable to a system. The functional requirements of smart attendance system are stated below. The system must be able to:

1. Registration
2. Login / Logout
3. Manage User Profile
4. Update user profile
5. View My Attendance
6. View Attendance by Date
7. View Attendance by Employee
8. Manage Attendance
9. Mark my attendance In
10. Mark my attendance Out
11. Add photos
12. Add new employee
13. Train the system
14. View Attendance record by date
15. View no. of employee present today
16. View Total number of employees

NON FUNCTIONAL REQUIREMENTS:

A non-functional requirement is a system must behave or how is the system's behavior. This also specifies how the system's quality characteristics or quality attributes In order to put Department of CSE, Face Recognition Attendance Management System this constraint upon the specific system behavior, the qualities goals of the designed system
should go in these Execution qualities:

Functionality
Security
Usability
Effectiveness & Efficiency Evolution qualities:
Availability
Reliability
Manageability

Non-functional requirements are qualities or traits of the framework that can pass judgment on its activity. The following point explains them:

Accuracy and Precision: the framework should perform its process in accuracy and Precision in order to avoid problems.

Security: For saving the student's privacy, the framework should be secure as data privacyplays important role in software development.

Modifiability: the system should be easy to modify, in case any attendance or any record contains wrong or incorrect data can be easily corrected.

Usability: the framework should be easy to deal with and simple to understand.

Speed and Responsiveness: the execution of operations must be fast

Hardware Configuration:

Ram	8 GB
Hard Disk	10GB
Processor	1.0 GHz

Software Requirements:

- **Programming Languages :** Python
- **Technology :** Machine Learning
- **Built-in Libraries :** Open CV
  My Sql Connector
  Numpy
  Datetime
  OS
  Pillow
- **GUI :** Tkinter
- **Operating System :** Windows
- **Database :** MySql

Tools Used

Anaconda:

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager.It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. Before version 20.3, when pip installed a package, it automatically installed any dependent Python packages without checking if these conflict with previously installed packages. It would install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Tensor Flow, could find that it stopped working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by TensorFlow. In some cases, the package would appear to work but produce different results in detail. While pip has since implemented consistent dependency resolution, this difference accounts for a historical differentiation of the conda package manager.

In contrast, conda analyses the current environment including everything currently installed, and, together with any version limitations specified (e.g. the user may wish to have TensorFlow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done. Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries for Windows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [16] PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7.

Coding and Implementation

Facerecognition.py

```
        def face_recog(self):
                def draw_boundray(img,classifier,scaleFactor,minNeighbors,color,text,clf):
                        gray_image=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
                        features=classifier.detectMultiScale(gray_image,scaleFactor,minNeighbors)

                        coord=[]
                        for (x,y,w,h) in features:
                                cv2.rectangle(img,(x,y),(x+w,y+h),(0,255,0),3)
                                id,predict=clf.predict(gray_image[y:y+h,x:x+w])
                                confidence=int((100*(1-predict/300)))

                                conn=mysql.connector.connect(host="localhost",user="root",passwor
d="tharuni123",database="face_recognizer",port="8086",auth_plugin='mysql_native_password')
                                my_cursor=conn.cursor()

                                my_cursor.execute("select Student_id from student where
Student_No="+str(id))
                                id1=my_cursor.fetchone()
                                if id1:
                                        id1="+".join(id1)

                                my_cursor.execute("select Student_Name from student where
Student_No="+str(id))
                                n=my_cursor.fetchone()
                                 if n:
                                         n="+".join(n)

                                my_cursor.execute("select AMS_Roll_No from student where
Student_No="+str(id))
                                 r=my_cursor.fetchone()
                                if r:
                                        r="+".join(r)

                                my_cursor.execute("select Branch from student where
Student_No="+str(id))
                                 br=my_cursor.fetchone()
                                if br:
                                        br="+".join(br)

                                if confidence>77:
                                        cv2.putText(img,f"IDNo:{id1}",(x,y-
100),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,255,0),2)
```

```
                                        cv2.putText(img,f"Name:{n}",(x,y-
75),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,255,0),2)
                                        cv2.putText(img,f"AMS_roll_no:{r}",(x,y-
40),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,255,0),2)
                                        cv2.putText(img,f"branch:{br}",(x,y-
10),cv2.FONT_HERSHEY_COMPLEX,0.8,(0,255,0),2)
                                        self.mark_attendance(id1,n,r,br)
                                else:
                                        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3)
                        cv2.putText(img,"Unknown Face",(x,y-5),cv2.FONT_HERSHEY_COMPLEX,0.8,
(255,0,0),2)

                    coord=[x,y,w,y]

                return coord
            def recognize(img,clf,faceCascade):
                coord=draw_boundray(img,faceCascade,1.1,10,(255,25,255),"Face",clf)
                return img

                 faceCascade=cv2.CascadeClassifier(cv2.data.haarcascades+"haarcascade_frontalface_defa
ult.xml")
                clf=cv2.face.LBPHFaceRecognizer_create()
                clf.read("classifier.xml")

                 video_cap=cv2.VideoCapture(0,cv2.CAP_DSHOW)

            while True:
                ret,img=video_cap.read()
                img=recognize(img,clf,faceCascade)
                cv2.imshow("Welcome To Face Recognition",img)

                if cv2.waitKey(1)==13:
                    break
            video_cap.release()
            cv2.destroyAllWindows()
```

The above code is for recognizing face and marking attendance of a student.

Testing

System testing is the stage of implementation, which aimed at ensuring that system works accurately and efficiently before the live operation commence. Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an error. A successful test is one that answers a yet undiscovered error.

Testing is vital to the success of the system. System testing makes a logical assumption that if all parts of the system are correct, the goal will be successfully achieved. The candidate system is subject to variety of tests-on-line response, Volume Street, recovery and security and usability test. A series of tests are performed before the system is ready for the user acceptance testing. Any engineered product can be tested in one of the following ways. Knowing the specified function that a product has been designed to from, test can be conducted to demonstrate each function is fully operational. Knowing the internal working of a product, tests can be conducted to ensure that "al gears mesh", that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised.a

UNIT TESTING:

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields.

INTEGRATION TESTING:

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance. There are two types of integration testing. They are: Top-down integration testing,Bottom-up integration testing.

WHITE BOX TESTING:

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

BLACK BOX TESTING:

Black box testing is done to find incorrect or missing function Interface error Errors in external database access Performance errors Initialization and termination errors In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behaviour of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational.

VALIDATION TESTING:

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

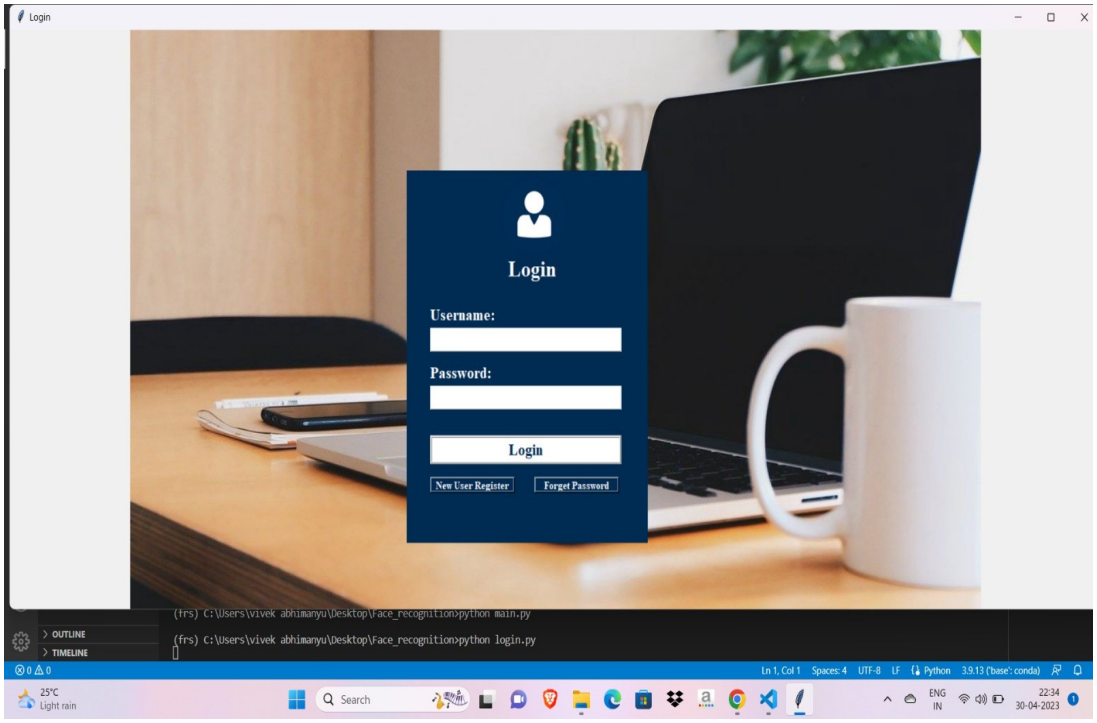
USER ACCEPTANCE TESTING:

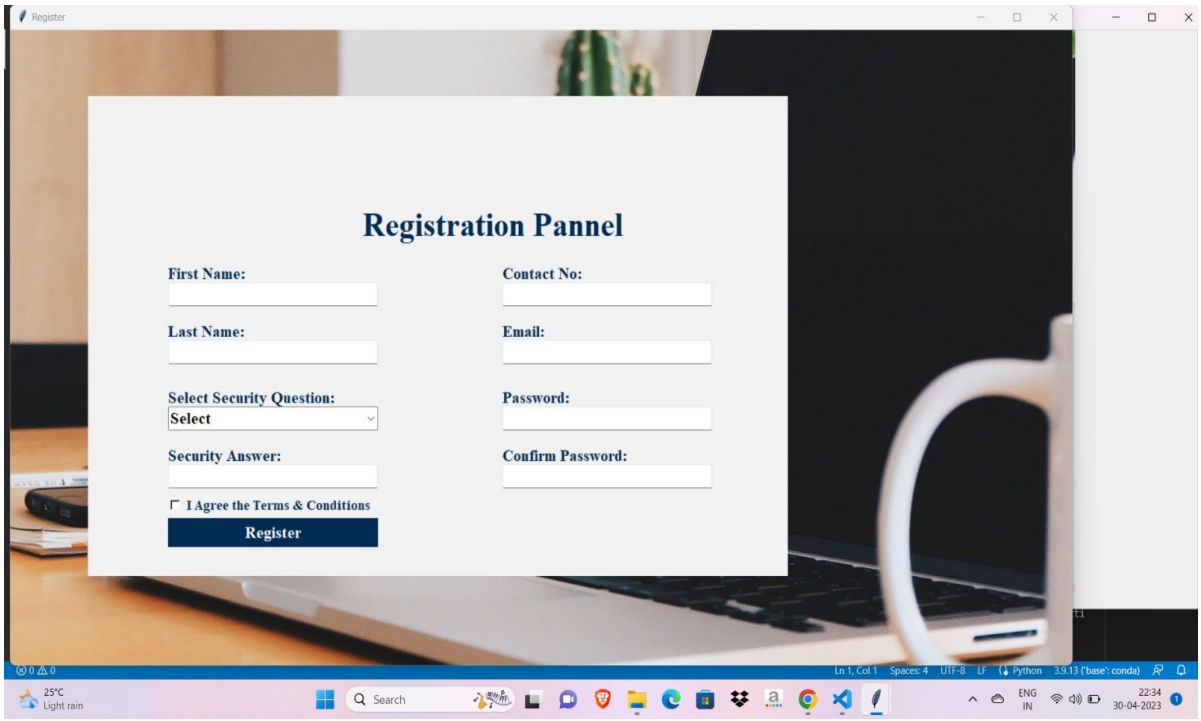
User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

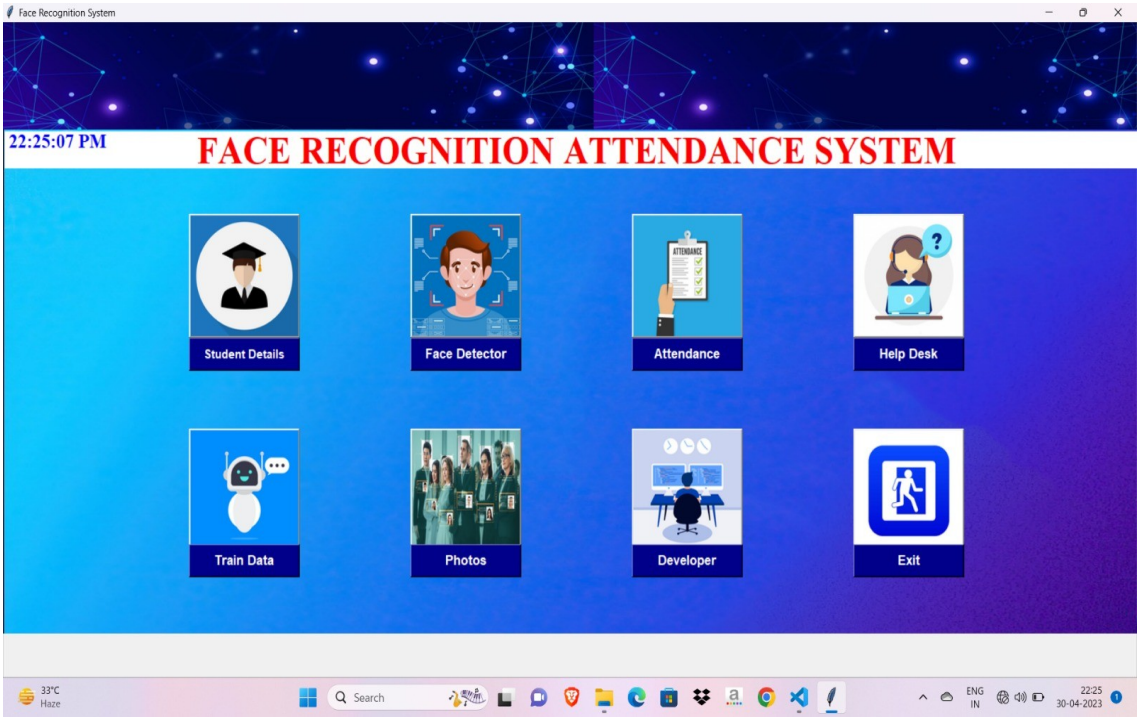
OUTPUT TESTING:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

Simulation Results




Register
Registration Pannel
First Name:
Contact No:
Last Name:
Email:
Select Security Question:
Select
Password:
Security Answer:
Confirm Password:
I Agree the Terms & Conditions
Register


Face Recognition System
22:25:07 PM
FACE RECOGNITION ATTENDANCE SYSTEM
Student Details
Face Detector
Attendance
Help Desk
Train Data
Photos
Developer
Exit

Conclusion

The Face Recognition Attendance Management System is simple, accurate and works efficiently. This system works automatically once the registration of individual student is created by the administration. There is a need to utilize few algorithms that can perceive the appearances in order to improve the system performance and recognition accuracy.This system aims to build an effective class attendance system using face recognition techniques. The proposed system will be able to mark the attendance via face Id. It will detect faces via webcam and then recognize the faces. After recognition, it will mark the attendance of the recognized student and update the attendance record.

Future Scope

- A feature which can give intruder alert can be included in the system. Furthermore, the images of unknown people can be saved in an efficient manner and displayed in the system for better security.
- The number of training images can be reduced so that less storage is required. This can be done by removing duplicate images of the same person, or images with similar embeddings.
- The training time can be reduced by retraining the classifier only for the newly added images.
- A feature can be added where an employee is automatically sent a warning if his attendance or working hours are below the threshold.
- Wrongly classified images can be added to the training dataset with the correct label so as to increase the accuracy of the recognition model.

References

1. Francis Galton, “Personal identification and description,” In Nature, pp. 173-177, June 21,1888.
2. W. Zaho, “Robust image based 3D face recognition,” Ph.D. Thesis, Maryland University,1999.
3. R. Chellappa, C.L. Wilson and C. Sirohey, ”Humain and machine recognition of faces: A survey,” Proc. IEEE, vol. 83, no. 5, pp. 705- 740, may 1995.
4. T. Fromherz, P. Stucki, M. Bichsel, “A survey of face recognition,” MML Technical Report, No 97.01, Dept. of Computer Science, University of Zurich, Zurich, 1997.
5. T. Riklin-Raviv and A. Shashua, “The Quotient image: Class based recognition and synthesis under varying illumination conditions,” In CVPR, P. II: pp. 566-571,1999.
6. G.j. Edwards, T.f. Cootes and C.J. Taylor, “Face recognition using active appearance models,” In ECCV, 1998.
7. T. Sim, R. Sukthankar, M. Mullin and S. Baluja, “Memory-based face recognition for vistor identification,” In AFGR, 2000.
8. T. Sim and T. Kanade, “Combing models and exemplars for face recognition: An illuminating example,” In Proceeding Of Workshop on Models Versus Exemplars in Computer Vision, CUPR 2001.
9. L. Sirovitch and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” Journal of the Optical Society of America A, vol. 2, pp. 519–524, 1987.
10. M. Turk and A. Pentland “Face recognition using eigenfaces,” In Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591, 1991.
11. P. Belhumeur, P. Hespanha, and D. Kriegman, “Eigenfaces vs fisherfaces: Recognition using class specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

**********THANK YOU**********