Project Report on

Smart Attendance Management System

Submitted in the subject
of
Project Based Learning V
in
ELECTRONICS & TELECOMMUNICATION ENGINEERING
by

Yashi Chaturvedi	05
Amal Mehta	38
Bhavesh Purohit	60

Under the Guidance of
Ms. Sukruti Kaulgud
Assistant Professor

Thakur College of Engineering and Technology
A-Block, Thakur Educational Campus, Shyamnarayan Thakur Marg, Thakur Village,
Kandivali(E). Mumbai - 400101.

Year 2020-2021

Abstract

In the present academic system, regular class attendance of students' plays a significant role in performance assessment and quality monitoring. Maintaining an attendance register daily is a difficult and time consuming task. Providing an accurate attendance marking system in real-time is challenging. It is tough to mark the attendance of a student in the large classroom when there are many students attending the class. The conventional methods practised in most of the institutions are by calling names or signing on papers, which is highly time-consuming and insecure.

There are many automated methods for the same available like Biometric, RFID, eye detection, voice recognition, and many more. This paper provides an efficient and smart method for marking attendance. As it is known that primary identification for any human is its face, face recognition provides an accurate system which overcomes the ambiguities like fake attendance, high cost, and time consumption. This system uses a face recognizer library for facial recognition and storing attendance.

The technology working behind will be the face recognition system. The human face is one of the natural traits that can uniquely identify an individual. Therefore, it is used to trace identity as the possibilities for a face to deviate or being duplicated is low. In this project, face databases will be created to pump data into the recognizer algorithm. Then, during the attendance taking session, faces will be compared against the database to seek for identity. When an individual is identified, its attendance will be taken down automatically saving necessary information into a database.

CONTENTS

LIST OF FIGURES/TABLES

Chapter 1
Overview

1.1 Importance of the Project

Many attendance management systems that exist nowadays lack of the efficiency and information sharing. Therefore, in this project, those limitations will be overcome and also further improved and are as follows :

- Students will be more punctual in attending classes. This is due to the attendance of a student can only be taken personally where any absentees will be noticed by the system. This can not only train the student to be punctual as well as avoid any immoral ethics such as signing the attendance for their friends.
- The institution can save a lot of resources as enforcement is now done by means of technology rather than human supervision which will waste a lot of human resources for an insignificant process.
- The application can operate on any device at any location as long as there is Wi-Fi coverage or Ethernet connection which makes the attendance system to be portable to be placed at any intended location. For example, the device can be placed at the entrance of the classroom to take the attendance.
- It saves a lot of cost in the sense that it has eliminated the paperwork completely.
- The system is also time effective because all calculations are all automated.In short, the project is developed to solve the existing issues in the old attendance system.

1.2 Social Problem Identification

According to the previous attendance management system , the **accuracy of the data** collected is the biggest issue. This is because the attendance might not be recorded personally by the original person, in another word, the attendance of a particular person can be taken by a third party without the realization of the institution which violates the accuracy of the data. For example, student A is lazy to attend a particular class, so student B helped him/her to sign for the attendance which in fact student A didn't attend the class, but the system overlooked this matter due to no enforcement practice. Supposing the institution establishes enforcement, it might need to waste a lot of human resources and time which in turn will not be practical at all. Thus, all the recorded attendance in the previous system is not reliable for analysis usage.

The second problem of the previous system is where it is too **time consuming.** Assuming the time taken for a student to sign his/her attendance on a 3-4 paged name list is approximately 1 minute. In 1 hour, only approximately 60 students can sign their attendance which is obviously inefficient and time consuming.

The third issue is with the **accessibility of those information by the legitimate concerned party**. For example, most of the parents are very concerned to track their child's actual whereabouts to ensure their kid really attends the classes in college school. However in the previous system, there are no ways for the parents to access such information. Therefore, evolution needs to be done to the previous system to improve efficiency, data accuracy and provide accessibility to the information for those legitimate parties.

1.3 Technical/Feasible Solution

The technical solutions are:

- To develop a portable Smart Attendance System which is handy and self-powered.
- To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.
- Have enough memory space to store the database.
- Able to recognize the face of an individual accurately based on the face database.
- Allow parents to track their child's attendance.
- Develop a database for the attendance management system.
- Provide a user-friendly interface for admins to access the attendance database and for non-admins (parents) to check their child"s attendance by mailing the attendance.
- Allow new students or staff to store their faces in the database by using a GUI.
- Able to show an indication to the user whether the face- recognition process is successful or not.

1.4 Scope of the Project

The main intention of this project is to solve the issues encountered in the old attendance system while reproducing a brand new innovative smart system that can provide convenience to the institution. In this project, an application will be developed which is capable of recognising the identity of each individual and eventually record down the data into a database system.

- The targeted groups of the attendance monitoring system are the students and staff of an educational institution.
- The database of the attendance management system can hold up to 2000 individual's information.
- The facial recognition process can only be done for 1 person at a time.
- The project has to work under a Wi-Fi coverage area or under Ethernet connection, as the system needs to update the database of the attendance system constantly.
- The device on which the application is running is powered up by a power bank to improve the portability of the application.
- The higher the megapixel count on a camera sensor, the greater amount of detail that can be captured in a picture. The recognition will be quicker and more accurate.

Chapter 2
Proposed Work

2.1 Block Diagram and Flow Chart

A database created with the student's personal data along with their face images. Figure 1, shows the block diagram of the system. A camera is used to capture the images of the faces or to capture the real-time video. Optical devices such as the camera or video recorder are used to accomplish this task. The students' face images to be recognized are fed to the image processing block where it performs preprocessing, face detection, and face recognition tasks. Preprocessing includes tasks such as cropping of image and enhancement procedures. These processed images are fed to the face recognition algorithm. These database images are then compared with the real-time recognized faces to identify the student.

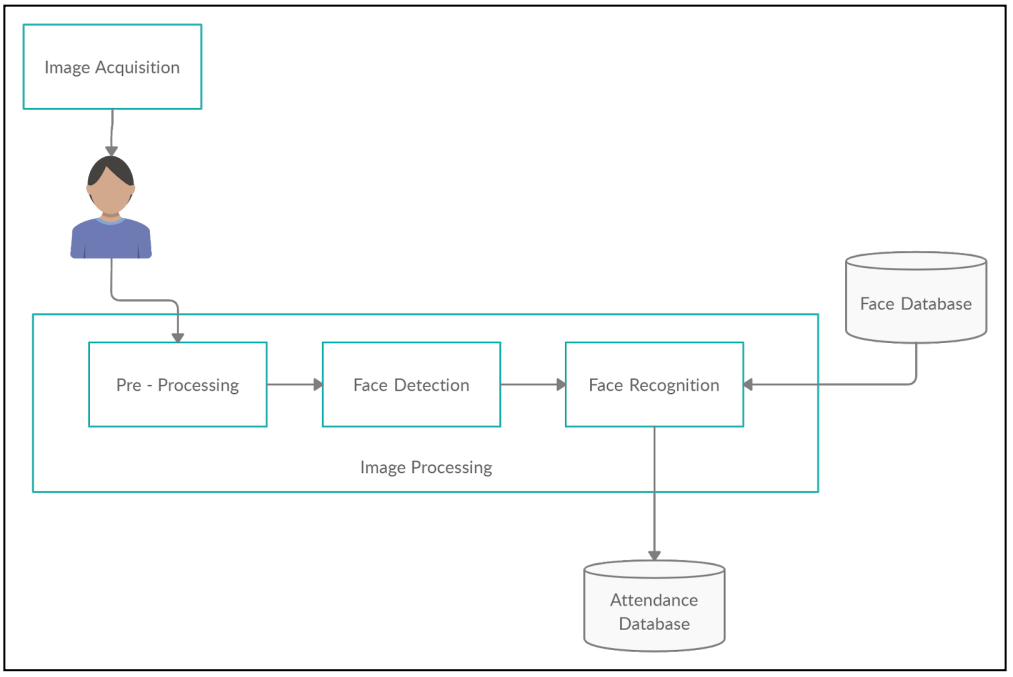

Figure 1: Block Diagram of the System

Figure 2, shows the flow chart of the proposed methodology of the project. The integrated model illustrating the basic steps for database creation of all the students enrolled in the class, then recognition process for further attendance marking, after that comparing the results obtained with the predefined database for accurate output, if any student who is present in the database but was absent in the recognition process will be considered absent in that class. For the students who are present, a file is generated in the server database with students' details along with the timestamp. This database can be accessed by all the teachers and create an excel file for the attendance sheet.

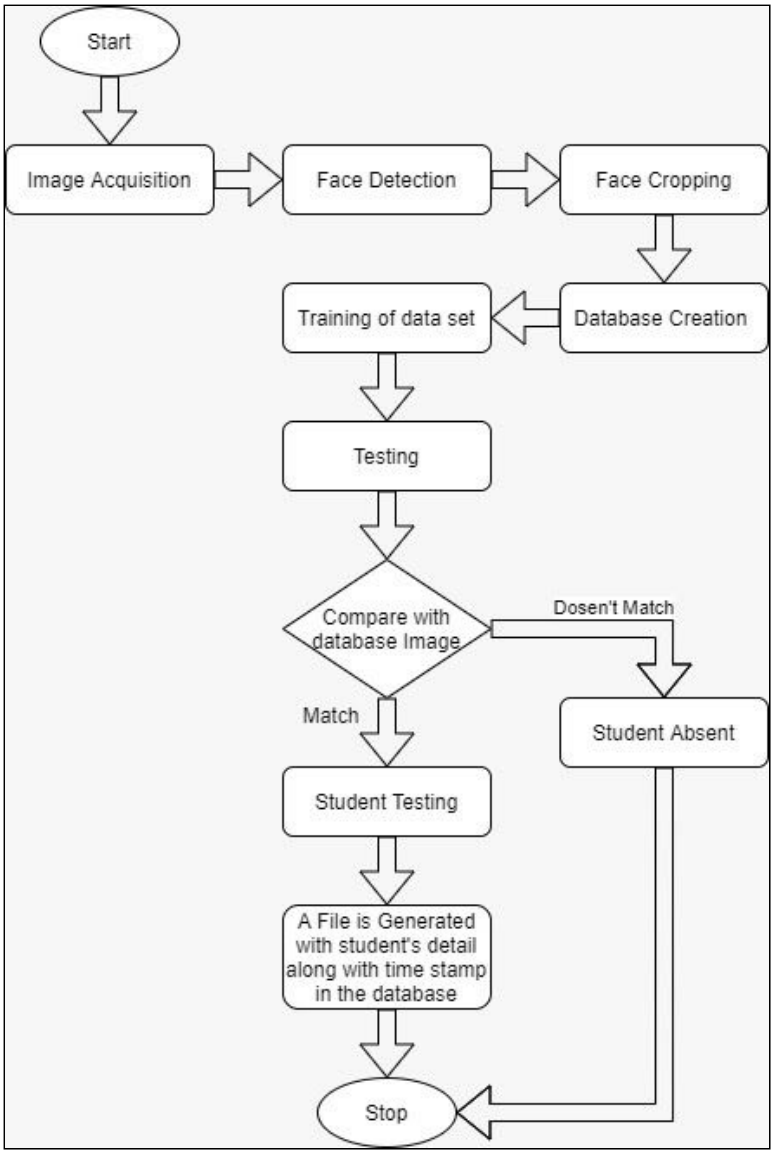

Figure 2: Flowchart of the System

2.2 Methodology

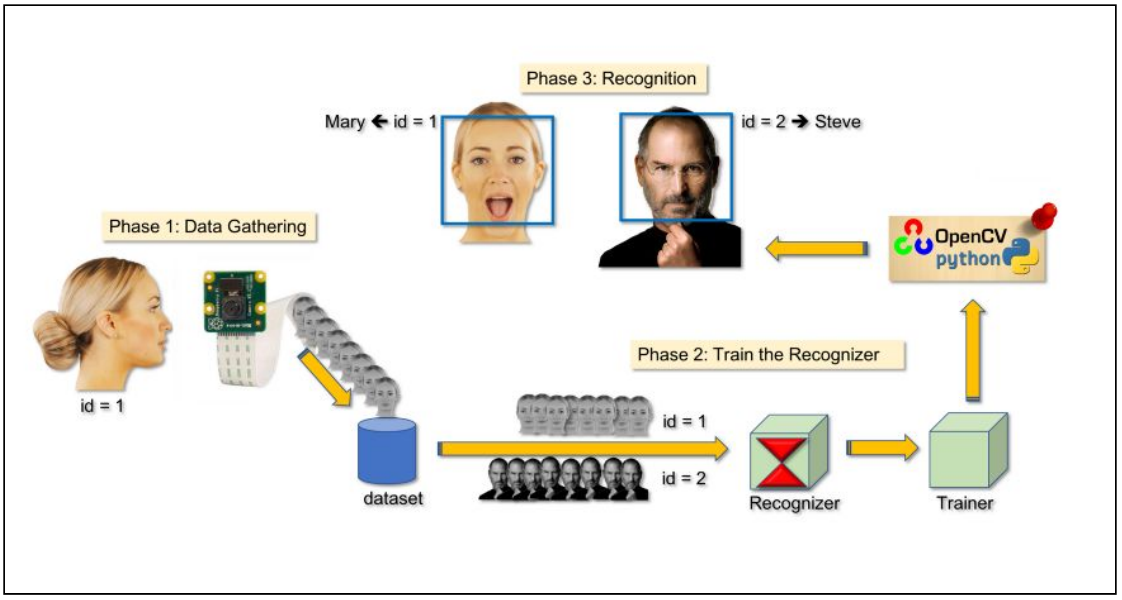

Figure 3: Data Flow Diagram

There are three main steps in this project:

- **Face Detection and Data Gathering**
- **Train the Recognizer**
- **Face Recognition**

Face Detection and Data Gathering:

The basic task on Face Recognition is 'Face Detecting'. The first step is to "capture" a face in order to recognize it, when compared with a new face captured in future. Here, we have used the 'Haar cascade classifier'. Object Detection using Haar feature-based cascade classifiers is an effective object detection method. 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. 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. OpenCV comes with a trainer as well as a detector.

Train the Recognizer:

The second task is to take all user data from our dataset and train the OpenCV Recognizer. This is done directly by a specific OpenCV function. The result is a .yml file that will be saved on a "trainer/" directory. We have used it as a recognizer, the LBPH (LOCAL BINARY PATTERNS HISTOGRAMS) Face Recognizer, included on the OpenCV package.

The function "getImagesAndLabels (path)", will take all photos in the directory: "dataset/", returning 2 arrays: "Ids" and "faces". With those arrays as input, we train our recognizer. As a result, a file named "trainer.yml" is saved in the trainer directory that was previously created by us.

Face Recognition:

The last task of our project is to capture a fresh face on our camera and if this person had his face captured and trained before, our recognizer will make a "prediction" returning its id and an index, showing how confident the recognizer is with this match. We are including here a new array, so we will display "names", instead of numbered ids. So, for example: Mary will the user with id = 1; Steve: id=2, etc.

Next step is to detect a face, same as the Haar Cascade classifier. Once the face is detected we call recognizer.predict(). The recognizer.predict (), takes as a parameter a captured portion of the face to be analyzed and returns its probable owner, indicating its id and how much confidence the recognizer is in relation with this match.

The last step is, if the recognizer could predict a face, a text over the image with the probable id and how much is the "probability" in % that the match is correct ("probability" = 100 — confidence index). If not, an "unknown" label is put on the face.

2.3 Learning issues related to it and Development Experience

We have learned Python in our 4th semester under the subject Professional Skills. We have also learned Python from a course in Coursera by Rice University. In that we learnt the basics of python which includes scripting, data types, data structures, syntax and much more. For Face Recognition in Python, we did an online course and guided project under Coursera Project Network.

By developing this project we learnt different types of libraries like numpy, open cv, os(operating system) etc. we also came to know how to convert image into grayscale then how to train the gray scaled image by using a trainer function and at last, how to make the trainer recognise the face of a person.We developed this project using open cv and python. We have used different types of libraries (like numpy, os, open cv etc) to develop this project.

While doing the project we faced the following problems:

1. Downloading and importing the libraries.
2. Many Syntax errors, one of which the camera wasn't capturing the images.
3. Creation of the database and connecting the code with it.

Chapter 3
Analysis

3.1 User Lens

Crime Investigation Department: Face recognition is currently being used to instantly identify when known shoplifters, organized retail criminals or people with a history of fraud enter retail establishments. Photographs of individuals can be matched against large databases of criminals so that loss prevention and retail security professionals can be instantly notified when a shopper enters a store that prevents a threat.

Cell Phone Companies: A variety of phones including the latest iPhone are now using face recognition to unlock phones. This technology is a powerful way to protect personal data and ensure that, if a phone is stolen, sensitive data remains inaccessible by the perpetrator.

Smarter Advertising: Face recognition has the ability to make advertising more targeted by making educated guesses at people's age and gender. Companies like Tesco are already planning on installing screens at gas stations with face recognition built in. It's only a matter of time before face-recognition becomes an omni-present advertising technology.

Aid Forensic Investigations: Facial recognition can aid forensic investigations by automatically recognizing individuals in security footage or other videos. Face recognition software can also be used to identify dead or unconscious individuals at crime scenes.

Casinos: Face recognition can help casinos recognize the moment that a cheater or advantage gambler enters a casino. In addition, face recognition can recognize members of voluntary exclusion lists, who can cost casinos hefty fines if they're caught gambling.

Transactions: In China, there is a financial services company called Ant Financial that enables customers to pay for meals by scanning their faces. Customers place orders through a digital menu, and then use face scan as a payment option. After providing their telephone number they can then purchase their meal.

ATMs: It seems likely that face scans will eventually replace ATM cards completely since face recognition is such a powerful identity authentication tool. But in the meantime, face recognition can be used to make sure that individuals using ATMs cards are who they say they are. Face recognition is currently being used at ATMs in Macau to protect peoples' identities.

Airports: Airlines have already started using face recognition to help people check bags, check into flights and board planes faster. It seems like we are quickly moving toward a future in which air travel is not only safer than ever before, but also more convenient than any period in history.

3.2 Application of the Project

Face Identification: Face recognition systems identify people by their face images. Face recognition systems establish the presence of an authorized person rather than just checking whether a valid identification (ID) or key is being used or whether the user knows the secret personal identification numbers (Pins) or passwords.

Access Control: In many of the access control applications, such as office access or computer logon, the size of the group of people that need to be recognized is relatively small. The face pictures are also caught under natural conditions, such as frontal faces and indoor illumination. The face recognition system of this application can achieve high accuracy without much cooperation from the user.

Security: Today more than ever, security is a primary concern at airports and for airline staff offices and passengers. Airport protection systems that use face recognition technology have been implemented at many airports around the world.

Image Database Investigations: Searching image databases of licensed drivers, benefit recipients, missing children, immigrants and police bookings.

General Identity Verification: Electoral registration, banking, electronic commerce, identifying new-borns, national IDs, passports, employee IDs.

Surveillance: Like security applications in public places, surveillance by face recognition systems has a low user satisfaction level, if not lower. Free lighting conditions, face orientations and other divisors all make the deployment of face recognition systems for large scale surveillance a challenging task.

3.3 Future Lens

The further advancements in the project can be made by working on the following points:

- Creating a much **larger database** to store the credentials of the employees in a particular organization.
- Adding **login permissions** for the staff, students or any other users.
- For students, sending an **email notification** to the parents for the students who are absent.
- Check whether the person is wearing **Face Mask** or not.
- **GSM module:** Suppose if a culprit is detected, then the detected signal can be transmitted using the GSM module to the central control room of the police station. With the help of the ISDN number of GSM, the culprit in the survivillence area will be recognized.
- Advancing facial recognition, **from the 2D facial recognition technology to 3D facial recognition technology.**

Chapter 4
Results

4.1 Results and Discussions

The results obtained by the model are shown in the following figures:

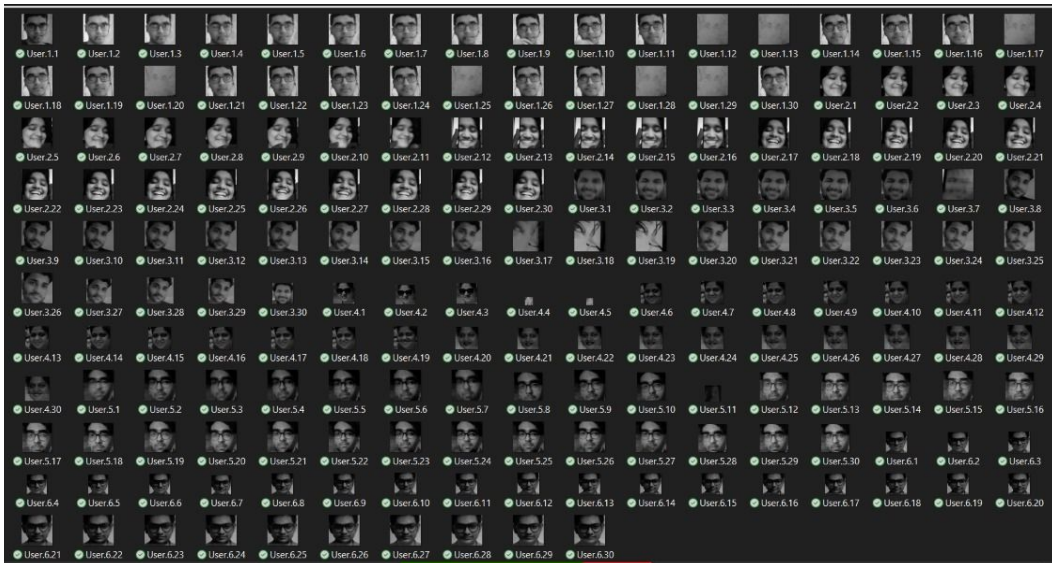

Figure 4: The Dataset Folder

Database creation of the students enrolled in the class. Webcam takes 30 images of each student and stores for further process.

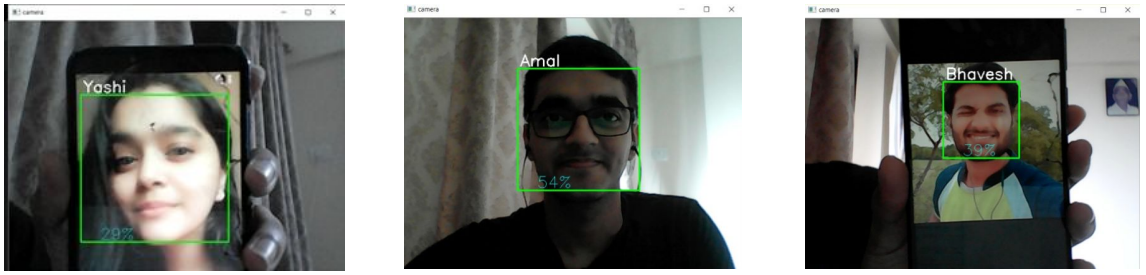

Figure 5: Face Recognition

Once the system is trained using the Dataset Folder, trainer.yml file is created. When the face recognition code is run, it takes help of the trainer.yml file. This file returns the id and the name of the person. The name along with the confidence percent is reflected on the screen if the face is detected.

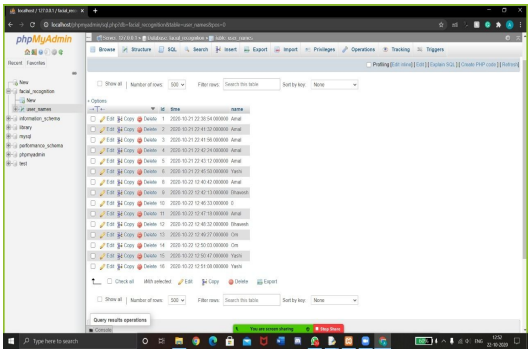

Figure 6: Attendance Database (Xampp Server)

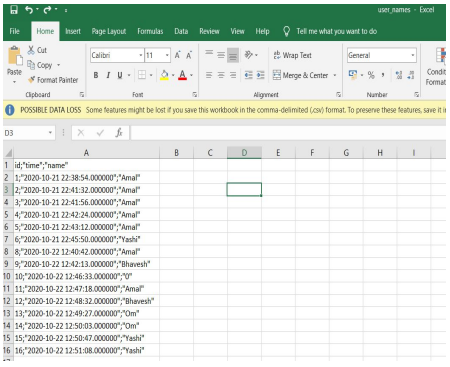

Figure 7: Attendance Excel Sheet

Here, we have created a database table, 'user_names', on the Xampp Server's Localhost. Once the face is recognized by the camera, the credentials (ID and Name) are passed to the attendance database. A row is created along with the name of the person and the timestamp, which indicates the time and date when he/she enters the room. If no face/unknown face is detected, an entry with the name '0' is made.

The teacher can create an excel sheet from the database table for the record purposes.

Chapter 5
Conclusion

Conclusion/Learning from the project:

Before the development of this project, there were many loopholes in the process of taking attendance using the old method which caused many troubles to most of the institutions. Therefore, the facial recognition feature embedded in the attendance monitoring system can not only ensure attendance to be taken accurately and also eliminate the flaws in the previous system. As we all know that face is an important identity of each and every person, with the concept of face recognition our project describes the importance of automation in the running era. Using technology to conquer the defects cannot merely save resources but also reduces human intervention in the whole process by handling all the complicated tasks to the machine.

The names of the recognized students are then updated in the excel sheet. This system could be used for attendance marking of the students and staff in any organization. This system saves time and manual effort otherwise required to put by the lecturer. This system helps lecturers' to efficiently manage a large number of students present in the classroom. The system will also help prevent a large number of students from skipping the daily classes.

Chapter 6
Appendices

6.1 Codes

Code for 01_face_dataset.py:

```
import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# For each person, enter one numeric face id
face_id = input('\n enter user id end press <return> ==>  ')

print("\n [INFO] Initializing face capture. Look the camera and wait ...")
# Initialize individual sampling face count
count = 0

while(True):

    ret, img = cam.read()
    #img = cv2.flip(img, -1) # flip video image vertically
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_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)
        count += 1

        # Save the captured image into the datasets folder
        cv2.imwrite("dataset/User." + str(face_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

        cv2.imshow('image', img)

    k = cv2.waitKey(100) & 0xff # Press 'ESC' for exiting video
    if k == 27:
        break
    elif count >= 30: # Take 30 face sample and stop video
         break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

Code for 02_face_training.py:

```
import cv2
import numpy as np
from PIL import Image
import os

# Path for face image database
path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer_create()
detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml");

# function to get the images and label data
def getImagesAndLabels(path):

    imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
    faceSamples=[]
    ids = []

    for imagePath in imagePaths:

        PIL_img = Image.open(imagePath).convert('L') # convert it to grayscale
        img_numpy = np.array(PIL_img,'uint8')

        id = int(os.path.split(imagePath)[-1].split(".")[1])
        faces = detector.detectMultiScale(img_numpy)

        for (x,y,w,h) in faces:
            faceSamples.append(img_numpy[y:y+h,x:x+w])
            ids.append(id)

    return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...")
faces,ids = getImagesAndLabels(path)
recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml
recognizer.write('trainer/trainer.yml')

# Print the numer of faces trained and end program
print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))
```

Code for 03_face_recognition.py:

```
import cv2
import numpy as np
import os
import time
import datetime
import mysql.connector

recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.read('trainer/trainer.yml')
cascadePath = "haarcascade_frontalface_default.xml"
faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT_HERSHEY_SIMPLEX

#iniciate id counter
id = 0

# names related to ids: example ==> Yashi: id=1,  etc
names = ['None', 'Amal', 'Yashi', 'Bhavesh', "Sukruti Ma'am ", 'Om' , 'Vatsal']

# Initialize and start realtime video capture
cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video widht
cam.set(4, 480) # set video height

# Define min window size to be recognized as a face
minW = 0.1*cam.get(3)
minH = 0.1*cam.get(4)

while True:

    ret, img =cam.read()
    #img = cv2.flip(img, -1) # Flip vertically

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )
```

```
    for(x,y,w,h) in faces:

        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            confidence = "  {0}%".format(round(100 - confidence))
        else:
            id = "unknown"
            confidence = "  {0}%".format(round(100 - confidence))

        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

    cv2.imshow('camera',img)
    cv2.waitKey(3000)

    ts = time.time()
    timestamp = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')
    mydb = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="facial_recognition"
    )
    mycursor = mydb.cursor()
    mycursor.execute("""INSERT into user_names (time,name) values(%s,%s)""",(timestamp,id))
    mydb.commit()
    mydb.close()
    break

    #k = cv2.waitKey(10) & 0xff # Press 'ESC' for exiting video
    #if k == 27:
    #break

# Do a bit of cleanup
print("\n [INFO] Exiting Program and cleanup stuff")
cam.release()
cv2.destroyAllWindows()
```

Chapter 7
References

References:

[1] S. Sawhney, K. Kacker, S. Jain, S. N. Singh and R. Garg, "Real-Time Smart Attendance System using Face Recognition Techniques," *9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, *Noida, India, pp. 522-525,* 2019.

[2] Divyarajsinh Parmar & Brijesh Mehta, "Face Recognition Methods & Applications", *International Journal of Computer Technology and Applications, vol.4, p.p. 84-86,* 2014.

[3] Borra Surekha, Kanchan Jayant Nazare, S. Viswanadha Raju and Nilanjan Dey, "Attendance Recording System Using Partial Face Recognition Algorithm," *Intelligent Techniques in Signal Processing for Multimedia Security Studies in Computational Intelligence, vol 660. Springer.*

[4] Sakshi Patel, Prateek Kumar, Shelesh Garg, Ravi Kumar, "Face Recognition based smart attendance system using IOT", *International Journal of Computer Sciences and Engineering, Vol.6, Issue.5, pp.871-877, 2018.*

[5] Saras Chandu Gaddam and N. V. K. Ramesh, "Attendance Management and User Security System based on Eigen Faces Algorithm using Raspberry pi 2 and Ethernet", *Indian Journal of Science and Technology, Vol 9(17), May 2016.*

[6] Sudhir Bussa , Ananya Mani , Shruti Bharuka , Sakshi Kaushik, "Smart Attendance System using OPENCV based on Facial Recognition," *International Journal Of Engineering Research & Technology (Ijert) Volume 09, Issue 03, March 2020.*

[7] Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, "Facial Recognition Attendance System Using Python and OpenCV," *Quest Journals Journal of Software Engineering and Simulation Volume 5 ~ Issue 2, pp: 18-29, 2019.*

[8] "Image Processing with Python", Coursera Project Network, *url: https://www.coursera.org/projects/image-processing-with-python*

[9] "Introduction to Scripting in Python", Rice University on Coursera, *url: https://www.coursera.org/specializations/introduction-scripting-in-python*

[10] "Python MySQL", W3Schools.com,
url: https://www.w3schools.com/python/python_mysql_getstarted.asp