ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION

A MINI PROJECT REPORT

Submitted by

Evana Elsa Bose (RET19CS076)
Haleema P P (RET19CS088)
Harisankar M (RET19CS089)
Haritha Koorkkaparambil Rajeev (RET19CS090)

Under the guidance of

Mr. Harikrishnan M

to the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology
in
Computer Science and Engineering

Department of Computer Science & Engineering
Rajagiri School of Engineering and Technology(Autonomous)
Rajagiri Valley, Kakkanad, Kochi, 682039
July 2022

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RAJAGIRI SCHOOL OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)

RAJAGIRI VALLEY, KAKKANAD, KOCHI, 682039

RSET
RAJAGIRI SCHOOL OF ENGINEERING & TECHNOLOGY
(AUTONOMOUS)

CERTIFICATE

*Certified that Mini Project work entitled "**Attendance Management System using Face Recognition**" is a bonafide work done by **Evana Elsa Bose (RET19CS076), Haleema P P (RET19CS087), Harisankar M (RET19CS089), Haritha Koorkkaparambil Rajeev (RET19CS090)** of Sixth semester in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the APJ Abdul Kalam Technological University during the academic year 2021-2022.*

Mr. Harikrishnan M	**Ms. Sangeetha Jamal**	**Dr. Dhanya P M**
Project Guide	**Project Coordinator**	**Head of Department**

Place : Kakkanad

Date : 29 July, 2022

ACKNOWLEDGEMENT

We are extremely honored to thank Prof. (Dr). P.S. Sreejith, Principal, RSET who has always made sure that our ladder to success was leaning against the right wall. We thank him for his constant help and support. We thank Dr. Dhanya P M, HoD, Department of Computer Science & Engineering, RSET whose help and guidance has been a major factor in completing our journey.

We express our gratitude to our Project Co-ordinator and Class-In-Charge, Ms. Sangeetha Jamal, Asst. Professor, Dept. of Computer Science Engineering for her support and timely help whenever it was required. She has been a great mentor without whom this would have been incomplete.

We extend our sincere and heartfelt thanks to our guide Mr. Harikrishnan M, Asst. Professor, Dept. of Computer Science & Engineering, RSET for taking the time and effort to review our work and providing valuable advice and feedback from time to time. Last but not the least we would like to express our heartfelt and sincere gratitude to our family and friends for their constant support throughout this entire journey.

Evana Elsa Bose, Haleema P P, Harisankar M, Haritha Koorkkaparambil Rajeev

ABSTRACT

Attendance Management system using face recognition is an automated attendance marking system that uses face recognition algorithms to mark the attendance. In the manual system, the attendance is marked manually where the teacher calls out the students name and when the student responds, the corresponding attendance for the student is marked. This redundant task is tedious and time consuming. It is prone to errors and malpractice like proxy. An automated attendance marking system can make this process easier. The main aim of this project is to reduce the flaws of existing systems with the help of machine learning.

This project aims to work by first recognizing students from the real time video. After that, the students recognized are compared to the one in the Image folder and the system will mark the attendance. This system identifies the students in the frame, detects and recognises faces and marks the attendance of the respective student in the text file. This process is done by the teacher when he/she clicks on a button called ‘CAPTURE’. Finally, the teachers and the students can view the attendance in their respective login portals and the teaching staffs can make any changes if any. New student details and images are added by the admin staff.

CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1. General Background

Face detection is a computer technology being used in a variety of applications that identifies human faces in digital images. Face detection applications use algorithms and ML to find human faces within larger images, which often incorporate other non-face objects such as landscapes, buildings and other human body parts like feet or hands. Face detection algorithms typically start by searching for human eyes -- one of the easiest features to detect. The algorithm might then attempt to detect eyebrows, the mouth, nose, nostrils and the iris. Once the algorithm concludes that it has found a facial region, it applies additional tests to confirm that it has, in fact, detected a face.

Haar Cascade is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper “Rapid Object Detection using a Boosted Cascade of Simple Features” published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper “Rapid Object Detection using a Boosted Cascade of Simple Features” published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.

Face recognition is a method of identifying or verifying the identity of an individual using their face. Face recognition systems can be used to identify people in photos, video, or in real-time.

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Cv2 is the module import name for opencv-python, "Unofficial pre-built CPU-only OpenCV packages for Python". The traditional OpenCV has many complicated steps involving building the module from scratch, which is unnecessary.

OpenCV has a function to read video, which is cv2.VideoCapture(). We can access our webcam using pass 0 in the function parameter.

1.2. Objective

The primary objective of this project is to mark the attendance of the students for a particular period with the help of face recognition.

1.3. Motivation

In colleges and universities, attendance marking has become an important criterion for marking a student's punctuality. In most cases, the marking of attendance is taken manually where the teacher would call out names and the student would have to speak up to mark their attendance before every lecture. This redundant task is tedious and time consuming when it comes to a class with more than 50 students. This method of attendance marking is also prone to errors and malpractice like proxy attendance. Hence a system where attendance of the students could be taken automatically without any human efforts and errors could be used to make the process of attendance easier.

1.4. Summary of Report

The main goal of this project is to provide an easier way to mark attendance avoiding all the errors involved in the manual process. It makes the lives of the teaching staff easy as the redundant task of manually taking the attendance is eliminated. Time which was initially wasted in manually calling and marking the attendance is saved here.

Chapter 2

LITERATURE SURVEY

2.1. Paper 1

FACE RECOGNITION BASED CLASS MANAGEMENT AND ATTENDANCE SYSTEM

The main objective of this paper is the implementation of an automated attendance system which uses the face recognition algorithms to record the attendance of the class and manage the class database.
Apart from the manual system, there are some systems that use technology like fingerprint system, iris scanning, etc. but what makes the system proposed here stands out is that it is a one-stop system to manage and record class attendance.
Steps of working of the system:
1. Add the record of each student
2. Training process-The images of the student are trained using LBPH and Haar Cascade and saved in the form of a YML file.
3. Tracking process - database of trained images is compared with the student's face to track the student's face.
4. The attendance is marked in the CSV.
The system is developed using Python: GUI, PDBC using sqlite3, and OpenCV. The following is the detailed methodology of the proposed system:

A. Creation of a database: We create the database for storing the details of the student (name, roll no,etc) of the students of a particular class. Here, the record of the student is to be added by clicking on the "Add Details" button. To one-stop the image of the student for the respective name click on the "Take Images" button. Using OpenCV, a real-time live video of the student will be recorded and split into sixty frames. These sixty frames are converted into grayscale and stored as student's face images. The "Status" label gives an update on the successful execution of database creation and image training process.

B. Image Training: To train the image, the database is already constructed which consists of an image and the image name. The Haar Cascade algorithm is used to calculate the histogram value of each image stored in the database. The database created by the image training process is saved as "trainer.yml" i.e in the form of a YML file such that it can be used in the future.

C. Image Tracking: The LBPH compares the histogram of the existing images in the database with the current real-time video of the student. The histogram of the current student video when matched with the existing database will display the student detail as "roll number: name" on the

detected face. In case the student is not recognised it will label the student as “Unknown”. The database of unknown images is saved in the respective folder.

D. Attendance Report: Once the image of the student is detected among the existing database, the attendance of the student is marked. The “Attendance Report” label gives an update of the attendance record. These four steps are to be used only during adding a new student in the database. For recording the attendance of the existing student the professor only needs to click on the “Track Images” using a laptop.

Advantages:

- Reliable and efficient system.(because of using LBPH and Haar Cascade algorithm)
- The system can give an accuracy of about 96.88%.
- This system improves the productivity of the class since there is no longer any source of disturbance caused by taking attendance manually

Disadvantages:

- To get better results and accuracy, the class should be well illuminated.

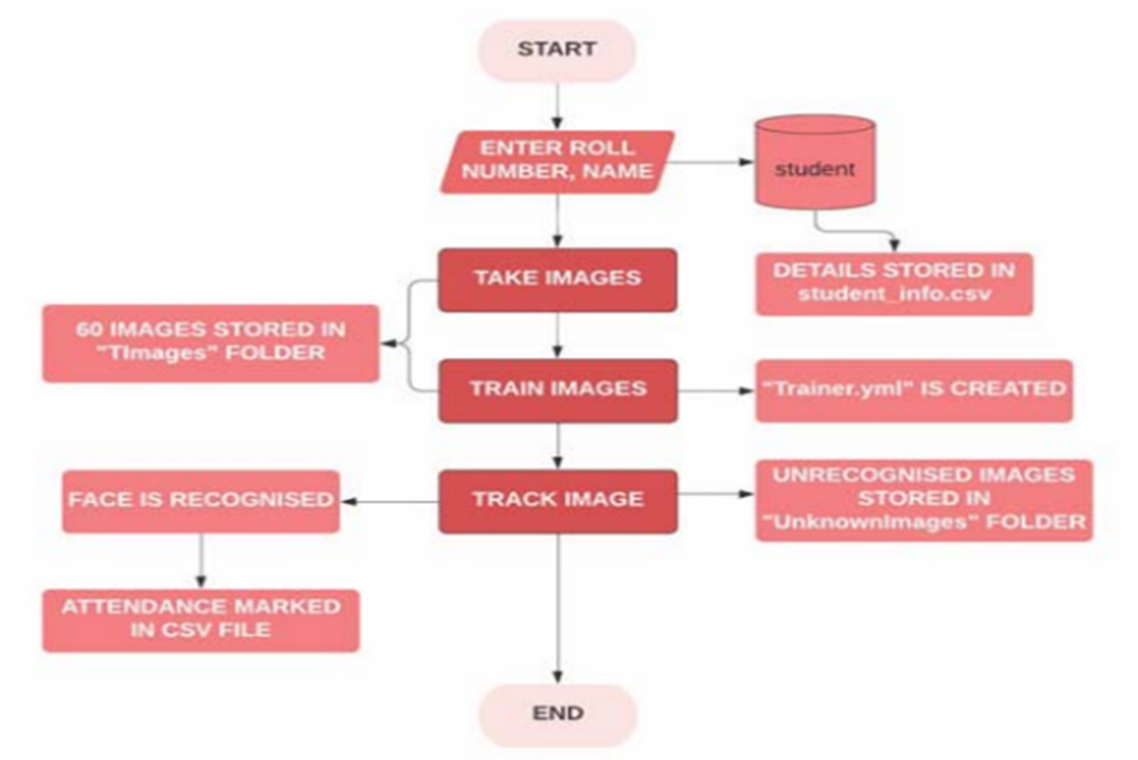


Figure 2.1. Flowchart of the system

2.2. Paper 2

ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION

In the system proposed in this paper, initially all the students will be enrolled by storing their facial images with a unique ID. At the time of attendance, real time images will be captured and the faces in those images will be matched with the faces in the pre-trained dataset.

Face detection – Haar cascade algorithm helps in the face detection part. To simplify the As Haar cascade uses 24*24 windows there can be 160000+ features within a detector which needs to be calculated. AdaBoost is an algorithm based on machine learning. This algorithm finds the best features among all the possible features and eliminates the irrelevant features work of calculating the feature values, an Integral image algorithm is introduced. But this paper proposes to use Paul-Viola Jones Algorithm for face detection.

For the collection of the dataset, a minimum of 20 images are captured per individual student along with a unique ID.

Face recognition and training the stored dataset - Local Binary Patterns Histogram (LBPH) algorithm.

Histogram generated for stored images and real time images. Then the histograms are compared for face recognition and the one with lower difference is taken.

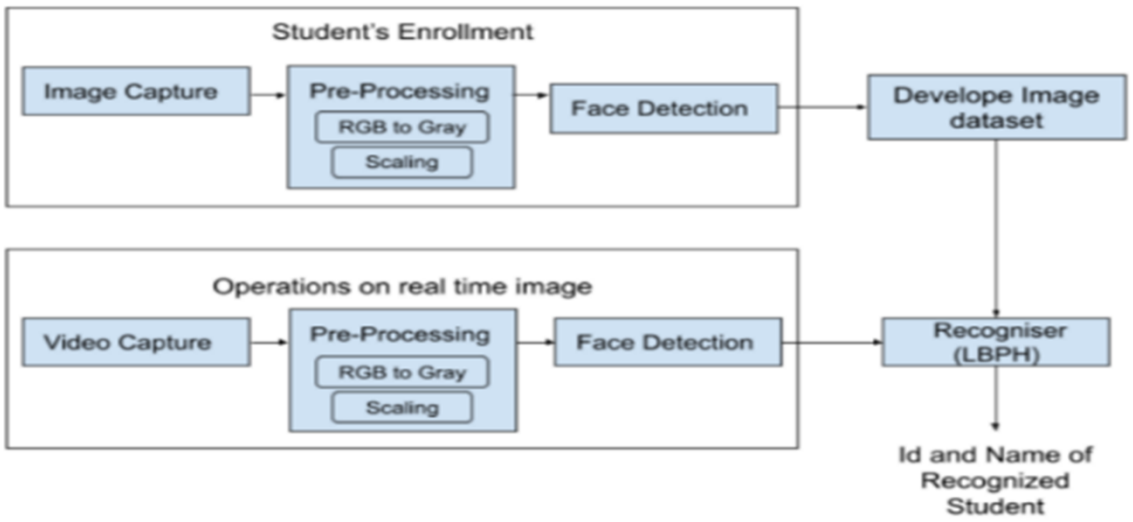


Figure 2.2. Proposed Methodology for Attendance Monitoring

2.3. Paper 3

STUDENT ATTENDANCE SYSTEM USING FACE RECOGNITION

The system proposed in this paper tackles all the problems of the traditional attendance marking system. This proved to be an efficient and robust device for taking attendance in a classroom without any time consumption and manual work.

Here two or more cameras are used depending on the need and size of the classroom and images are captured. These images serve as input to the system. Here, when it is time to take attendance , the teacher would trigger the system by clicking on the start button thereby the system will undergo face detection. Also, the blurred images are upgraded using Generative Adversarial Networks. These upgraded images are sent to the system for face detection.

Feature extraction & face recognition- Gabor filter.

Face recognition is done using the K-nearest neighbour algorithm, Convolutional neural networks, and SVM Algorithm.

After recognition the name and identification number of the person appears and attendance marked on the excel sheet.

As there can be multiple cameras, chances of redundancy are high. So, redundancy is removed before marking the attendance.

The system was tested on three different algorithms out of which the KNN algorithm proved to be better with the accuracy of 99.27 %.

KNN proved to be better by achieving the overall accuracy of 97 %. When tested on conditions listed above CNN achieved the overall accuracy of 95 % and SVM achieved an accuracy of 88 %. viewing the aspect of time complexity.

The proposed system is tested on 200 real-time images of a classroom with a maximum strength of 70 students. The proposed system is robust enough to take attendance of 70 students in a classroom.

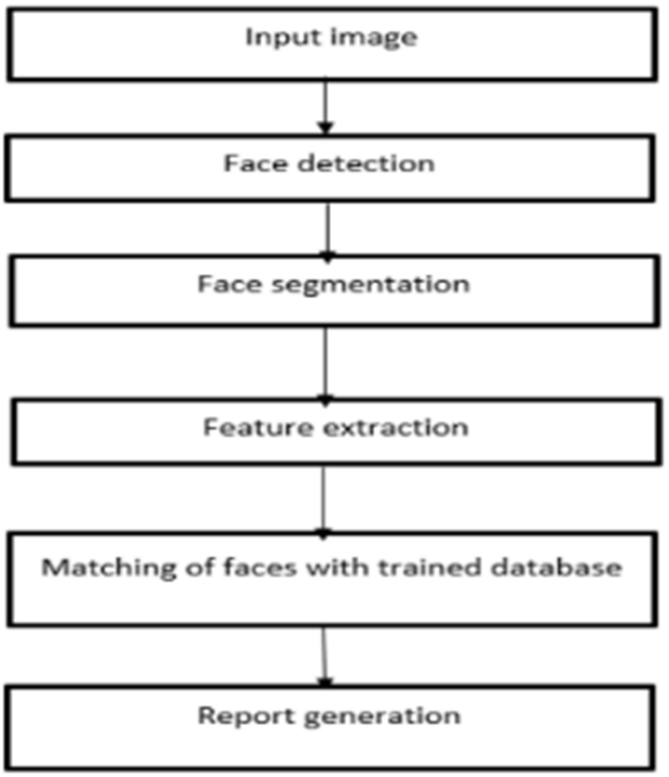


Figure 2.3. Operating Process of attendance system

2.4. Paper 4

FACE RECOGNITION BASED ATTENDANCE MANAGEMENT SYSTEM

This system eliminates the time-consuming manual attendance system by replacing it with the concept of face recognition. This system consists of four phases- database creation, face detection, face recognition and attendance updation. Database is created by the images of the students in class.

Face detection and recognition is performed using the Haar-Cascade classifier and Local Binary Pattern Histogram algorithm respectively. Faces are detected and recognized from live streaming video of the classroom.

Dataset Creation- Images of students are captured using a webcam. Multiple images of a single student will be acquired with varied gestures and angles. These images undergo pre-processing and are cropped to obtain the Region of Interest (ROI) which will be further used in the recognition process.

Face detection- Haar-Cascade Classifier with OpenCV.

Face Recognition- Face recognizer used in this system is Local Binary Pattern Histogram. Initially, the list of local binary patterns (LBP) of the entire face is obtained. These LBPs are converted into decimal numbers and then histograms of all those decimal values are made. At the end, one histogram will be formed for each image in the training data.

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.

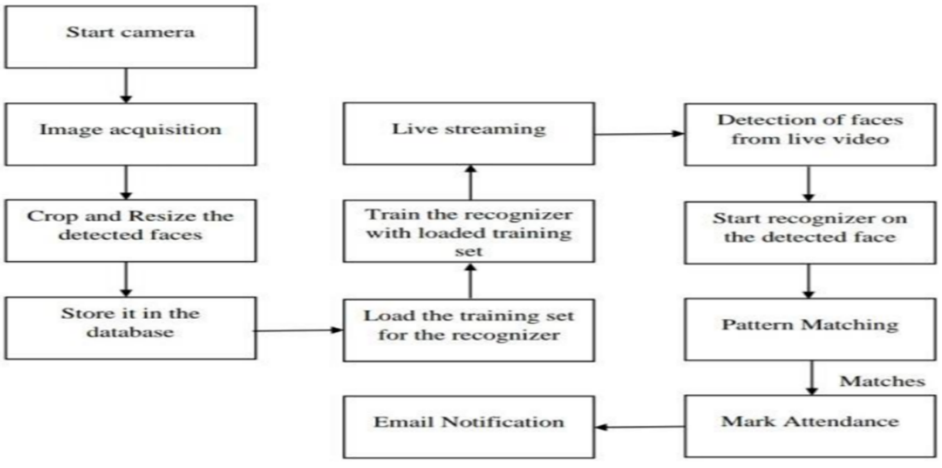


Fig.1. System Architecture

Figure 2.4. System Architecture

2.5. Paper 5

FACIAL RECOGNITION ATTENDANCE SYSTEM USING PYTHON AND OPENCV

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 an excel sheet. At the end of the day, the excel sheet containing attendance information regarding all individuals is mailed to the respective faculty.

Software used- OpenCV and Numpy through Python

There are two major system flows in the software development section as shown below-

1. The creation of the face database
2. The process of attendance taking

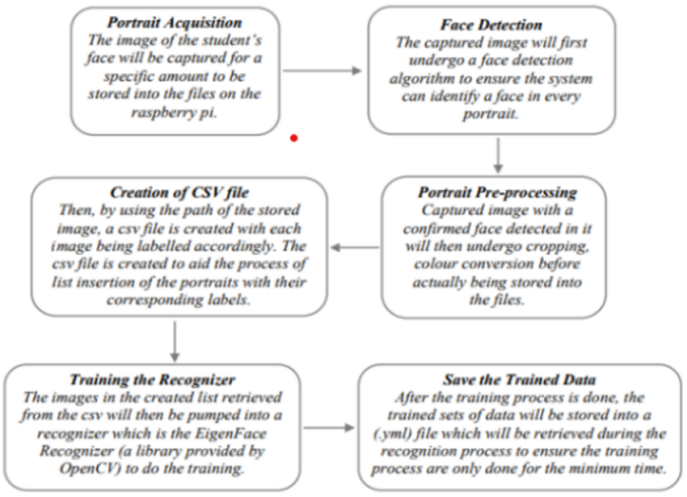


Figure 2.5.1. Creation of the face database

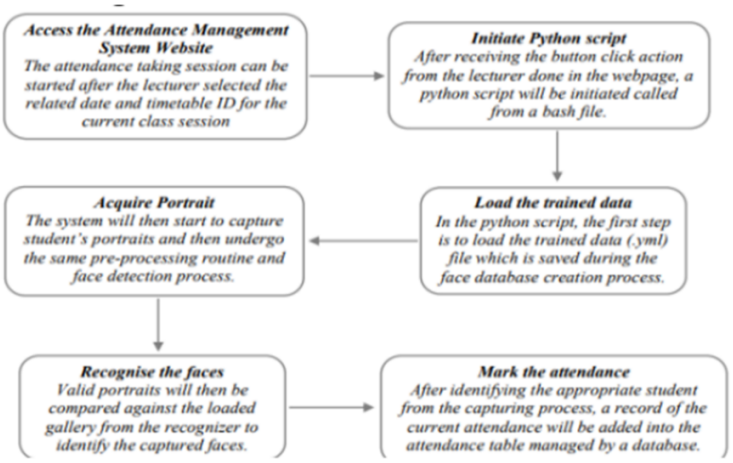


Figure 2.5.2. The process of attendance taking

Chapter 3

PROPOSED METHOD

3.1. Problem Definition

The main aim of this project is to reduce the flaws of existing systems with the help of machine learning. This project aims to work by first reading the input image and pre-process the image, in which an unwanted element in the face is removed. After that, the image is compared to the one in the database and the system will mark the attendance.

3.2. Scope of the work

This system intends to provide an easier way to mark attendance. It makes the lives of the teaching staff easy as the redundant task of manually taking the attendance is eliminated. Time which was initially wasted in manually calling and marking the attendance is saved here. This system takes the image of the class during the start of a particular period, detects and recognises faces and marks the attendance of the respective student in the database. The image is captured by the teacher when the teacher clicks on a button called 'CAPTURE'. Finally, the teachers and the students can view the attendance in their respective login portals and the teaching staffs can make any changes if any. New student details and images are added by the admin staff.

3.3. Methodology

3.3.1. Taking image of the class

For the attendance taking process, the image of the class is taken using a webcam. This image undergoes various preprocessing and faces of the students are detected, recognized and their attendance is marked.

3.3.2. Detection of faces

The detection algorithm used in this program is HaarCascade. It detects multiple faces of students and their face coordinates are stored in the form of a list.
The user would view the detected faces as a rectangular box around the faces detected. In this program, the detection algorithm is used for registering a student image and also used during face recognition for marking the attendance.

3.3.3. Face recognition

The algorithm used for face recognition is CNN algorithm. This algorithm recognizes a particular student uniquely and attendance is marked in a text file.

During the registering process, the faces of students are stored in folders where the folder name is the name of the student. The detected faces during the attendance marking session are compared to the images present in each folder. The folder names i.e. the names of the students are marked in text files that are stored in each subject folder.

3.4. System Architecture

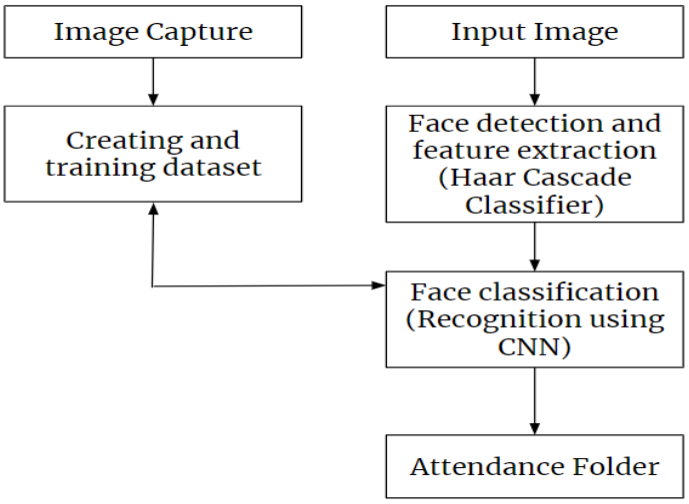


Figure 3.4. System Architecture

3.5. Design of the system

3.5.1. Front End Design

The front-end design includes a gui for the user interface part. It consists of 3 login pages for 3 different users- admin, student and faculty. The password and the details are all stored in the database. To implement this, we use python TKinter and we implemented it using the platform visual studio code.

When the admin logs in, he/she has the provision to register a student and view the attendance.

When the faculty logs in, he/she has the provision to mark the attendance and view the attendance.

When the student logs in, he/she has the provision to view attendance.

3.5.2. Back End Design

The back-end design consists of the face detection and recognition code that is coded on the platform visual studio code. The code was implemented using Python 3.10. The face detection code is

implemented using the Haar Cascade algorithm. Face recognition is implemented using CNN algorithm. They are called as functions when needed.

3.5.3. Overall System Design

The overall system design includes a GUI for user interface. It also includes a file system for marking attendance and storage of student images. There is also a database that stores the details of the students, faculty and the admin.

When the student is registered, the student details are added to the database and the image of the student is added to the file system. The file system consists of a folder whose name is the name of the student and the image of the student is stored inside the folder.

During the recognition part, the faces are detected and identified in real time and these details are then passed to a text file of that particular subject. This can be viewed from the file directory.

3.6. Module Division

3.6.1. Face Detection and Feature Extraction

The Haar Cascade algorithm is used for face detection and feature extraction. This algorithm helps us to detect multiple faces from an image and store the detected image. Registration of a new student is also done using this algorithm. Images of students are captured and stored in separate folders that are created for each student (dataset creation). This dataset is used for training the model.

3.6.2. Face Classification (Recognition)

The CNN algorithm is used for face recognition. Before recognition, the faces that are detected and stored in folders are trained using this same algorithm. Then at the time of attendance the image of the class is taken and faces are detected from the image. The detected faces are compared with the faces present in the folders and attendance is marked. This comparison of images detected and from the folders is done by face recognition algorithms.

3.6.3. GUI

The user interface is implemented using tkinter. There are 3 users - Admin, Teacher and Student. In the opening page, the user has the provision to log in either as admin, teacher, or as student. Login details of each user are stored in a database. There are separate login pages for each user. Once logged in successfully, the admin can register new students to the system and he/she can also view the attendance of a particular class. The teacher can take attendance of the students by capturing the image of the class and can also view and update the attendance. Students can only view their attendance.

3.7. Algorithms

3.7.1. Detection Code Algorithm

STEP 1 – Importing OpenCV
STEP 2 – Importing XML file
STEP 3 – Allowing Webcam to capture video
STEP 4 – Capturing video in terms of frames
STEP 5 – Converting the image to grayscale
STEP 6 – Detecting Multi-scale faces
STEP 7 – Mentioning sides of the rectangle for face detection
STEP 8 – Displaying the detected frames in terms of real-time video

3.7.2. Recognition Code Algorithm

STEP 1 – Importing OpenCV
STEP 2 – Importing XML file
STEP 3 – Allowing Webcam to capture video
STEP 4 – Capturing video in terms of frames
STEP 5 – Converting the image to grayscale
STEP 6 – Detecting Multi-scale faces
STEP 7 – Mentioning sides of the rectangle for face detection
STEP 8 – Displaying the detected frames in terms of real-time video

3.8. Implementation

Implementation of this project was using Python 3.8 which was coded on the platform visual studio code.

3.8.1. Software Utilized

- Visual Studio Code
- Python version 3.8

3.8.2. Behind the Scene

The project is done using Python 3.8 version. First, the images of students are taken using the webcam, features extracted and students are registered in the system. Each student's image is stored in folders which are later used for recognition. Images of students taken are then used to train the model. During attendance, the image of students(or class) is taken using the webcam and the faces of students are detected. The detected images are then compared with the registered image and faces are recognized. Best match is found if the difference is less during the comparison. Attendance is marked in a text file if matches are found.

3.8.3. User interaction with the Application

A GUI was implemented for user interaction. Three types of users were expected-Admin, Teacher and Student. Each of them have their respective login page. Admin has the provision to add a student and view attendance. Faculty has the provision to mark attendance and view attendance. Students have the provision to view the attendance.

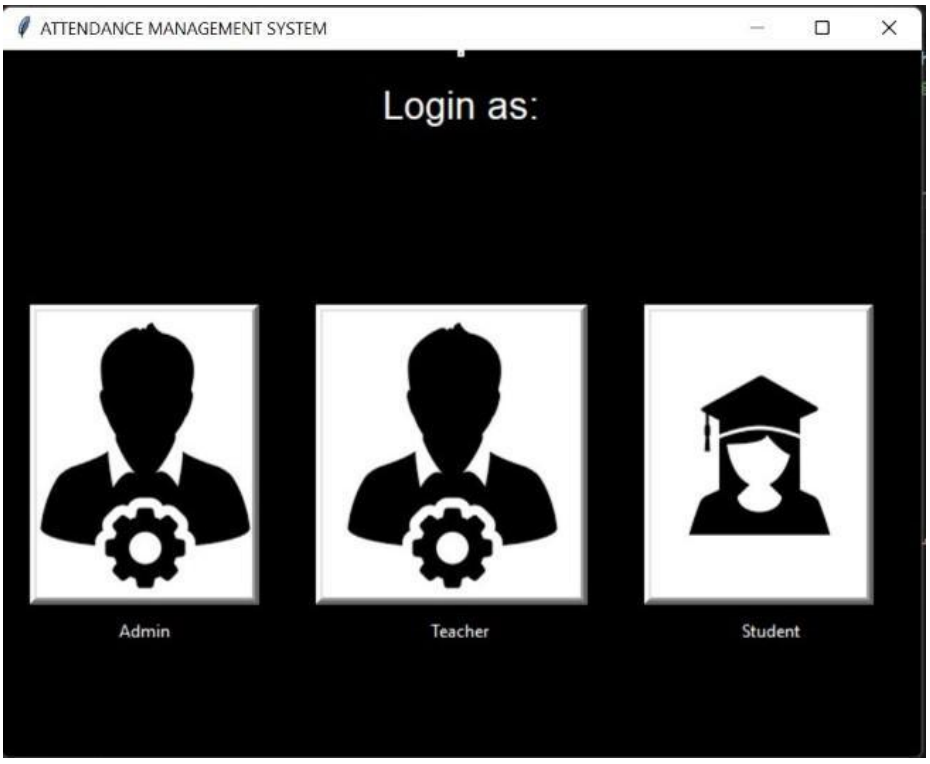
Chapter 4

RESULTS AND DISCUSSIONS

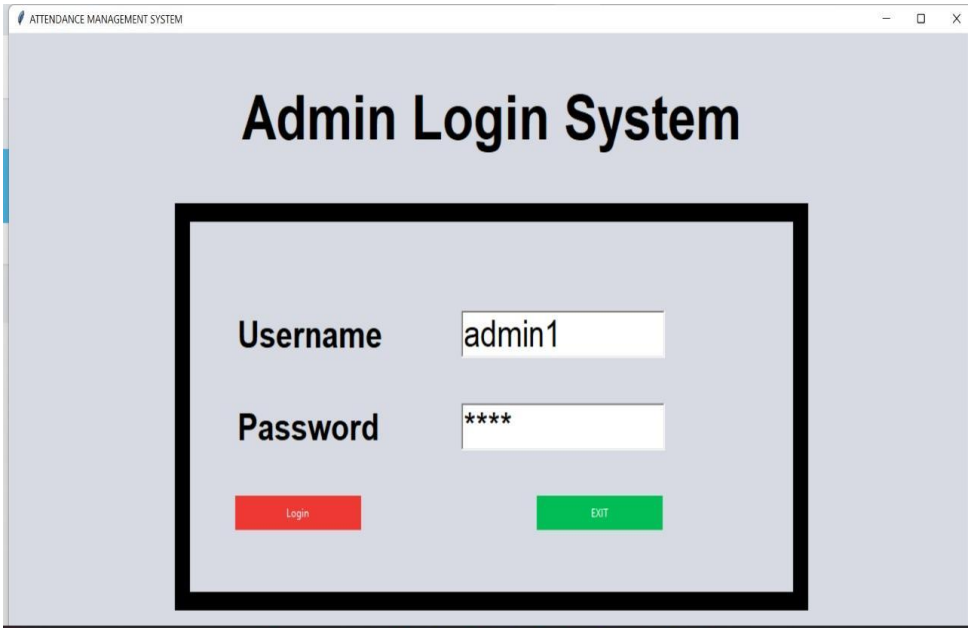
Part 1: GUI

1. Opening Page:

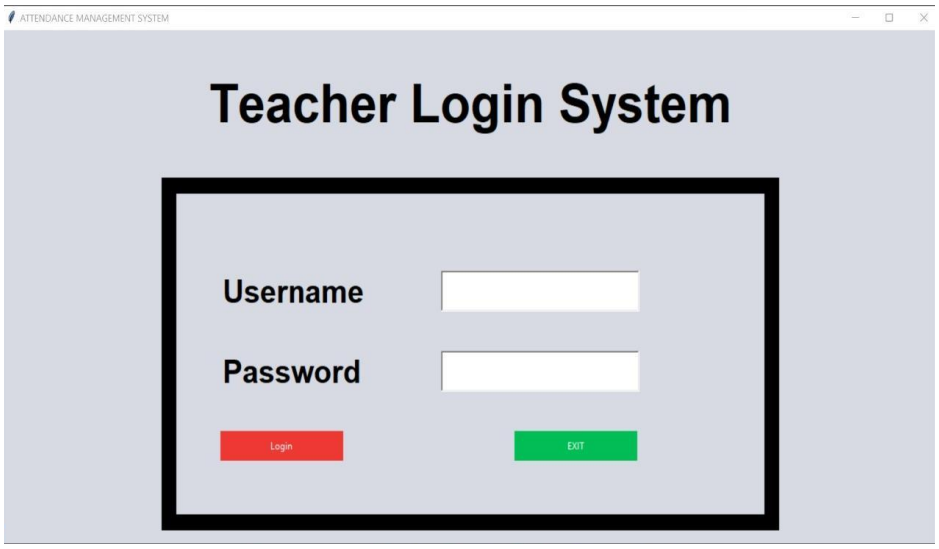
   The initial GUI page provides a provision to the user to login as either an Admin, Teacher or a student



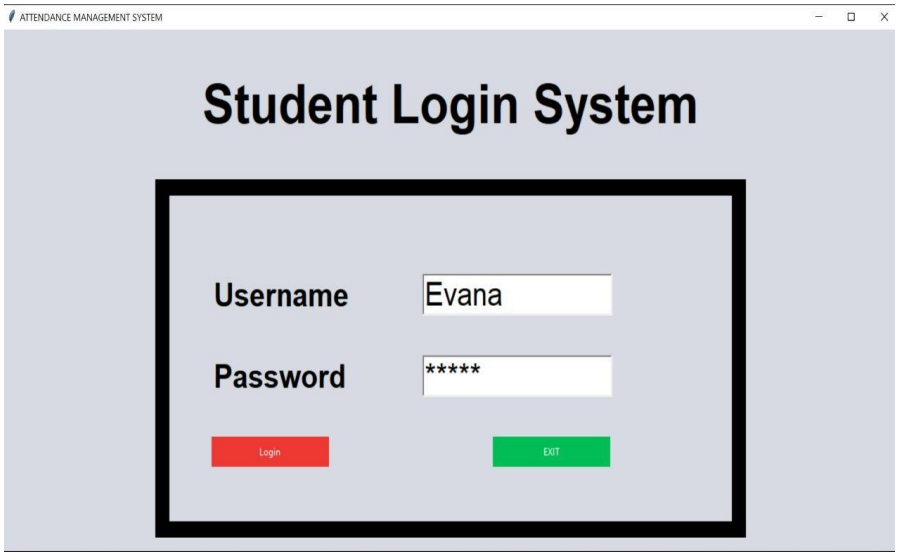
2. Admin redirected from the Admin button of initial page



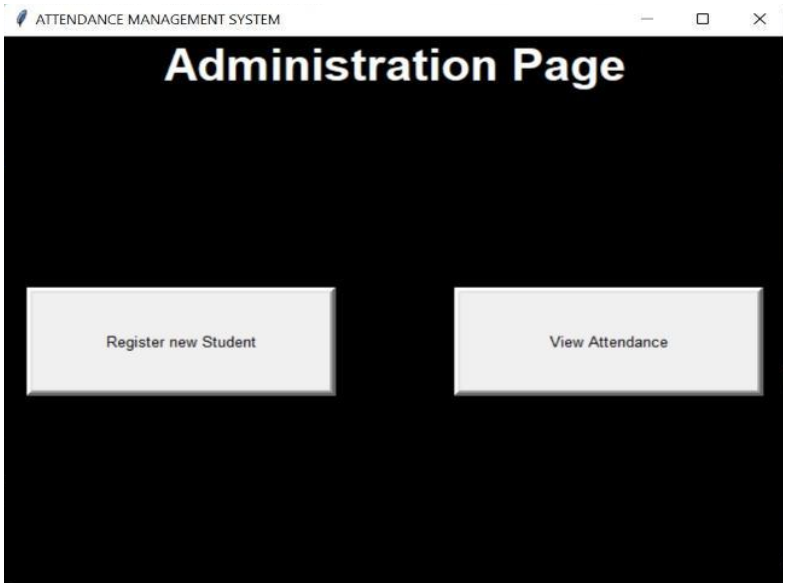
3. Teacher redirected from the Teacher button of initial page



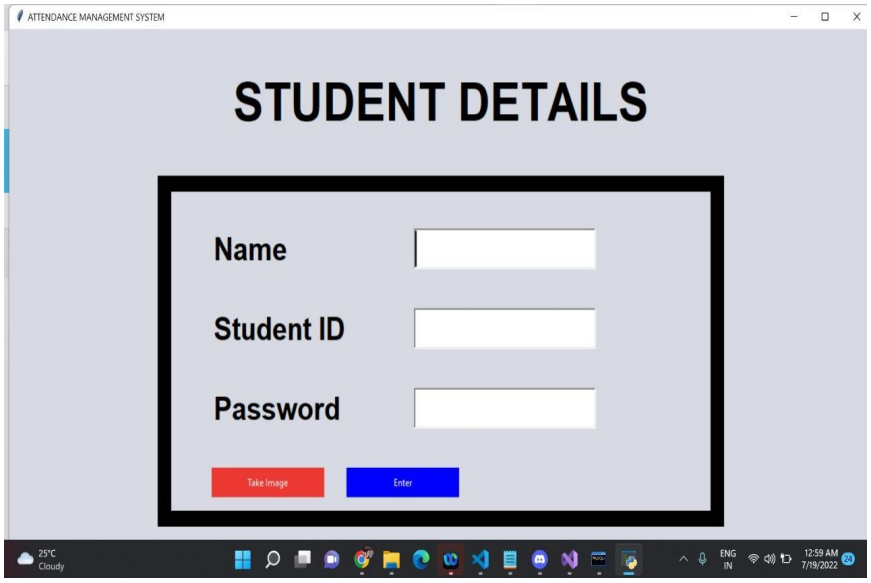
4. Student redirected from Student button of the initial page



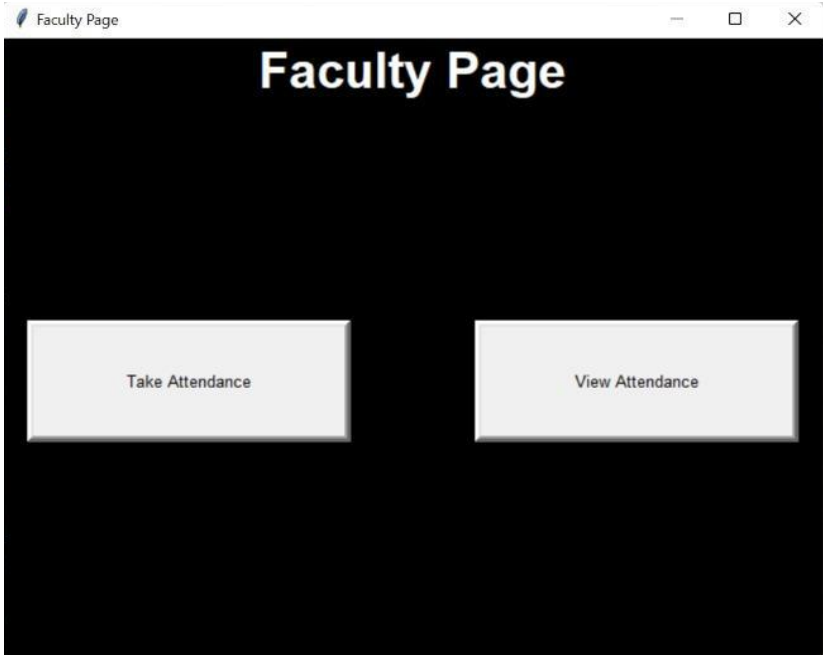
5. Admin Front page redirected from the admin login page



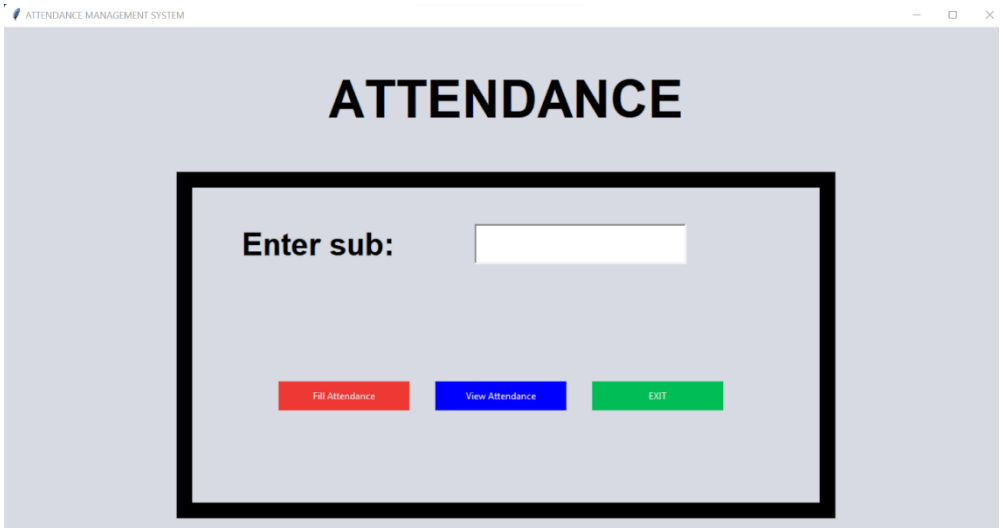
6. Register the student details redirected from ‘Register New Student’ from the admin front page



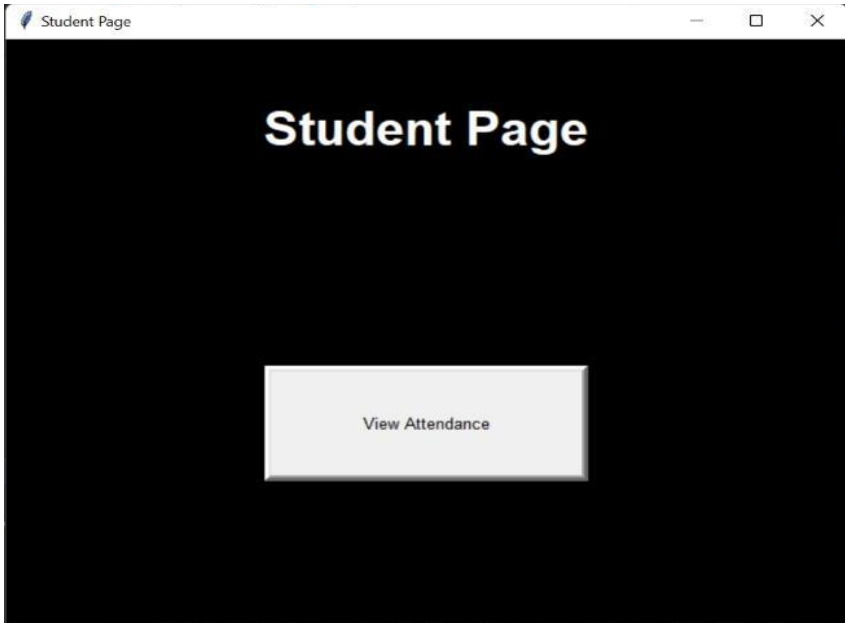
7. Faculty front page redirected from the Faculty login page



8. Taking attendance redirected from the ‘Take Attendance’ button from the faculty front page



9. Student front page redirected from students login page



10. View attendance redirected from the front page of the student login page

Part 2: Database

Three tables are implemented:

- Admin Table

```
mysql> desc admin;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| USERNAME | varchar(30) | YES  |     | NULL    |       |
| PASSWORD | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

- Teacher Table

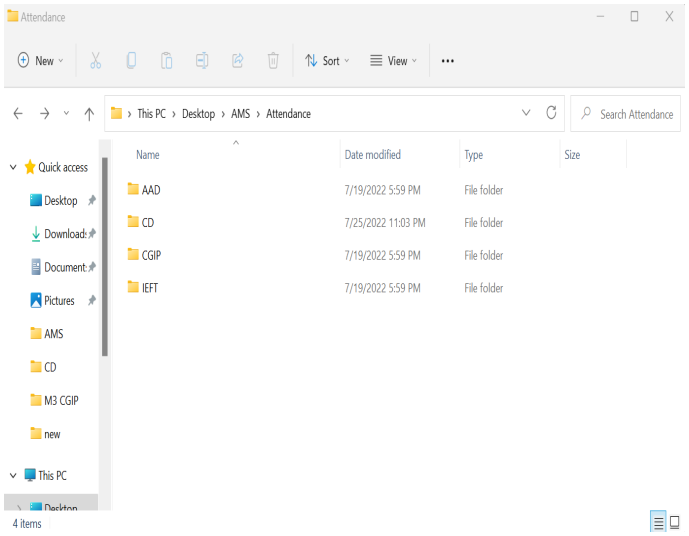
```
mysql> desc teacher;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| TID      | varchar(10) | NO   | PRI | NULL    |       |
| NAME     | varchar(20) | YES  |     | NULL    |       |
| SUBJECT  | varchar(10) | YES  |     | NULL    |       |
| PASSWORD | varchar(20) | NO   |     | NULL    |       |
+----------+------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

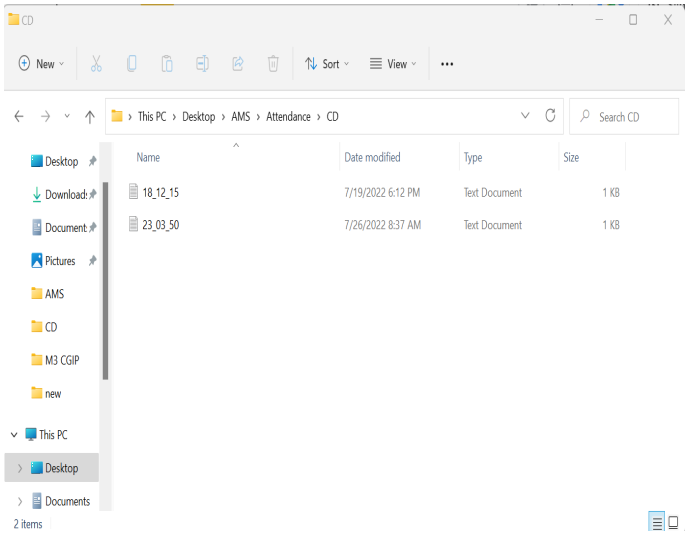
- Student Table

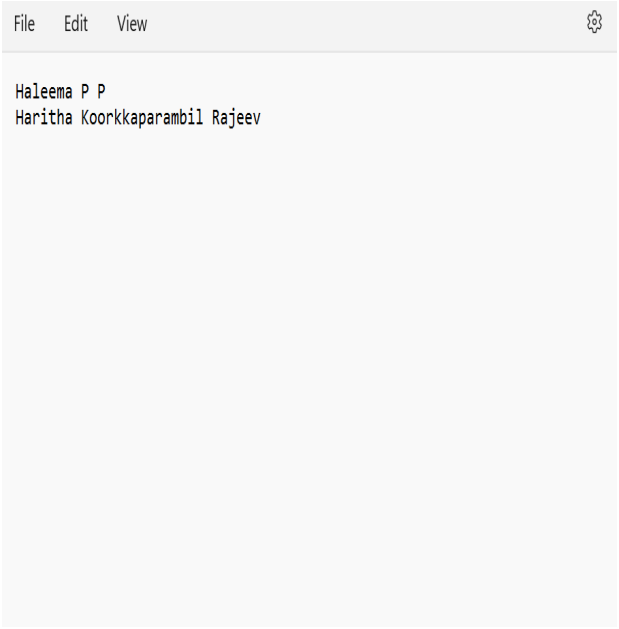
```
mysql> desc student;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| SID      | varchar(10) | NO   | PRI | NULL    |       |
| NAME     | varchar(20) | YES  |     | NULL    |       |
| PASSWORD | varchar(10) | NO   |     | NULL    |       |
+----------+------------+------+-----+---------+-------+
3 rows in set (0.37 sec)
```

Part 3: File System Organization

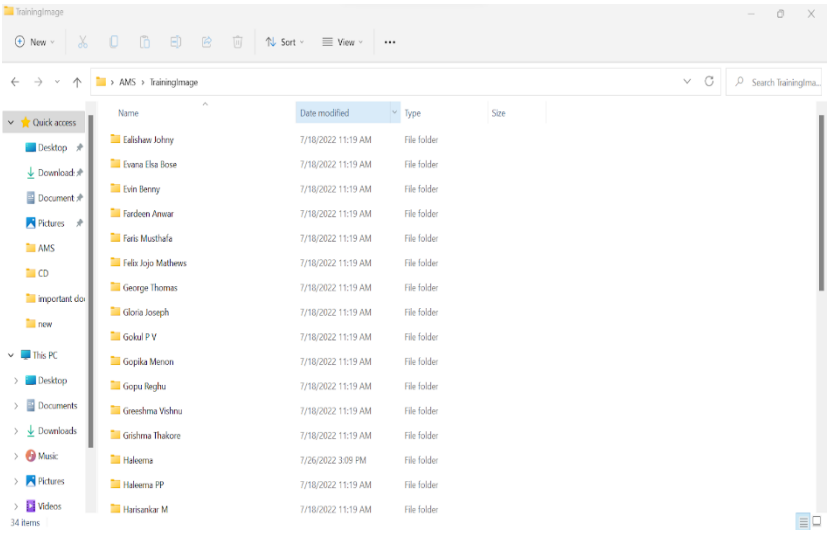
- File directory that stores the attendance details for every subject

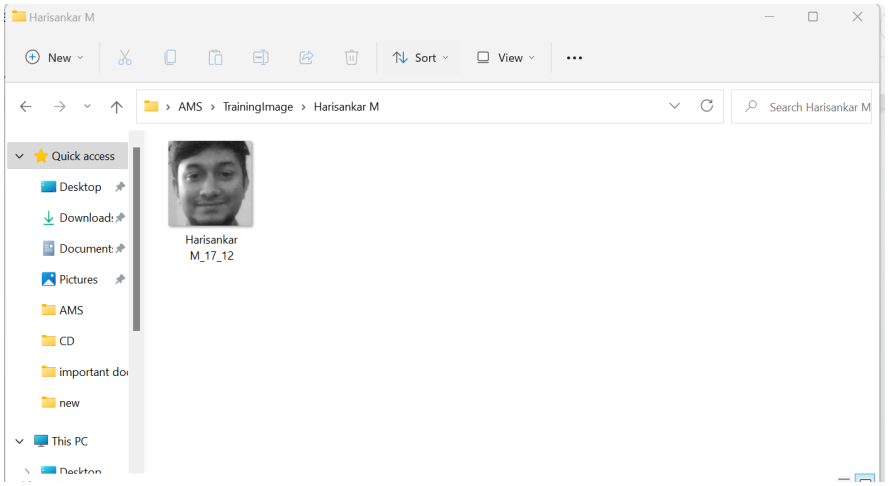

Attendance
New
Sort
View
This PC › Desktop › AMS › Attendance
Search Attendance
Quick access
Desktop
Downloads
Documents
Pictures
AMS
CD
M3 CGIP
new
This PC
Name
Date modified
Type
Size
AAD
7/19/2022 5:59 PM
File folder
CD
7/25/2022 11:03 PM
File folder
CGIP
7/19/2022 5:59 PM
File folder
IEFT
7/19/2022 5:59 PM
File folder
4 items


CD
New
Sort
View
This PC › Desktop › AMS › Attendance › CD
Search CD
Desktop
Downloads
Documents
Pictures
AMS
CD
M3 CGIP
new
This PC
Desktop
Documents
Name
Date modified
Type
Size
18_12_15
7/19/2022 6:12 PM
Text Document
1 KB
23_03_50
7/26/2022 8:37 AM
Text Document
1 KB
2 items



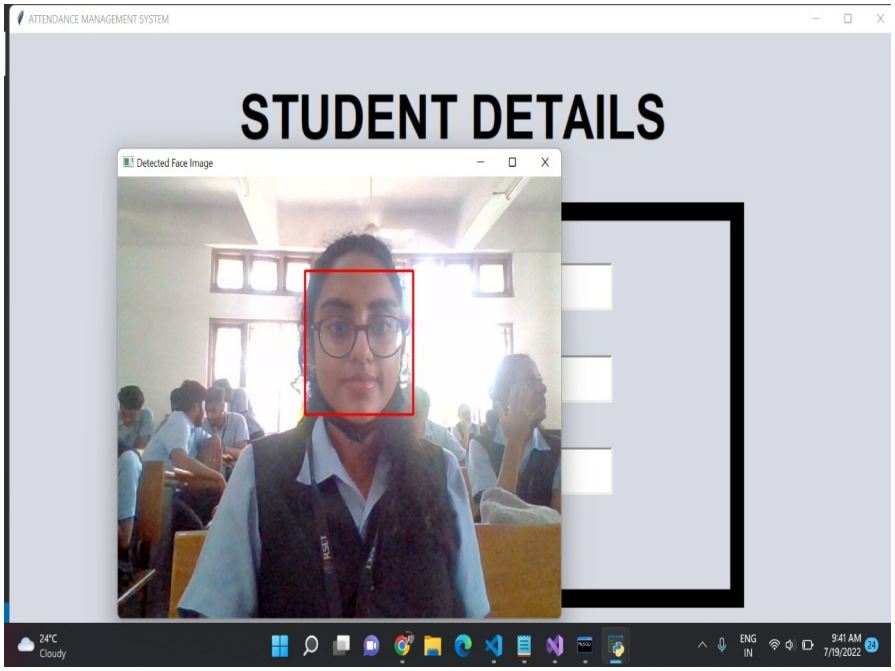
- File directory that stores the images of the student





Part 4: Face detection and Recognition

- Face Detection



- Face Recognition

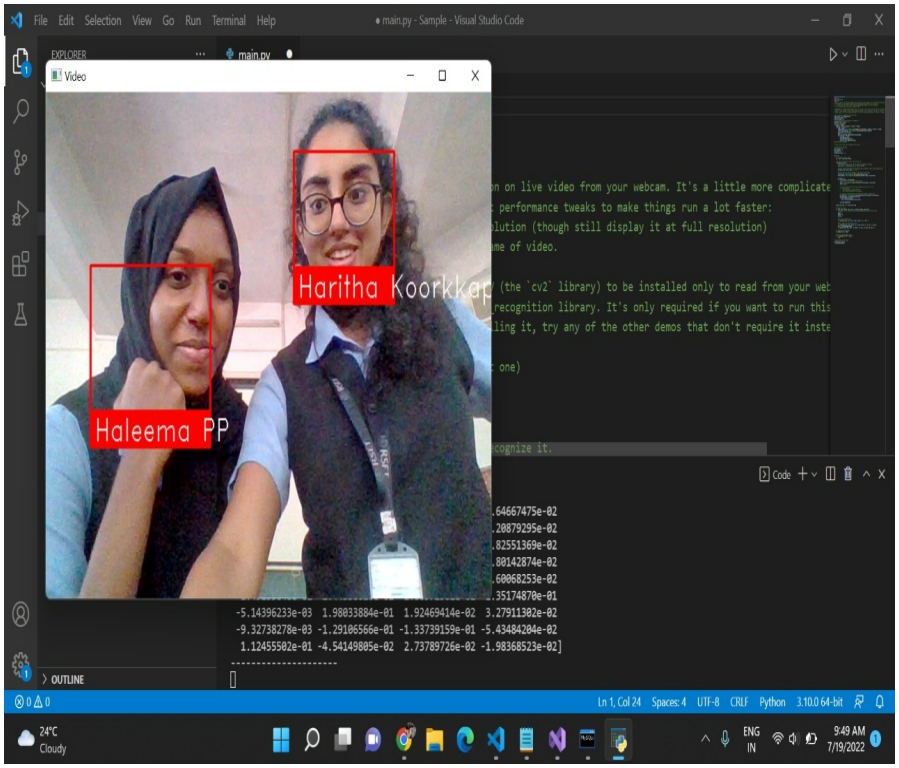


Figure 4.4.2. Face recognition

Chapter 5

CONCLUSION AND FUTURE SCOPE

5.1. Conclusion

Our project, Attendance Management System using Face Recognition was completed using Haar Cascade Classifier for face detection and Dlib's Face Recognition library for face recognition. Initially the student's face is registered with the system and an image is captured and is used for training the model. Then when the teacher clicks on the capture button, the faces are recognized by comparing the students faces to the input image. We were able to store the names of the students recognized into a text file.

5.2. Recommendation

Depending upon the time and compute power available, the following are some of the additional features that can be included to make our system function better:

- Marking the attendance in the database
- Increase in size of the dataset
- Using a better webcam/camera to capture the images

5.3. Scope of Future Work

- Accuracy can be improved by using more images of a student for training purposes
- The project can be improved to mark attendance of not only students but also employees in an organization or in any other fields of application
- Android or IOS applications can be developed in the front end so that the attendance marking can be performed through a phone.

REFERENCES

[1] Sanyukta Santosh Pawaskar, Ashwini Mandar Chavan, “Face Recognition based Class Management and Attendance System”, 2021.

[2] Sandhya Potadar, Riya Fale, Prajakta Kothawade, Arati Padale, “Attendance Management System using Face Recognition", August 2021

[3] Dev, Tushar Patnaik, “Student Attendance System using Face Recognition”,2020.

[4] Smitha, Pavithra S Hegde, Afshin, “Face Recognition based Attendance Management System”,May 2020

[5] Dr. V Suresh, Srinivasa Chakravarthi Dumpa, Chiranjeevi Deepak Vankayala, HaneeshaAduri, Jayasree Rapa, “Facial Recognition Attendance System Using Python and OpenCv”,2019

Appendix A IEEE Paper

Paper 1

FACE RECOGNITION BASED CLASS MANAGEMENT AND ATTENDANCE SYSTEM

The main objective of this paper is the implementation of an automated attendance system which uses the face recognition algorithms to record the attendance of the class and manage the class database.

Apart from the manual system, there are some systems that use technology like fingerprint system, iris scanning, etc. but what makes the system proposed here stands out is that it is a one-stop system to manage and record class attendance.

Steps of working of the system:

1. Add the record of each student
2. Training process-The images of the student are trained using LBPH and Haar Cascade and saved in the form of a YML file.
3. Tracking process - database of trained images is compared with the student's face to track the student's face.
4. The attendance is marked in the CSV.

The system is developed using Python: GUI, PDBC using sqlite3, and OpenCV. The following is the detailed methodology of the proposed system:

E. Creation of a database: We create the database for storing the details of the student (name, roll no,etc) of the students of a particular class. Here, the record of the student is to be added by clicking on the "Add Details" button. To one-stop the image of the student for the respective name click on the "Take Images" button. Using OpenCV, a real-time live video of the student will be recorded and split into sixty frames. These sixty frames are converted into grayscale and stored as student's face images. The "Status" label gives an update on the successful execution of database creation and image training process.

F. Image Training: To train the image, the database is already constructed which consists of an image and the image name. The Haar Cascade algorithm is used to calculate the histogram value of each image stored in the database. The database created by the image training process is saved as "trainer.yml" i.e in the form of a YML file such that it can be used in the future.

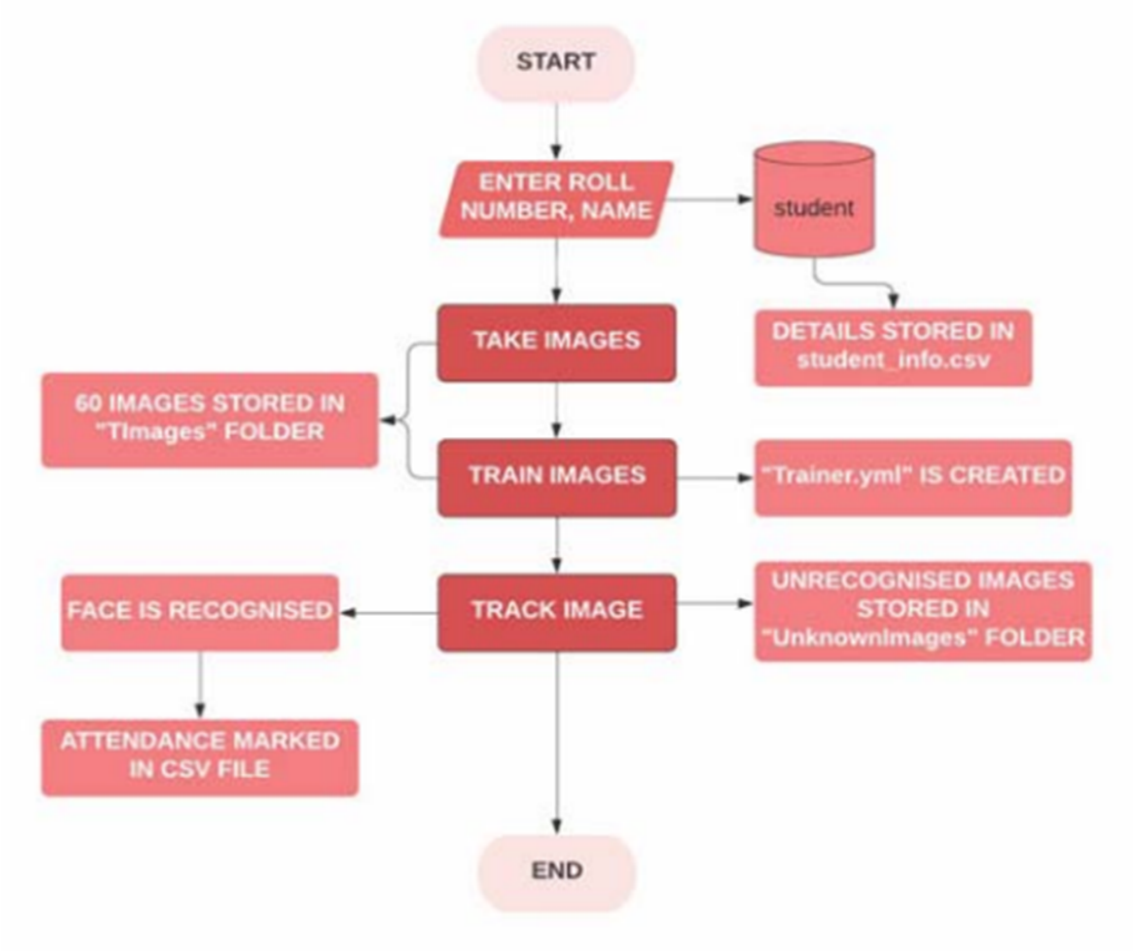
G. Image Tracking: The LBPH compares the histogram of the existing images in the database with the current real-time video of the student. The histogram of the current student video when matched with the existing database will display the student detail as "roll number: name" on the detected face. In case the student is not recognised it will label the student as "Unknown". The database of unknown images is saved in the respective folder.
H. Attendance Report: Once the image of the student is detected among the existing database, the attendance of the student is marked. The "Attendance Report" label gives an update of the attendance record. These four steps are to be used only during adding a new student in the database. For recording the attendance of the existing student the professor only needs to click on the "Track Images" using a laptop.

Advantages:

- Reliable and efficient system.(because of using LBPH and Haar Cascade algorithm)
- The system can give an accuracy of about 96.88%.
- This system improves the productivity of the class since there is no longer any source of disturbance caused by taking attendance manually

Disadvantages:

- To get better results and accuracy, the class should be well illuminated.



2.2. Paper 2

ATTENDANCE MANAGEMENT SYSTEM USING FACE RECOGNITION

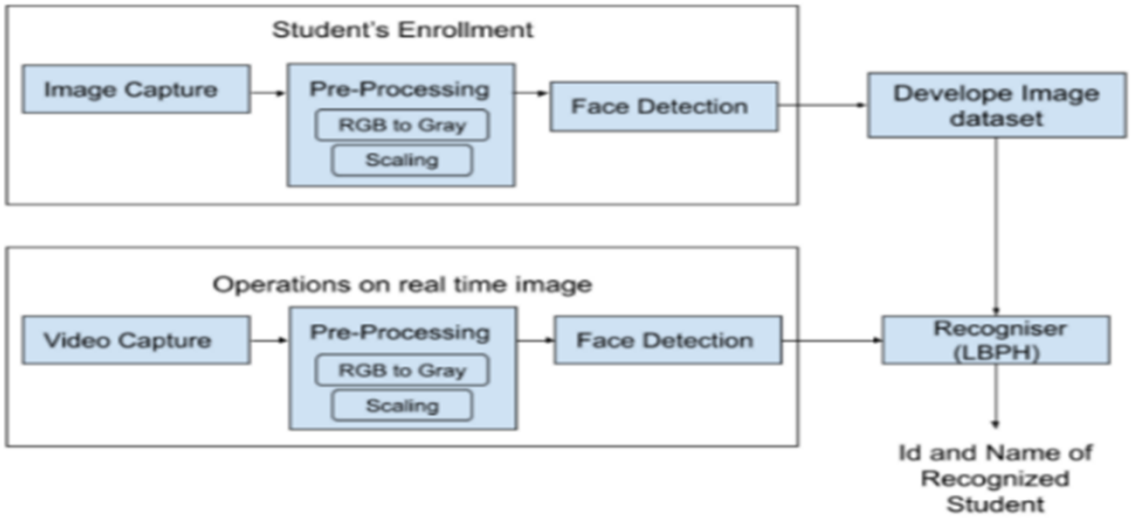
In the system proposed in this paper, initially all the students will be enrolled by storing their facial images with a unique ID. At the time of attendance, real time images will be captured and the faces in those images will be matched with the faces in the pre-trained dataset.

Face detection – Haar cascade algorithm helps in the face detection part. To simplify the As Haar cascade uses 24*24 windows there can be 160000+ features within a detector which needs to be calculated. AdaBoost is an algorithm based on machine learning. This algorithm finds the best features among all the possible features and eliminates the irrelevant features work of calculating the feature values, an Integral image algorithm is introduced. But this paper proposes to use Paul-Viola Jones Algorithm for face detection.

For the collection of the dataset, a minimum of 20 images are captured per individual student along with a unique ID.

Face recognition and training the stored dataset - Local Binary Patterns Histogram (LBPH) algorithm.

Histogram generated for stored images and real time images. Then the histograms are compared for face recognition and the one with lower difference is taken.



Appendix B
Code

Face Detection Code:

```
#Importing OpenCV
import cv2
#Importing HAAR CASCADE XML file
#face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')

#Capture Video from webcam hence (0) or else add your own media file
cap = cv2.VideoCapture(0)

#Creating a loop to capture each frame of the video in the name of Img
while True:
    _,img = cap.read()
```

```
    #Converting to grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #Allowing multiple face detection
    faces = face_cascade.detectMultiScale(gray, 1.1, 6)

    #Creating Rectangle around face
    for(x, y, w, h) in faces:
        cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 250), 2)
try:
        filename = 'img.jpg'

        # Show the image which was just taken.
        cv2.imshow('Detected Face Image', img)
    except Exception as err:
        # Errors will be thrown if the user does not have a webcam or if they do not
        # grant the page permission to access it.
        print(str(err))

    # save image
    cv2.imwrite(filename, img)



    #Waiting for escape key for image to close adding the break statement to end the face detection screen
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

#Real-time releasing the captured frames
cap.release()
```

Face Recognition Code:

```
#Training and recognition code

def recognition():
    video_capture = cv2.VideoCapture(0)
    known_face_encodings = []
    known_face_names = []
    path="C:\\Users\\97455\\Desktop\\AMS\\TrainingImage"
    folders=os.listdir(path)
    for folder in folders:
        images=os.listdir(path+"\\"+folder)
        for image in images:
            student_image = face_recognition.load_image_file(path+"\\"+folder+"\\"+image)
            student_face_encoding = face_recognition.face_encodings(student_image)[0]
```

```
            print(student_face_encoding)
            print("----------------------")
            known_face_encodings.append(student_face_encoding)
for foldername in folders:
                str=foldername
                personname=str
                known_face_names.append(personname)
    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True
    attnd=[]
    while True:
        ret, frame = video_capture.read()
        if process_this_frame:
            small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
            rgb_small_frame = small_frame[:, :, ::-1]
            face_locations = face_recognition.face_locations(rgb_small_frame)
            face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
   face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
            name = "Unknown"
            face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]
face_names.append(name)
            process_this_frame = not process_this_frame
            for (top, right, bottom, left), name in zip(face_locations, face_names):
                top *= 4
                right *= 4
                bottom *= 4
                left *= 4

                cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
                cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
                font = cv2.FONT_HERSHEY_DUPLEX
                cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
            cv2.imshow('Video', frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    video_capture.release()
    cv2.destroyAllWindows()
```

```
    t = time.localtime()
    current_time = time.strftime("%H_%M_%S", t)
    file=open('Attendance\\'+sub.get()+'\\'+current_time+'.txt','a')
    for stname in attnd:
        file.write(stname+'\n')
    file.close()
```

Tkinter Code (Main Program):

```
from base64 import b16decode
import tkinter
from tkinter import *
from tkinter import messagebox
import cv2
import mysql.connector
import os
import face_recognition
import numpy as np
import time
#Main page code in the end

#-----------------------------------------------------------------------------------------------------------------

#Login pages

#2.Login page for admin

def main_screen1():
    global my_window
    global username
    global password
    global screen1
    my_window.destroy()
    screen1=Tk()
    screen1.geometry("1280x1280")
    screen1.configure(bg="#d7dae2")


    screen1.title("ATTENDANCE MANAGEMENT SYSTEM")


    lblTitle=Label(text="Admin                                                                    Login
System",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=50)
```

```
    bordercolor=Frame(screen1,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Username", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=300)
    Label(mainframe,text="Password", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=400)
    username=StringVar()
    password=StringVar()

    entry_username=Entry(mainframe,textvariable=username,width=12,bd=2,font=("arial",30)).place(x=
600,y=300)
    entry_password=Entry(mainframe,textvariable=password,width=12,bd=2,font=("arial",30),show="*"
).place(x=600,y=400)

    b1=Button(mainframe,text="Login",height="2",width=23,bg="#ed3833",fg="white",bd=0,
command=login1).place(x=300,y=500)
    b2=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
screen1.destroy).place(x=700,y=500)
    screen1.mainloop()

#3.Login page for teacher

def main_screen2():
    global my_window
    global username
    global password
    global screen2
    my_window.destroy()
    screen2=Tk()
    screen2.geometry("1280x1280")
    screen2.configure(bg="#d7dae2")

    screen2.title("ATTENDANCE MANAGEMENT SYSTEM")

    lblTitle=Label(text="Teacher                                                    Login
System",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=50)

    bordercolor=Frame(screen2,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Username", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=300)
    Label(mainframe,text="Password", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=400)
```

```
    username=StringVar()
    password=StringVar()

    entry_username=Entry(mainframe,textvariable=username,width=12,bd=2,font=("arial",30)).place(x=
600,y=300)
    entry_password=Entry(mainframe,textvariable=password,width=12,bd=2,font=("arial",30),
show="*").place(x=600,y=400)


    b1=Button(mainframe,text="Login",height="2",width=23,bg="#ed3833",fg="white",bd=0,
command=login2).place(x=300,y=500)
    b2=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
screen2.destroy).place(x=700,y=500)
    screen2.mainloop()

#4.Login page for student

def main_screen3():
    global my_window
    global username
    global password
    global screen3

    my_window.destroy()
    screen3=Tk()
    screen3.geometry("1280x1280")
    screen3.configure(bg="#d7dae2")

    screen3.title("ATTENDANCE MANAGEMENT SYSTEM")


    lblTitle=Label(text="Student                                                                    Login
System",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=50)

    bordercolor=Frame(screen3,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Username", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=300)
    Label(mainframe,text="Password", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=400)
    username=StringVar()
    password=StringVar()
```

```
    entry_username=Entry(mainframe,textvariable=username,width=12,bd=2,font=("arial",30)).place(x=
600,y=300)
    entry_password=Entry(mainframe,textvariable=password,width=12,bd=2,font=("arial",30),show="*"
).place(x=600,y=400)

    b1=Button(mainframe,text="Login",height="2",width=23,bg="#ed3833",fg="white",bd=0,
command=login3).place(x=300,y=500)
    b2=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
screen3.destroy).place(x=700,y=500)
    screen3.mainloop()

#-------------------------------------------------------------------------------------------------------------------

#Login Credentials

#Login credentials for admin

def login1():

    mysqldb                    =                    mysql.connector.connect(host='127.0.0.1',user='root',
password='root',database='ATTENDANCE')
    mycursor = mysqldb.cursor()
    user=username.get()
    code=password.get()

    sql=f"select * from ADMIN where USERNAME ='{user}' and PASSWORD ='{code}'"
    mycursor.execute(sql)
    r=mycursor.fetchone()

    if r:
        admin_window()

    else:
        messagebox.showinfo("","incorrect username and password")

#Login credentials for teacher

def login2():
    mysqldb                    =                    mysql.connector.connect(host='127.0.0.1',
user='root',password='root',database='ATTENDANCE')
    mycursor = mysqldb.cursor()
    user=username.get()
    code=password.get()

    sql=f"select * from TEACHER where NAME ='{user}' and PASSWORD ='{code}'"
```

```
    mycursor.execute(sql)
    r=mycursor.fetchone()
    if r:
        teacher_window()

    else:
        messagebox.showinfo("","incorrect username and password")

#Login credentials for student

def login3():
    mysqldb = mysql.connector.connect(host='127.0.0.1',
user='root',password='root',database='ATTENDANCE')
    mycursor = mysqldb.cursor()
    user=username.get()
    code=password.get()

    sql=f"select * from STUDENT where SID ='{user}' and PASSWORD ='{code}'"
    mycursor.execute(sql)
    r=mycursor.fetchone()

    if r:
        student_window()
    else:
        messagebox.showinfo("","incorrect username and password")

#------------------------------------------------------------------------------------------------------------

#PAGE-1 AFTER LOGIN FOR ADMIN, STUDENT AND TEACHER

#5.Admin page 1

def admin_window():
    global screen
    global b1,b2
    global admin_window_page
    screen1.destroy()
    admin_window_page=Tk()
    admin_window_page.title("ADMIN LOGIN PAGE")
    admin_window_page.geometry('650x500')
    admin_window_page.configure(bg="#d7dae2")

    admin_window_page.title("ATTENDANCE MANAGEMENT SYSTEM")
    Label(text="Administration Page",font=("arial",30,"bold"),fg="white",bg="black").pack()
```

```
    Button1=Button(admin_window_page,text="Register new Student",bd='5',height=5,width=30,
font=("Ariel",10),command =page2_admin).pack(pady= 15, padx= 20,side='left')
    Button2=Button(admin_window_page,text="View Attendance",bd='5',height=5,width=30,
font=("Ariel",10)).pack(pady= 15, padx= 20,side='right')
    admin_window_page.configure(bg="black")
    admin_window_page.mainloop()

#6.Teacher page 1

def teacher_window():
    global screen2
    global b1,b2
    global teacher_window_page
    screen2.destroy()
    teacher_window_page=Tk()
    teacher_window_page.title("ATTENDANCE MANAGEMENT SYSTEM")
    teacher_window_page.geometry('650x500')
    teacher_window_page.configure(bg="#d7dae2")

    teacher_window_page.title("Faculty Page")
    Label(text="Faculty Page",font=("arial",30,"bold"),fg="white",bg="black").pack()

    Button1=Button(teacher_window_page,text="Take Attendance",bd='5',height=5,width=30,
font=("Ariel",10),command=page2_teacher).pack(pady= 15, padx= 20,side='left')
    Button2=Button(teacher_window_page,text="View Attendance",bd='5',height=5,width=30,
font=("Ariel",10)).pack(pady= 15, padx= 20,side='right')
    teacher_window_page.configure(bg="black")
    teacher_window_page.mainloop()

#7.Student page 1

def student_window():
    global screen3
    global b1,b2
    global student_window_page
    screen3.destroy()
    student_window_page=Tk()
    student_window_page.title("ATTENDANCE MANAGEMENT SYSTEM")
    student_window_page.geometry('650x500')
    student_window_page.configure(bg="#d7dae2")

    student_window_page.title("Student Page")
    lblTitle=Label(text="Student Page",font=("arial",30,'bold'),fg="white",bg="black").pack(pady=50)
```

```
    Button1=Button(student_window_page,text="View Attendance",bd='5',height=5,width=100,
font=("Ariel",10), command=page2_student).pack(pady= 15, padx= 200,side='left')
    #Button2=Button(student_window_page,text="View Attendance",bd='5',height=5,width=30,
font=("Ariel",10)).pack(pady= 15, padx= 20,side='right')
    student_window_page.configure(bg="black")
    student_window_page.mainloop()

#-------------------------------------------------------------------------------------------------------------

#Training and recognition code

def recognition():
    #file=open('Attendence.txt','rb')
    video_capture = cv2.VideoCapture(0)
    known_face_encodings = []
    known_face_names = []
    path="C:\\Users\\97455\\Desktop\\AMS\\TrainingImage"
    folders=os.listdir(path)
    for folder in folders:
        images=os.listdir(path+"\\"+folder)
        for image in images:
            print(image)
            student_image = face_recognition.load_image_file(path+"\\"+folder+"\\"+image)
            student_face_encoding = face_recognition.face_encodings(student_image)[0]
            #print(student_face_encoding)
            print("----------------------")
            known_face_encodings.append(student_face_encoding)
            for foldername in folders:
                str=foldername
                personname=str
                known_face_names.append(personname)

    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True
    attnd=[]
    while True:
        ret, frame = video_capture.read()
        if process_this_frame:
            small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

            rgb_small_frame = small_frame[:, :, ::-1]

            face_locations = face_recognition.face_locations(rgb_small_frame)
```

```
        face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)

        face_names = []
        for face_encoding in face_encodings:
            matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
            name = "Unknown"

            face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]
                if name not in attnd:
                    attnd.append(name)
            face_names.append(name)

    process_this_frame = not process_this_frame

    for (top, right, bottom, left), name in zip(face_locations, face_names):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4

        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)

    cv2.imshow('Video', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video_capture.release()
cv2.destroyAllWindows()
t = time.localtime()
current_time = time.strftime("%H_%M_%S", t)
file=open('Attendance\\'+sub.get()+'\\'+current_time+'.txt','a')
for stname in attnd:
    file.write(stname+'\n')
file.close()
```

#----------

#PAGE-2 AFTER EACH RESPECTIVE PAGES

```
#9.Teacher page 2
def page2_teacher():
    global sub
    global teacher2
    global teacher_window_page
    teacher_window_page.destroy()
    teacher2=Tk()
    teacher2.geometry("1280x1280")
    teacher2.configure(bg="#d7dae2")


    teacher2.title("ATTENDANCE MANAGEMENT SYSTEM")


    lblTitle=Label(text="ATTENDANCE",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=
50)

    bordercolor=Frame(teacher2,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Enter sub:", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=250)

    sub=StringVar()


    entry_name=Entry(mainframe,textvariable=sub,width=12,bd=2,font=("arial",30)).place(x=600,y=250
)



    b1=Button(mainframe,text="Fill
Attendance",height="2",width=23,bg="#ed3833",fg="white",bd=0,command=recognition).place(x=350,
y=450)
    b2=Button(mainframe,text="View
Attendance",height="2",width=23,bg="blue",fg="white",bd=0).place(x=550,y=450)
    b3=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
teacher2.destroy).place(x=750,y=450)
    teacher2.mainloop()

#10.Student page 2
def page2_student():
    global sub
    global student2
    global student_window_page
```

```
    student_window_page.destroy()
    student2=Tk()
    student2.geometry("1280x1280")
    student2.configure(bg="#d7dae2")


    student2.title("ATTENDANCE MANAGEMENT SYSTEM")


    lblTitle=Label(text="Welcome
Page",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=50)

    bordercolor=Frame(student2,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Enter sub:", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=250)

    sub=StringVar()

    entry_name=Entry(mainframe,textvariable=sub,width=12,bd=2,font=("arial",30)).place(x=600,y=250
)

    b1=Button(mainframe,text="View
Attendance",height="2",width=23,bg="blue",fg="white",bd=0).place(x=400,y=450)
    b2=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
student2.destroy).place(x=650,y=450)
    student2.mainloop()

#-----------------------------------------------------------------------------------------

#registration of student details to the database
def register_student():
    global name1
    try:
        mysqldb                    =                    mysql.connector.connect(host='127.0.0.1',
user='root',password='root',database='ATTENDANCE')
        mycursor = mysqldb.cursor()

        name1 =name.get()
        roll1 = roll_no.get()
        password1 = password.get()

        insert_stmt = ("INSERT INTO STUDENT( SID, NAME, PASSWORD)""VALUES(%s,%s,%s)")
```

```
        data= (roll1,name1,password1)

        mycursor.execute(insert_stmt,data)
        mysqldb.commit()

    except:
            pass

#detectioncode
def detection():
    face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

#Capture Video from web cam hence (0) or else add your own media file
    cap = cv2.VideoCapture(0)
    directory=name1
    path=os.path.join("C:\\Users\\97455\\Desktop\\AMS\\TrainingImage",directory)
    os.mkdir(path)

#Creating a loop to capture each frame of the video in the name of Img
    while True:
        _,img = cap.read()

    #Converting to grey scale
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #Allowing multiple face detection
    #detectMultiScale(img,ScaleFactor,MinNeighbours)
        faces = face_cascade.detectMultiScale(gray, 1.1, 6)

    #Creating Rectangle around face
    #cv2.rectangle(image, start_point, end_point, color, thickness)
        for(x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x+w, y+h), (0, 0, 250), 2)
            cv2.imwrite(f"{path}\ " + name1 + ".jpg", gray[y : y+h, x : x+w])

        try:

       # Show the image which was just taken.
            cv2.imshow('Detected Face Image', img)
        except Exception as err:
       # Errors will be thrown if the user does not have a webcam or if they do not
       # grant the page permission to access it.
            print(str(err))
```

```
    #Waiting for escape key for image to close adding the break statement to end the face detection screen
        k = cv2.waitKey(30) & 0xff
        if k == 27:
            break

#Real-time releasing the captured frames
     cap.release()


#TakeImageUI
def page2_admin():
    global admin_window_page
    global screen1
    global name
    global roll_no
    global password
    admin_window_page.destroy()
    admin2=Tk()
    admin2.geometry("1280x1280")
    admin2.configure(bg="#d7dae2")

    admin2.title("ATTENDANCE MANAGEMENT SYSTEM")

    lblTitle=Label(text="STUDENT
DETAILS",font=("arial",50,'bold'),fg="black",bg="#d7dae2").pack(pady=50)

    bordercolor=Frame(admin2,bg="black",width=800,height=400)
    bordercolor.pack()

    mainframe=Frame(bordercolor,bg="#d7dae2",width=800,height=400).pack(padx=20,pady=20)

    Label(mainframe,text="Name", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=250)
    Label(mainframe,text="Student ID", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=350)
    Label(mainframe,text="Password", font=("arial",30,"bold"),bg="#d7dae2").place(x=300,y=450)

    name=StringVar()
    roll_no=StringVar()
    password=StringVar()

    entry_name=Entry(mainframe,textvariable=name,width=12,bd=2,font=("arial",30)).place(x=600,y=2
50)
    entry_id=Entry(mainframe,textvariable=roll_no,width=12,bd=2,font=("arial",30)).place(x=600,y=350
)
    entry_password=Entry(mainframe,textvariable=password,width=12,bd=2,font=("arial",30)).place(x=
600,y=450)
```

```
    takeImg=Button(mainframe,text="Take
Image",height="2",width=23,bg="#ed3833",fg="white",bd=0,command=detection).place(x=300,y=550)
    b2=Button(mainframe,text="Enter",height="2",width=23,bg="blue",fg="white",bd=0,
command=register_student).place(x=500,y=550)
    b5=Button(mainframe,text="EXIT",height="2",width=23,bg="#00bd56",fg="white",bd=0,command=
admin2.destroy).place(x=700,y=750)
    admin2.mainloop()

#-------------------------------------------------------------------------------------------------------------

#1.Main starting page

my_window=Tk()
my_window.title("ATTENDANCE MANAGEMENT SYSTEM")

my_window.geometry('650x500')

canvas=Canvas(my_window,height=1, width=1, bg="#263d42" )
canvas.pack()

login_as=Label(my_window,text="Login    as:",    font=("Ariel",20),fg="white",bg="black").pack(pady=
15, padx= 20)

icon_admin=PhotoImage(file='C:\\Users\\97455\\Desktop\\AMS\\Admin.png')
icon_teacher=PhotoImage(file='C:\\Users\\97455\\Desktop\\AMS\\Teacher.png')
icon_student=PhotoImage(file='C:\\Users\\97455\\Desktop\\AMS\\Student.png')

admin=Button(my_window,image=icon_admin,bd='5',height=200,width=150,          font=("Ariel",10),
command=main_screen1).pack(pady= 15, padx= 20, side='left')
admin_lbl=Label(my_window,text="Admin",fg="white",bg="black").place(x=80,y=400)

teacher=Button(my_window                              ,image=icon_admin,bd='5',height=200,width=180,
font=("Ariel",10),command=main_screen2).pack(pady= 15, padx= 20,side='left')
teacher_lbl=Label(my_window,text="Teacher",fg="white",bg="black").place(x=300,y=400)

student=Button(my_window,image=icon_student,bd='5',height=200,width=150,
font=("Ariel",10),command=main_screen3).pack(pady= 15, padx= 20,side='left')
student_lbl=Label(my_window,text="Student",fg="white",bg="black").place(x=520,y=400)

my_window.configure(bg="black")

my_window.mainloop()
```