A MINI PROJECT REPORT ON

SMART ATTENDANCE SYSTEM

SUBMITTED IN PARTIAL FULFILLMENT FOR THE
DEGREE OF BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND TECHNOLOGY

BY

SHREYA BAMANE

MITALI BELGE

SHWETA NIKAM

UNDER THE GUIDANCE OF

PROF. SONALI BODEKAR KALE

USHA MITTAL INSTITUTE OF TECHNOLOGY

S. N. D. T. WOMEN'S UNIVERSITY

MUMBAI – 400049

2021 – 2022.

CERTIFICATE

This is to certify that **Shreya Bamane, Mitali Belge** and **Shweta Nikam** have successfully completed seminar work on **Smart Attendance System using Face Recognition** in the partial fulfillment for the bachelor's degree in **Computer Science and Technology** during the year 2021-2022 as prescribed by SNDT Women's University.

GUIDE	HEAD OF THE DEPARTMENT
Prof. Sonali Bodekar Kale	**Prof. Kumud Wasnik**

PRINCIPAL

Prof. Shikha Nema

EXAMINAR 1	EXAMINAR 2

ACKNOWLEDGEMENT

We pay our deep sense of gratitude to our guide Prof. Sonali Bodekar who encouraged and guided us throughout the project. Our sincere thanks to our Principal and The Head of the Computer Science and Technology Department Prof. Kumud Wasnik for setting us a platform to explore. Last but not the least, sincere thanks to all our Teaching staff ,non teaching staff, family and friends who shared their support morally. With due regards, thank you to all the students of UMIT for helping us with our survey.

Date:_____

Shreya Bamane

Mitali Belge

Shweta Nikam

Contents

List of Figures

List of Tables

1 Abstract

An attendance system is used to track and monitor whether a student attends a class. In most educational institutions the student's attendance is taken manually by using an attendance sheet, given by the faculty member in a class but this method is monotonous and time consuming. To solve this problem, the process has been computerized. There are different types of computerized attendance systems like Biometric-based that use fingerprints, iris, speech and face for recognition, Radiofrequency card-based and QR code based systems. Out of them all, a face recognition based attendance system is more secure and time-saving. Therefore, we have built an attendance system using face recognition. In this we have used the Haar Cascade algorithm for face detection and Local Binary Patterns Histograms (LBPH) algorithm for facial recognition. The system built automatically marks the attendance in the excel sheet on recognising the face of a student. Our goal is to use the best automated method for the process of taking attendance.

2 Problem Statement

Maintenance of attendance is incredibly necessary at institutes for grading and keeping a track of students. Every institution has its method of taking student attendance, some institutes take attendance manually using paper or file-based approaches and a few have adopted ways of automatic attendance using biometric techniques. But the most common method for marking attendance is calling out names of students. It's a long and less efficient method for marking attendance because the information written in the paper typically gets lost or is less accurate as students often mark each other's proxy. Therefore, to unravel these issues and avoid errors, we recommend computerizing this method by providing a system that records and manages student's attending mechanically with no need for lecturers' interference. Every biometric system consists of the enrollment process during which distinctive features of a person are stored and then there are processes of identification and verification. These 2 processes compare the biometric feature of someone with previously

stored biometric details captured at the time of enrollment. Biometric templates are of the many varieties like Fingerprints, Eye Iris, Face, Signature, and Voice. Our system uses the face recognition approach for the automated Attendance System. This system gives many more solutions with accurate results in a user interactive manner rather than existing attendance systems.

3 Literature survey

In An Attendances System Unit Using the Radio Frequency Identification Concept [1] proposed an attendance System using RFID cards.Radio-frequency identification (RFID) is an automatic identification method, relying on storing and remotely retrieving data using devices called RFID tags. An RFID tag is an object that can be stuck on or incorporated into a product, animal, or person for the purpose of identification using radio waves. In this paper, the research concept is based on a microcontroller or arduino approach that digitalizes analogue signals obtained from sensors used to monitor the receipt of signals from radio frequency chips implanted in tags and cards. It monitors persons or objects and keeps record or a register of their attendance automatically through the aid of a timing mechanism, and stores the register information on an SD card through the SD Card Module incorporated into the project. The register information that is stored in the SD card can then be easily gotten by removing the SD card from the module and copying out the data for further manipulation through the various software's available such as SPSS, Microsoft Access, and Excel.The proposed RFID tag uses energy from the tag reader. The problem with this approach is that an unknown person can make use of a valid ID card and enter the institute. And another disadvantage of this system is that some students may make use of other students' ID to ensure their presence when the particular student is absent which may lead to misuse of the system.

In QR Code Based Smart Attendance System [2] this a QR code generator android app is developed using the details of students such as roll number, student ID. It takes the attendance with respect to the specific subject and generates the student attendance sheet as per attendance details. The QR codes will not be static, they will be dynamic. It will keep on changing every 30 seconds, so that no one can click a photo and send it to his friends or colleague to get a proxy attendance.

After scanning the QR codes successfully the attendance will get punched automatically. The attendance will be marked as 0 and 1, 0 for absent and 1 for present students. The problem is that not everyone can afford a mobile phone to generate the Qr code.

The Fingerprint Based Attendance System Using Microcontroller and LabView [3] proposed a solution of using fingerprint to mark the attendance. This system is using 2 microcontrollers to deal with the fingerprint recognition process. Firstly, the fingerprint pattern will be obtained through a fingerprint sensor, then the information will be transmitted to microcontroller 1. Next microcontroller 1 will pass the information to microcontroller 2 to do the checking with the database that resides in it. After finding a student's match, the details are sent to the PC through serial communication to be displayed. But again, this system is attached to a PC which makes it not portable. (Meaning that, for the parents who are interested in knowing their childs attendance cannot easily or conveniently access the information.) The drawback of this system is that there are health issues involved due to touching a single scanning sensor device by countless numbers of individuals. This has serious implications in pandemic situations such as COVID-19 etc.

In Speech Biometric Based Attendance System [4] they have presented the development and implementation of a speech biometric based attendance system. The users access the system by making a call from a few pre-decided mobile phones. An interactive voice response (IVR) system guides a new user in the enrollment and an enrolled user in the verification processes. The system uses text independent speaker verification with Mel-frequency cepstral coefficients (MFCC) features and i-vector based speaker modeling for authenticating the user. Linear discriminant analysis and within class covariance normalization are used for normalizing the effects due to session/environment variations. A simple cosine distance scoring along with score normalization is used as the classifier and a fixed threshold is used for making the decision. The disadvantage using speech recognition for attendance is that students can pre record their voice and ask their friends to mark their false attendance.

The focus of Design and Implementation of a Student Attendance System Using Iris Biometric Recognition [5] paper is to create an automated class attendance system using iris biometric. The attendance of students in a class is tracked and automatically updated each time their iris gets

scanned before entering the class. The designed system allows students to be registered or enrolled and verified when attendance is taken. An Iris Scanner was used to get the image of the iris. The graphical user interface (webpage) was made using HTML, xJavaScript, and CSS. The database was created using MySQL language. Students are identified when a match is found in the database for the iris image of each student just acquired. The designed system returns an integer value of (1) if there is a match and an integer value of (0) if there is no match. These results are stored in the database along with the time, date, month, and year the attendance was taken. They have also used false match rate(FMR) which is the percentage of times the student attendance system produces a false accepts, that occurs when an individual is incorrectly matched to another user. The main disadvantage of this system is that it needs powerful scanners to scan the iris which makes the system expensive and all institutes cannot afford it. Eyelashes, lenses, and reflections can cause obstacles.

In The Implementation of Eigenface Algorithm for Face Recognition in Attendance System [6] authors proposed a face recognition based attendance system based on Eigenface recognition. Firstly, features are extracted using a hardware device that takes a face image and sends it to the server for verification and identification process. Next the image is saved in low capacity size for verification and identification. And lastly in the identification process several training images are compared with input images during the identification process. In the Eigenface algorithm, images are converted into eigenfaces and recognition is performed by comparing eigenfaces obtained from the input image and eigenfaces in the Database. The problem with this approach is that this method is very sensitive to face background, head orientations and it doesn't recognize the face of a person if the person is wearing glasses or has a grown beard, etc. But in the approach proposed in this paper, our system is not sensitive to face background, head orientations and it recognizes a person's face even if he grows a beard or wears glasses, etc.

4 Introduction To Face Recognition

A person's face has distinctive physical shape and characteristics that are used to identify or verify an individual. Facial recognition records this biometrics of the face. Different face recognition methods measure the biometric of the face. Facial recognition has become a very important topic in recent years. Facial recognition is effectively applied in various applications like security systems, authentication, entrance control, surveillance system, unlocking of smartphones and social networking systems, etc. Most of the practices do not use facial recognition as the main form of conceding entry. However, with advancement in technology and algorithms, facial recognition systems have the potential to replace the standard passwords and fingerprint scanners. This project was carried out to show how a Local Binary Pattern Histogram (LBPH) face recognizer could be used for taking attendance of students. LBPH facial recognizer is a pre-trained facial recognition classifier. If enough data sets are available on the face that is needed to be identified, LBPH can perform facial recognition with high accuracy. Face Recognition Student Attendance System is a desktop application that identifies and verifies student's identities with the help of a digital image. Once the recognized face matches with the stored image, the attendance is completed and marked in the database for the student. This system will provide an alternative and easier way of taking attendance.[7] The facial recognition system has three main phases, which are described below: Face Detection Face detection is the ability to identify the person's faces within the digital images. This system identifies the human face present in an image or video. We need to define a general structure of a face to determine whether a certain picture or 4 videos contains a face (or several). Human faces have the same features such as eyes, nose, forehead, mouth, and chin. Therefore, the objective of face detection is to find the location and size of the face in an image. The located face is then used by the facial recognition algorithm. Feature Extraction In this , we are extracting the features from the detected face. In LBPH, the first local binary pattern images are computed, and a histogram is created for facial recognition. This generates a template. A template is a set of data that represents the unique and distinctive features of the detected face. Face Recognition Face Recognition is being able to uniquely identify and verify a person's face by comparing and analyzing a biometrics person's face. A face recognition system is an application that is used for identifying or verifying a person from a digital image.

5 Existing Systems

Types of Attendance System (Existing Systems)	Drawbacks
RFID based system	Unauthorized can enter by scanning some else ID card
Qr code based system	Phones are not affordable by everyone
Fingerprint based system	Diseases spread due to touching a single scanning sensor eg. Covid-19
Speech based system	Students can pre record their voice and mark false attendance
Iris based system	Eyelashes, lenses, and reflections can cause obstacles
Face recognition system using Eigenface	Doesn't recognize the face of a person if the person is wearing glasses or has a grown beard

Table 1: Drawbacks of Existing Systems

The existing, the attendance management system in school/colleges was done by manual reporting where the students attendance were recorded by placing a mark or signature beside their name in a name list to indicate their presence in a particular class. While the staff in the institution will report their attendance through the punch card machine which also has to be done manually. Later on, some of those attendance systems are computerized into using smart attendance systems. Attendance using RFID methods, Qr code and biometric methods such as Fingerprint system, Speech based system and Iris based system are some automated existing systems. These methods generally require more time for calculation.

RFID(Radio Frequency Identification) is a method where there are electromagnetic fields which are used to automatically detect and track the tags attached to persons. RFID can invade the privacy and security of human beings in the industrial environment. RFID strategies in turn affect the software that allows each person to be analyzed by the primary database. This environment can be easily studied by hackers. If RFID reader and receiver are not properly resonated then less read rate occurs. The biggest disadvantage of using RFID as an attendance system is that unauthorized people can enter by scanning somebody's ID card and cause chaos in the institute or harm others.

A Quick Response code (QR- code) is a two-dimensional bar code. Every student contains a unique Qr code. Students just have to scan it in front of a webcam and the system notes down their attendance as per dates. Each Qr code contains a unique id for students. System then stores all the students' attendance records. It also generates an overall report in excel sheet for admin. The disadvantage of this system is that many people have poor financial conditions hence phones are not affordable by everyone to generate the QR code for attendance.

Biometric fingerprint identification system makes use of fingerprints as a unique identity. Amongst all the systems used for recognition this is the most accurate and an efficient one. But recognition of an individual fingerprint from a set of enrolled fingerprints is a difficult process. The fingerprint system may not always reveal any detail about the original fingerprint. This may have been proved to be false as many algorithms reveal that a fingerprint can be reconstructed with minute templates. Also another drawback of this system is that there are health issues involved due to touching a single scanning sensor device by countless numbers of individuals, that is diseases spread due to touching a single scanning sensor. This has serious implications in pandemic situations such as COVID-19 etc.

Speech Recognition is the methodology of consequently perceiving a certain word talked by a specific speaker taking into account singular data included in speech waves. This system makes it conceivable to utilize the speaker's voice to confirm his/her personality and give controlled access to administrations of voice based biometrics. Speech processing front end for extricating the feature set is a critical stage in any voice recognition system. Drawbacks of this system are many, one of them being students can pre record their voice and mark false attendance.

Iris Recognition is another type of implementation where the iris of people are scanned, stored and then retrieved for the matching and attendance is done without fail in the server. But there is a lot of trouble in accumulating the iris of the students and therefore a fast application of face recognition with minimum illumination can be used. The main disadvantage of this system is that it needs powerful scanners to scan the iris which makes the system expensive and all institutes cannot afford it. Eyelashes, lenses, and reflections can cause obstacles.

Facial Recognition using Eigenface Algorithm, features are extracted using a hardware

device that takes a face image and sends it to the server for verification and identification process. Next the image is saved in low capacity size for verification and identification. And lastly in the identification process several training images are compared with input images during the identification process. In the Eigenface algorithm, images are converted into eigenfaces and recognition is performed by comparing eigenfaces obtained from the input image and eigenfaces in the Database.

Face recognition attendance system with IOT gives high accuracy, it access and control IOT devices, which is always low cost and low power. Main aim is to reduce the documentation cost and efforts of human beings which are generated in the digital classroom and offices. So the proposed face recognition system combines three part face detection, feature extraction and face recognition. The extended local binary pattern then extracts the local features of the face.

6 Proposed System

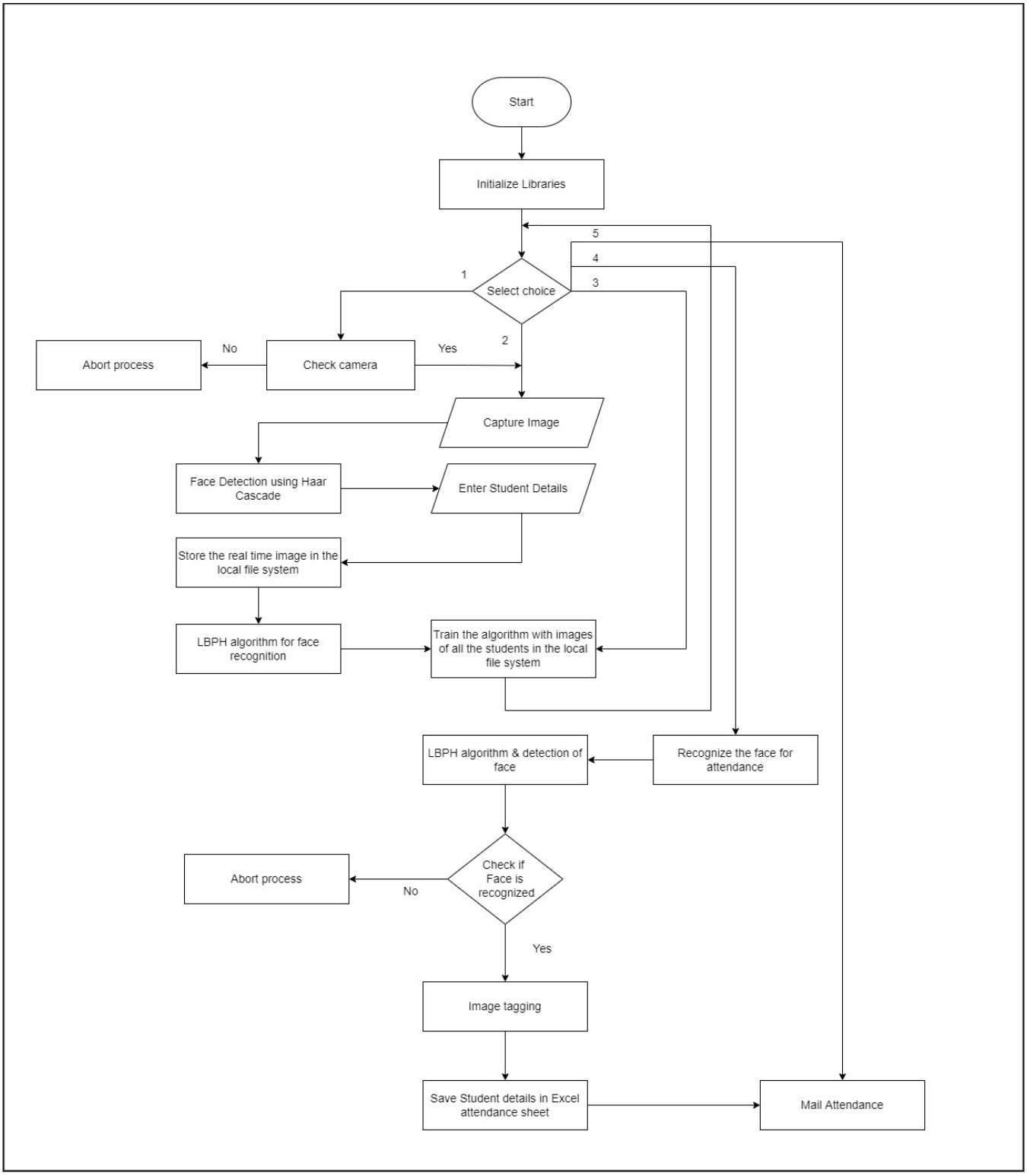

Figure 1: Mechanism of Proposed System

The proposed system is based on face recognition because it gives better security, accuracy and is robust as compared to the other systems. This system uses the Haar Cascade Classifier for face detection with Local Binary Pattern Histogram (LBPH) for face recognition.

At first, the camera will capture the images and take the student details i.e name, roll no, department. This Roll no, Name, and Department is stored in the Student Details folder and file name is saved as Student Details.csv. It takes 100 images as samples and stores them in the TrainingImage folder. Haar Cascade Classifier is used here for face detection. After capturing the image it will preprocess the image and convert the image into gray scale Image. In order to train the image samples we have to press 3 which is the third step, Train Image. Now it takes a few seconds to train the machine for the images and creates a Trainner.yml file and stores them in the TrainingImageLabel folder. When a student comes across the camera module, then his/her image/photo will be captured and recognized with validation. LBPH is used here for face recognition along with Haar Cascade Classifier for face detection. When the student is recognized by the camera then his/her roll no and name is shown on image. Press Q (or q) to quit this window. The details of the students will be saved in the Excel Sheet in the Attendance folder as a csv file with name, roll no, department, date and time. We can also mail the attendance sheet to the respective faculty by just entering their mail id in the system which is an added feature in this project. This proposed work is based on the above block diagram in which the attendance of the particular student is marked as present when his/her face is matched. It will be easier to use and comfortable with a decrease in cost as buying accessories will increase the overall cost.

This system provides functionalities such as taking images of students along with their details such as roll no., name and also stores time and date in excel sheet.

There are two phases in Smart Attendance System using Face Recognition:-

Face Detection: Face Detection is a method of detecting faces in the images. It is the first and essential step needed for face recognition. It mainly comes under object detection like for example a car in an image or any face in an image and can be used in many areas such as security, bio-metrics, law enforcement, entertainment, personal safety, etc. For Face Detection Haar cascade algorithm is used

6.1 Haar Cascade Classifier

The Scientist Paul Viola and Michael Jones proposed the effective object detection method Haar cascade classifier. This is an applied machine learning and in-depth learning-based approach. From the many cascade algorithms, a cascade method analyzes from the positive and negative images. Then it will be used in other images to detect objects. Here, without faces to explain the classifier, face detection algorithms will be applied. Here there is a need for a lot of positive and negative face frames in the video. For this, we use Haar features shown in the below image. Each feature is claimed to be one value which is obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle.

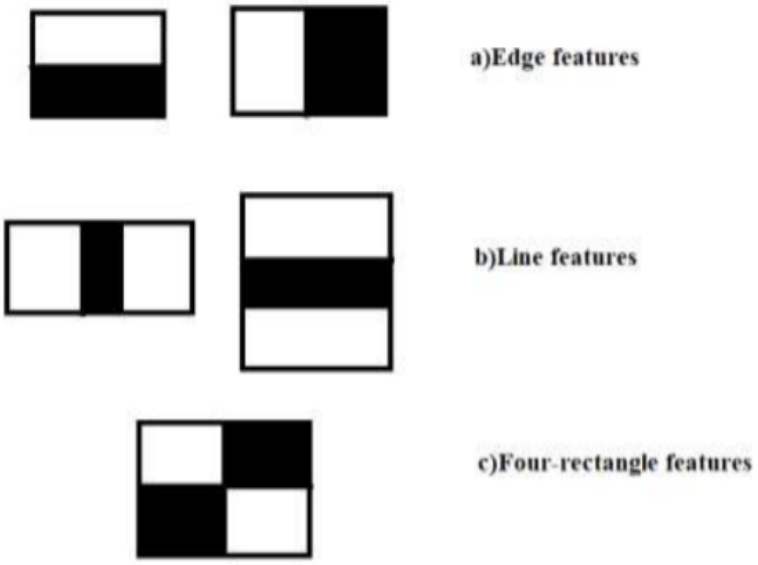

Figure 2: Haar Feature Extraction

There is a different type of haar cascading feature, which analyzes the function in the existing image. From this, the crop image into the 24X24 window in each operation subtracts the sum of white region pixels with the number of black region pixels in it. Therefore, there will be an integer value as the output. It also determines the validation of the features in it.

Face Recognition: Face Recognition is a method of identifying or verifying a person from images and videos that are captured through a camera. Its Key role is to identify people in photos, videos, or in real-time.

For Face Recognition LBPH algorithm is used

6.2 Local Binary Pattern Histogram

Local Binary Pattern (LBP) can be a clear yet productive surface that puts marks on parts of the picture by impeding the area, all things considered, and seeing the outcome as a paired width. It was resolved that when LBP is joined with histograms of characterized angles (HOG) directed, it improves the securing execution of more informational indexes. Utilizing LBP joined with histograms we can represent to confront pictures with a vector of direct information. The LBPH algorithm works in 5 steps as given below:

1. **Parameters: Lists are easy to create:**
    - Radius: the radius is used to create a local binary pattern and represents the radius around the center pixel. The frequency is set to 1.
    - Neighbors: the number of sample points to form a circular area for a binary. Keep in mind: the more points you enter, the higher the computer cost. The frequency is set to 8.
    - Grid X: the number of cells in a horizontal direction. The more cells, the better the grid, the vector size of the emerging element. The frequency is set to 8.
    - Grid Y - the number of cells in a straight line. The more cells, the better the grid, the vector size of the emerging element. The frequency is set to 8. 2. Algorithm Training.
2. **Algorithm Training:**

    The first step is to train the model. To do so, we use a database that contains photos of people we would like to inform. Next, an ID (Roll No.) (also a number or personal name) for all images is added, so the algorithm can use this data to identify the input image and give the result. photos of the same person must have the same ID(roll no.). Since the training set has already been created, let's take a look at the LBPH process steps.

3. **Applying for LBP operation:**

Next thing is to create an intermediate image that defines the original image on a highway, with the light of the face. To do so, algorithmic law uses a window view, supporting the range of frames and neighbors.

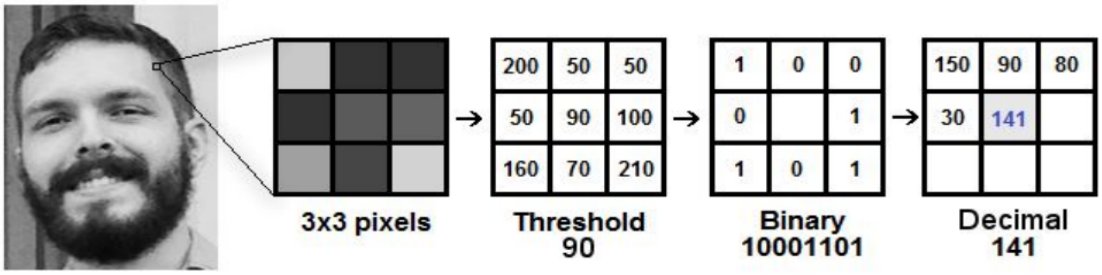

Figure 3: LBPH Operation

- We have a gray face picture.
- We can find part of this image as a 3x3 pixel window.
- It can also be represented as a 3x3 matrix containing the thickness of each pixel (0 255).
- After that, we need to take the median value of the matrix that will be used as the limit.
- This number will be used to describe new values from eight neighbors.
- For each median of the middle value (limit), we set a new binary value. We set 1 value equal to or higher than the limit and 0 value less than the limit.
- Now, the matrix will contain only binary values (regardless of average value). We need to estimate each binary value from each location from the matrix line by line to the new binary value (e.g. 10001101). Note: some authors use other methods to synchronize binary values (e.g. clock direction), but the result will be the same.
- After that, we convert this binary number to a decimal value and set it to the center value of the matrix, which is a pixel from the first image.
- At the end of this process (LBP process), we have a new image that better represents the features of the original image.

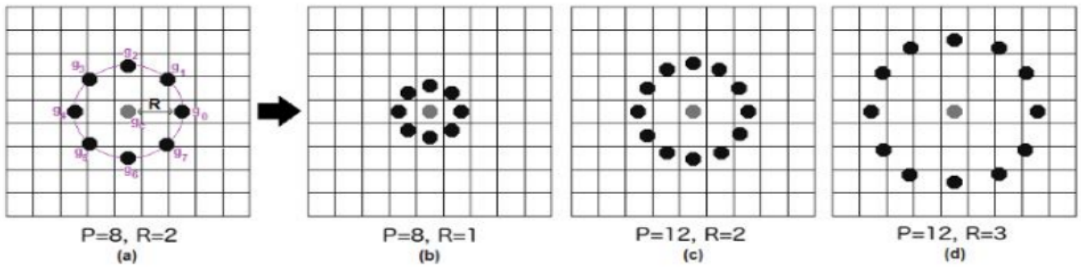

Figure 4: Applying LBPH Operation

- It can be done by using bilinear interpolation. If a certain point of data is within a pixel, it uses values from the nearest 4 (2x2) pixels to measure the point of the new data point.

4. **Recording Histograms:** Now, using the image created in the last step, we can use the Grid X and Grid Y parameters to split the image into multiple grids, as can be seen in the following picture:

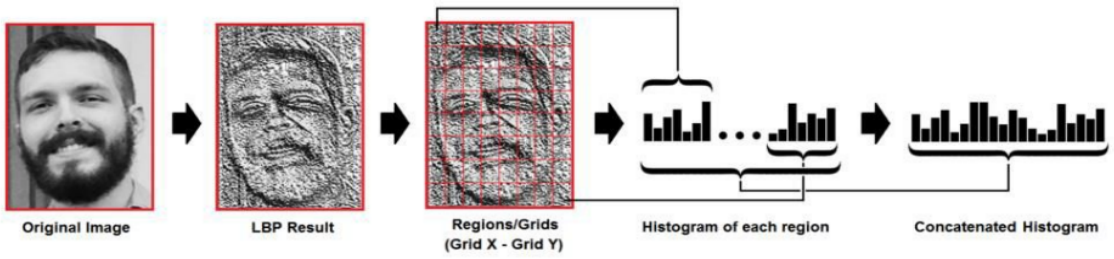

Figure 5: Extracting the Histograms

- Since we have a gray image, each histogram (from each grid) will only contain 256 (0 255) positions representing the potential for each pixel power.
- After that, we need to sync each histogram to create a new and larger histogram. If we assume we have 8x8 grids, we will have 8x8x256 = 16.384 positions in the final histogram. The final histogram represents the features of the first image.

5. **Performing face recognition:** At this point, the algorithm is already trained. Each created histogram is used to represent each image from the training database. So, when we are given an image to insert, we perform steps again for this new image and create a histogram representing the image.

- So to get an image similar to an input image we just need to compare the two histograms and replace the image with the nearest histogram.
- We can use various methods to compare histograms (calculating the distance between two histograms), for example, Euclidean distance, square-chi, total value, etc. In this example, we can use the Euclidean (most popular) range based on the following formula:

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

Figure 6: Euclidean Range

- So the output of the calculation is an ID from a picture with a close-by histogram. The calculation ought to likewise restore a determined reach, which can be utilized as a proportion of 'certainty'. Note: don't be tricked by the word 'certainty', since low camouflage is better since it implies that the separation between the two histograms is nearer.
- We can then use the threshold limit and 'confidence' to automatically adjust if the algorithm has detected the image correctly. We can assume that the algorithm has detected success when confidence is below the defined limit.

7 Architectural Overview:

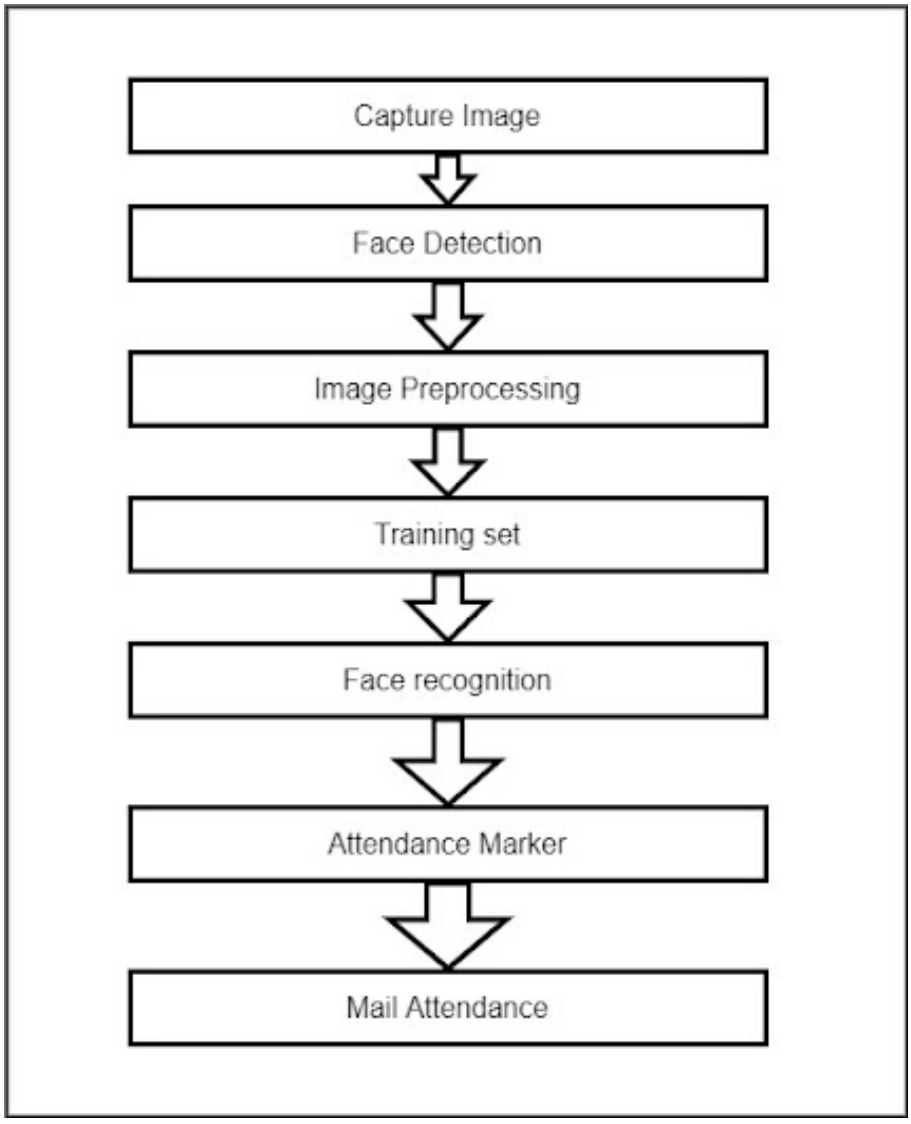

Figure 7: Architectural Overview

1. **Capturing the Image**

   The camera will be placed to get student's face images perfectly. Then it goes to further process of face detection.

2. **Face Detection**

   In this part, the system implements face detection, which helps to determine captured images with location and sizes of student faces. The image will be captured from detected faces

using the haar cascade classifier algorithm.

3. **Image Preprocessing**

   There is a preprocessing requirement for enhancing the input image, for improving the quality of image .This converts input image to grayscale image using color to gray image conversion technique.

4. **Training Set**

   Comparing the faces which are to be recognized with some other similar faces is the recognition process. Supply algorithm faces in training set to tell which person who belongs. When recognized face by algorithm, it uses the training set to make recognition.

5. **Face Recognition**

   The important part of this system is face recognition. Face recognition is an automatic method of identifying and verifying a person from the camera.

6. **Attendance marker**

   In this, the particular student's name will be marked in the students list. That is, collect the list of all students who were present in the class.

7. **Automail**

   In this project, we have added an extra feature called auto-mail. It automatically sends the attendance file to specific mail (specified in the system model). We just have to add the mail id of the respective faculty and automatically the system will mail the respective attendance sheet.

   Use Case:

   A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). A key concept of use case modeling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible

system behavior. It only summarizes some of the relationships between use cases, actors, and systems. It does not show the order in which steps are performed to achieve the goals of each use case.

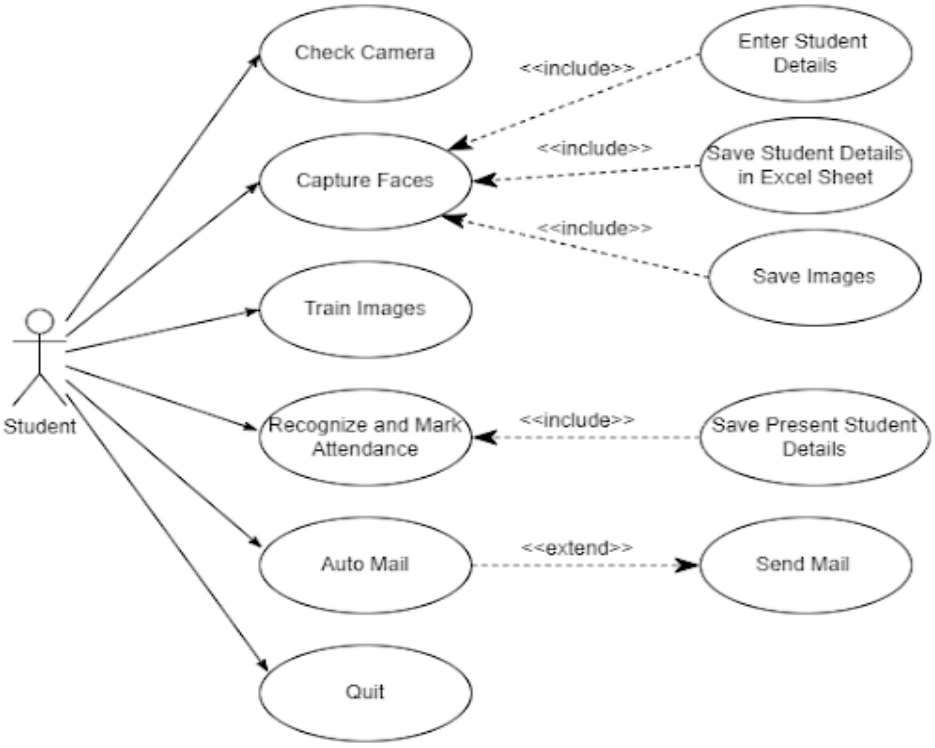

Figure 8: Use Case Diagram

As a user, the client wants a system where they can load an image that will automatically detect the number of faces on the image. The system should have the option to capture an image using an inbuilt webcam on a laptop. As a user, the system should be able to crop the faces on an image after detection and store them on a folder/dataset that will be used for recognition purposes in the second phase of the system. The system should be able to automatically count the number of faces detected on the image. As a user, the client requests the second phase of the system to be able to match faces stored on a dataset against input images which are either detected from the first phase or captured by an input device (camera). When the face is detected while taking attendance, it is recognized and details of name and time of attendance taken are entered in the database.

8 Resources: Hardware Software Requirements:

8.1 Hardware requirements:

Processor : intel core i3 6th gen onwards

RAM : 2 GB or higher

Hard Disk Space : 20 GB or higher

Webcamera

8.2 Software requirements:

Operating System : Windows 7 or higher.

Software : Visual Studio 2019 or higher.

Visual Studio Code-

Visual Studio Code is a source-code editor made by Microsoft for Windows , Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality

Figure 9: VS Code

Python –install python 3.9 (Programming language).

Python is an interpreter, high-level, and general-purpose programming language. Python's

design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects

Figure 10: Python

Numpy -pip install numpy

Numpy could be a library for Python, adding support for multi-dimensional arrays and matrices, in conjunction with an enormous assortment of high-level mathematical functions to operate on these arrays. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Figure 11: Numpy

Pandas - pip install pandas

Pandas is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language. It provides various data structures and operations for manipulating numerical data and time series.

Figure 12: Pandas

Datetime - pip install datetime

Datetime is a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and info.

Figure 13: Datetime

Face_Recognition-pip install face_recognition

Face_Recognition recognizes and manipulates faces from Python or the command line with the world's simplest face recognition library. It lets you do face recognition on a folder of images from the command line.

OpenCV- pip install opencv-python (face detection).

OpenCV is a library of programming functions primarily geared toward real-time computer vision. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, etc.

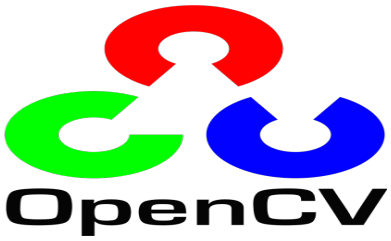

Figure 14: OpenCV

Pillow – pip install pillow

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF.

Figure 15: Pillow

9 Implementation

Below are the important python codes which are the core of our implementation.

1. Check_camera.py: To check whether the face is getting detected properly
2. Capture_Image.py: This will capture the images of the person
3. Train.py: Train the captured images and work on datasets.
4. Recognize.py: To recognize and mark attendance
5. automail.py: To send mail to a specific mail id.
6. main.py: Main program file to run the program

check_camera.py

In this step, the camera will be checked if it is working properly and connected with the hardware. Also to check whether the face is getting detected properly

```python
def camer():
    import cv2

    # Load the cascade
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

    # To capture video from webcam.
    cap = cv2.VideoCapture(0)

    while True:
        # Read the frame
        _, img = cap.read()

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

        # Detect the faces
        faces = face_cascade.detectMultiScale(gray, 1.3, 5, minSize=(30, 30),flags = cv2.CASCADE_SCALE_IMAGE)

        # Draw the rectangle around each face
        for (x, y, w, h) in faces:
            cv2.rectangle(img, (x, y), (x + w, y + h), (0,255,255), 2)


        # Display
        cv2.imshow('Webcam Check', img)

        # Stop if escape key is pressed
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    # Release the VideoCapture object
    cap.release()
    cv2.destroyAllWindows()
```

Figure 16: Code for Detecting Face

```
        ************************************************************
        ****************** Smart Attendance System ******************
        ************************************************************

********** WELCOME MENU **********
[1] Check Camera
[2] Capture Faces
[3] Train Images
[4] Recognize & Attendance
[5] Auto Mail
[6] Quit
Enter Choice: 1
```

Figure 17: Main menu

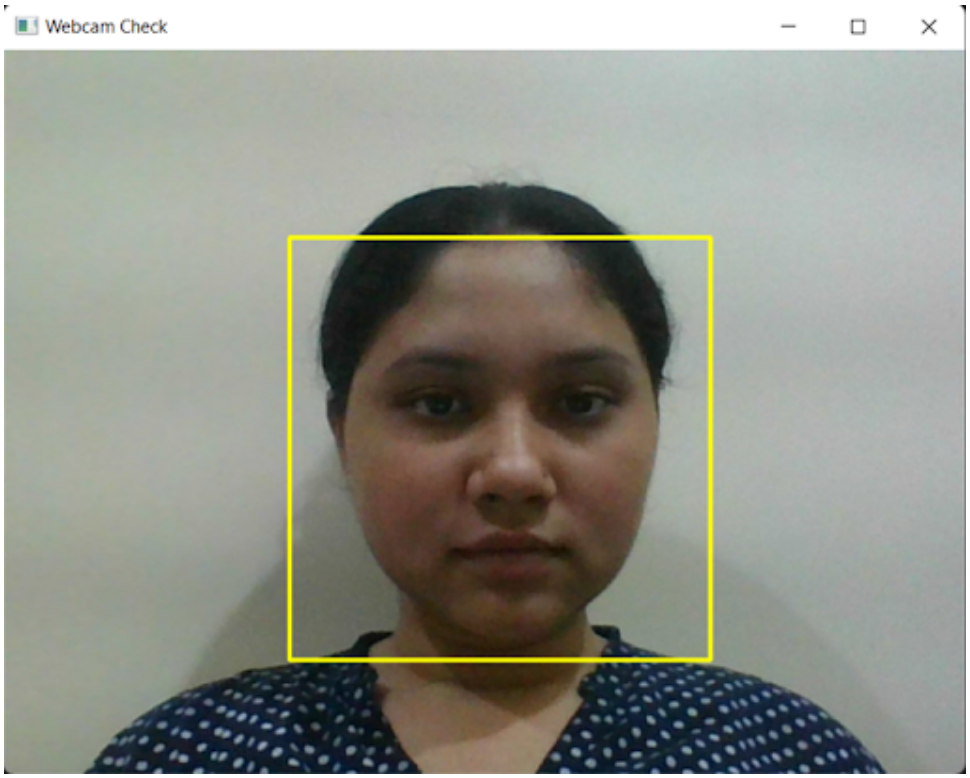

Figure 18: Face Detected by camera

Capture_Image.py

In Capture_Image.py, the camera will capture the images and take the student details i.e name, roll no. This Roll no, Name and Department is stored in the Student Details folder and file name is saved as Student Details.csv. It takes 100 images as samples and stores them in the TrainingImage folder. Haar Cascade Classifier is used here for face detection.

```
# Take image function

def takeImages():


    rollno = input("Enter Your Roll No.: ")
    name = input("Enter Your Name: ")
    dept = input("Enter Your Department: ")

    if(is_number(rollno) and name.isalpha() and dept.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector = cv2.CascadeClassifier(harcascadePath)
        sampleNum = 0
```

Figure 19: Code for taking Student Details as input

```
        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5, minSize=(30,30),flags = cv2.CASCADE_SCALE_IMAGE)
            for(x,y,w,h) in faces:
                cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255,0), 2)
                #incrementing sample number
                sampleNum = sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("TrainingImage" + os.sep +name + "."+ rollno + '.' +
                            str(sampleNum) + ".jpg", gray[y:y+h, x:x+w])
                #display the frame
                cv2.imshow('Capturing Images', img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is more than 100
            elif sampleNum > 100:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for Roll No. : " + rollno + " Name : " + name + " Department : " + dept
        row = [rollno, name, dept]
        with open("StudentDetails"+os.sep+"StudentDetails.csv", 'a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
        csvFile.close()
    else:
        if(is_number(rollno)):
            print("Enter Name:")
        if(name.isalpha()):
            print("Enter Roll No.:")
        if(dept.isalpha()):
            print("Enter Department:")
```

Figure 20: Capturing Images

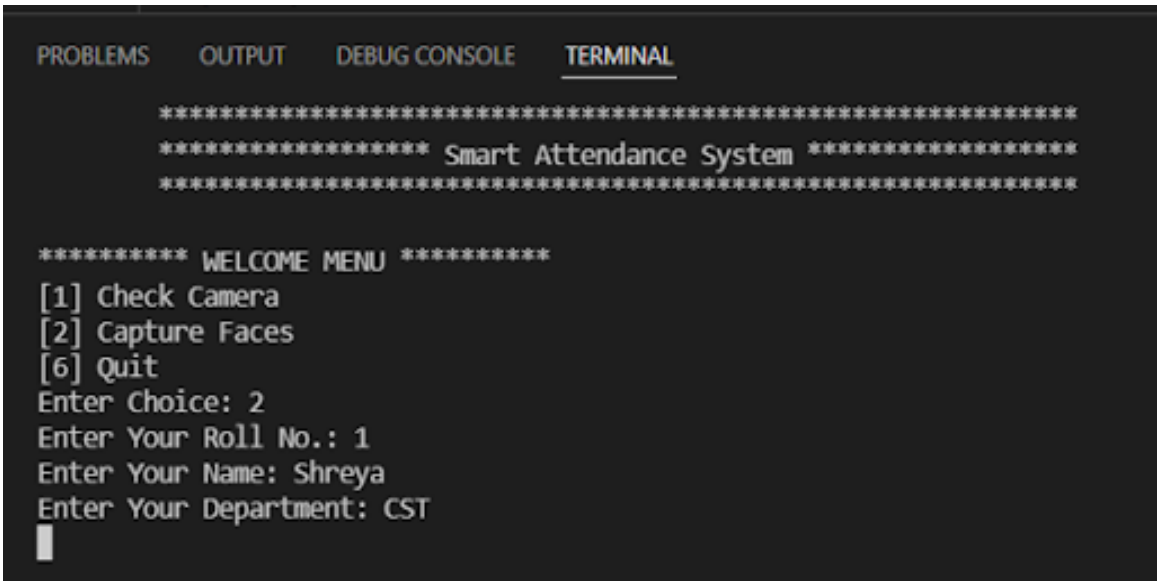

Figure 21: Taking Student Details

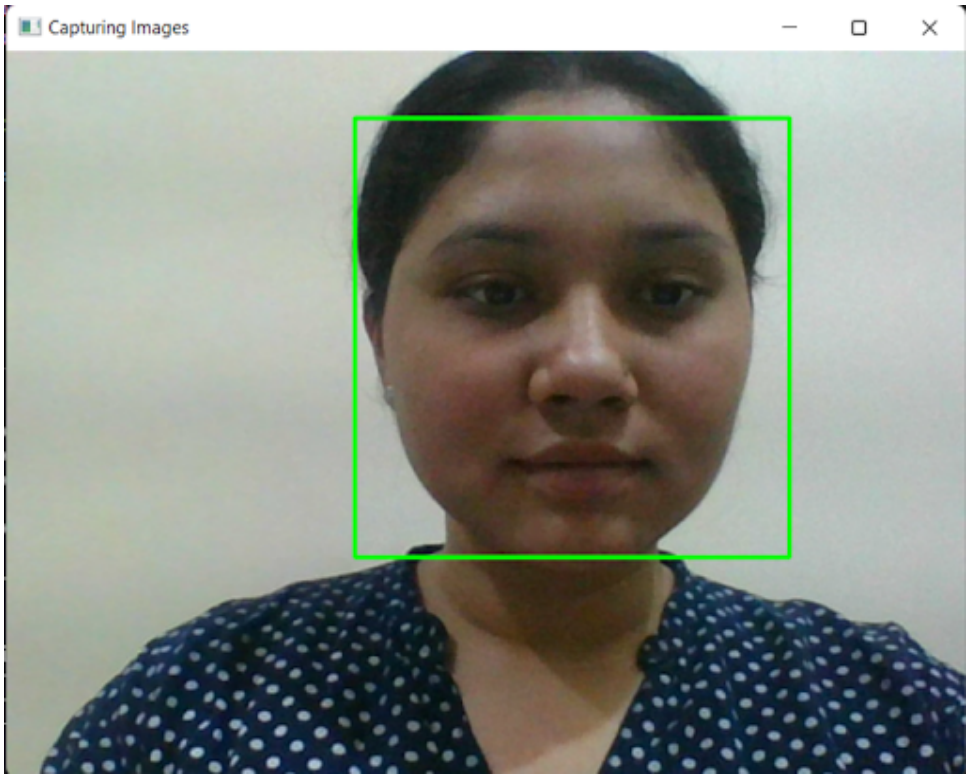

Figure 22: Capturing Images of the person

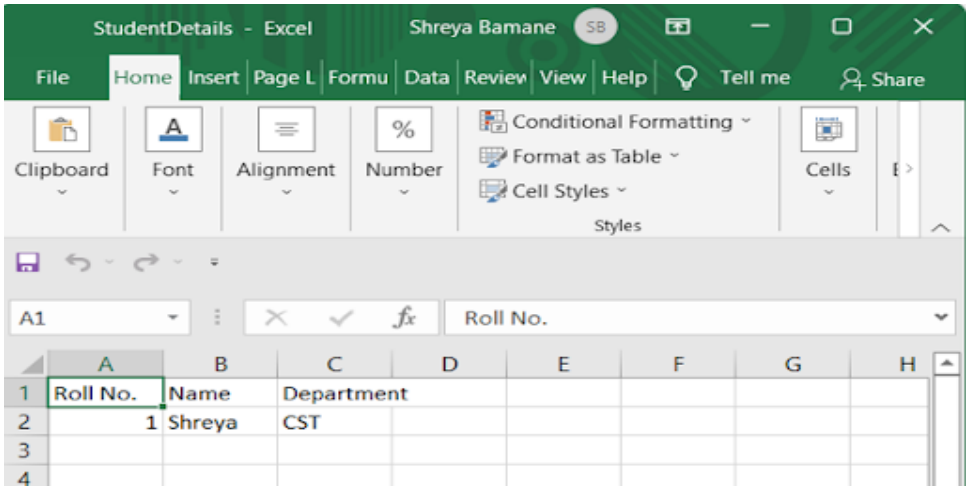

Figure 23: Details are saved to excel sheet

Figure 24: Images captured for training

Train_Image.py

In this step, the image gets trained with database images and machine learning techniques. This is done by making use of a haar cascade classifier. Once the images are trained and the model is ready for the next step, the “image trained“ message appears on the screen.

```python
# ---------- train images function ---------------
def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector = cv2.CascadeClassifier(harcascadePath)
    faces, rollno = getImagesAndLabels("TrainingImage")
    Thread(target = recognizer.train(faces, np.array(rollno))).start()
    # Below line is optional for a visual counter effect
    Thread(target = counter_img("TrainingImage")).start()
    recognizer.save("TrainingImageLabel"+os.sep+"Trainner.yml")
    print("All Images")

# Optional, adds a counter for images trained (You can remove it)
def counter_img(path):
    imgcounter = 1
    imagePaths = [os.path.join(path, f) for f in os.listdir(path)]
    for imagePath in imagePaths:
        print(str(imgcounter) + " Images Trained", end="\r")
        time.sleep(0.008)
        imgcounter += 1
```

Figure 25: Code to Train image

```
        ************************************************************
        ****************** Smart Attendance System ******************
        ************************************************************

********** WELCOME MENU **********
[1] Check Camera
[2] Capture Faces
[3] Train Images
[4] Recognize & Attendance
[5] Auto Mail
[6] Quit
Enter Choice: 3
All Images Trained
Enter any key to return main menu
```

Figure 26: Training Images

Recognize.py

In this step, the faces are recognized with the help of the LBPH Face recognizer function. Along with this, the time and date also get recorded. If the function works successfully then the attendance is marked for the recognized face.

```
#------------------------
def recognize_attendence():
    recognizer = cv2.face.LBPHFaceRecognizer_create()  # cv2.createLBPHFaceRecognizer()
    recognizer.read("TrainingImageLabel"+os.sep+"Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath)
    df = pd.read_csv("StudentDetails"+os.sep+"StudentDetails.csv")
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names = ['Roll No.', 'Name', 'Date', 'Time']
    attendance = pd.DataFrame(columns=col_names)

    # Initialize and start realtime video capture
    cam = cv2.VideoCapture(0, cv2.CAP_DSHOW)
    cam.set(3, 640)  # set video width
    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)
```

Figure 27: Function to recognize the person

Figure 28: Recognising the person

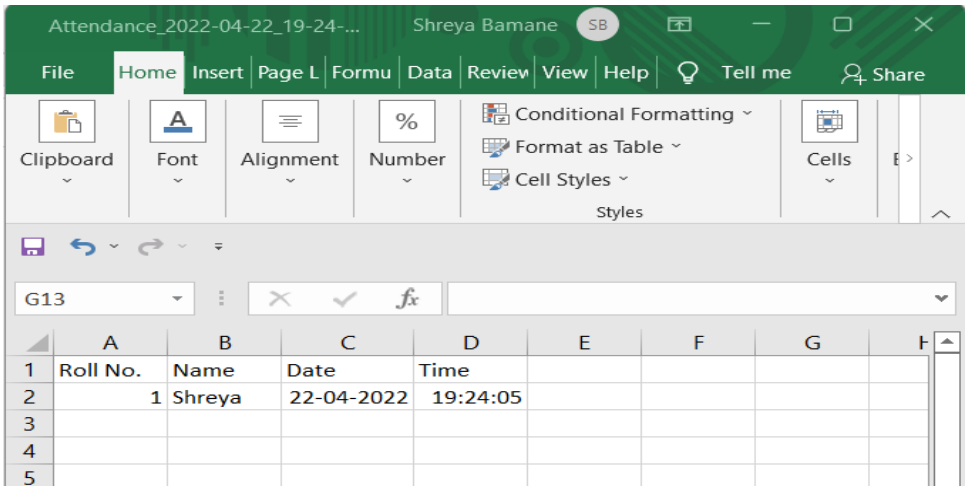

Figure 29: Student Attendance saved in Excel sheet

automail.py

In this project, we have added an extra feature called auto-mail. It automatically sends the attendance file to specific mail (specified in the system model). The code for the same is given below:

```
import yagmail
import os

receiver = "shreyabamane21@gmail.com.com"  # receiver email address
body = "Attendence File"  # email body
filename = "Attendance"+os.sep+"Attendance_2022-04-22_19-24-18.csv"  # attach the file

# mail information
yag = yagmail.SMTP("bamanerani21@gmail.com", "Ranibamane@21")

# sent the mail
yag.send(
    to=receiver,
    subject="Attendance Report",  # email subject
    contents=body,  # email body
    attachments=filename,  # file attached
)
```

Figure 30: Code to mail the attendance sheet

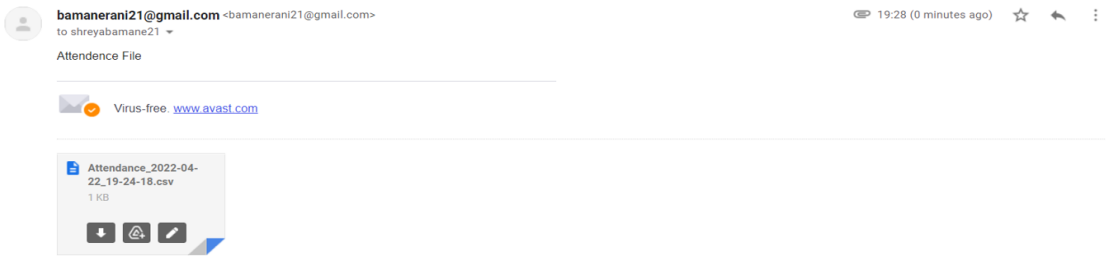

Figure 31: Sheet mailed to the receiver

main.py

This is the main python code of our project that combines functions from other files and has all the options allowed in the user interface. Once this file has been launched, the user gets to select any action to be performed, say whether it is for capture, or abort etc. The main function calls for other functions and specifies options for the user.

```python
def title_bar():
    os.system('cls')  # for windows

    # title of the program

    print("\t**********************************************************")
    print("\t****************** SmartAttendance System ******************")
    print("\t**********************************************************")


# creating the user main menu function

def mainMenu():
    title_bar()
    print()
    print(10 * "*", "WELCOME MENU", 10 * "*")
    print("[1] Check Camera")
    print("[2] Capture Faces")
    print("[3] Train Images")
    print("[4] Recognize & Attendance")
    print("[5] Auto Mail")
    print("[6] Quit")

    while True:
        try:
            choice = int(input("Enter Choice: "))

            if choice == 1:
                checkCamera()
                break
            elif choice == 2:
                CaptureFaces()
                break
            elif choice == 3:
                Trainimages()
                break
            elif choice == 4:
                RecognizeFaces()
                break
            elif choice == 5:
                os.system("py automail.py")
                break
                mainMenu()
            elif choice == 6:
                print("Thank You ")
                break
            else:
                print("Invalid Choice. Enter 1-4")
                mainMenu()
        except ValueError:
            print("Invalid Choice. Enter 1-4\n Try Again")
    exit
```

Figure 32: Creating User main menu

```
# ------------------------------------------------------------
# calling the camera test function from check camera.py file

def checkCamera():
    check_camera.camer()
    key = input("Enter any key to return main menu")
    mainMenu()


# ------------------------------------------------------------------
# calling the take image function form capture image.py file

def CaptureFaces():
    Capture_Image.takeImages()
    key = input("Enter any key to return main menu")
    mainMenu()


# ------------------------------------------------------------------
# calling the train images from train_images.py file

def Trainimages():
    Train_Image.TrainImages()
    key = input("Enter any key to return main menu")
    mainMenu()


# ------------------------------------------------------------------
# calling the recognize_attendance from recognize.py file

def RecognizeFaces():
    Recognize.recognize_attendence()
    key = input("Enter any key to return main menu")
    mainMenu()


# ---------------main driver ------------------
mainMenu()
```

Figure 33: Calling all the functions

10 Applications

1. This smart attendance system can be used to verify identities in Government organizations and mark their presence.
2. Also, it can be used in Enterprises to mark employee's attendance and to check whether they are punctual or not .
3. Taking attendance in schools and colleges would be easier, accurate and quick.
4. Fake entries at international borders can be detected

5. Reporting time of labors can be stored in Industries.

11 Future Scope

The future work is to improve the recognition rate of algorithms when there are unintentional changes in a person like a tonsuring head or using a scarf. Another important aspect where we can work towards is creating an online database of the attendance and automatic updating of the attendance into it keeping in mind the growing popularity of Internet of Things. These Developments can greatly improve the applications of the project. Besides, we can simplify the system and make it more efficient by taking advantage of multiple face detection to mark attendance of all the visible faces in a single attempt. This will be an economical and more efficient use of face recognition for attendance marking. We are also considering developing an android application for this system in the near future.

12 Conclusion

Thus, the aim of our project is to capture the images of the students, convert it into frames, relate it with the data to ensure their presence and the mark attendance to the particular student to maintain the record. This system is designed to minimize the human effort for taking the attendance manually that takes place in every college. As the attendance marking process is done without any human interference, here we make it automatic which is the main scope in the system. The Smart Attendance System using Face Recognition helps in increasing the accuracy and speed ultimately achieve the high-precision real- time attendance to meet the need for automatic classroom evaluation. As this system does not require touching anything so risk of germs transmission is also eliminated. Due to Covid-19, it is required by the government that social distancing must be followed in all organizations. If such systems are installed throughout the campus we can ensure that social distancing is followed everywhere in the campus. It will also help the institute by managing attendance and to avoid unknown entry to the institution. At the end, the system not only

resolves troubles that exist in the old model but also provides convenience to the user to access the information collected by mailing the attendance sheet to the respected faculty.

References

[1] Niket Nagwani et al. “An Attendances System Unit Using the Radio Frequency Identification Concept”. In: *Journal of Computer and Mathematical Sciences* 10.4 (2019).

[2] Xiong Wei et al. “QR Code Based Smart Attendance System”. In: *International Journal of Smart Business and Technology* 5.1 (2017).

[3] Devendra Kumar Yadav et al. “Fingerprint based attendance system using microcontroller and LabView”. In: *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 4.6 (2015).

[4] Subhadeep Dey et al. “Speech biometric based attendance system”. In: *2014 twentieth national conference on communications (NCC)*. IEEE. 2014.

[5] Kennedy O Okokpujie et al. “Design and implementation of a student attendance system using iris biometric recognition”. In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2017.

[6] Vincentius Kurniawan, Arya Wicaksana, and Maria Irmina Prasetiyowati. “The implementation of eigenface algorithm for face recognition in attendance system”. In: *2017 4th International Conference on New Media Studies (CONMEDIA)*. IEEE. 2017.

[7] XueMei Zhao and ChengBing Wei. “A real-time face recognition system based on the improved LBPH algorithm”. In: *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*. IEEE. 2017.