

Project Report, Requirement and Specification

on Face Recognition Based Attendance system

Problem Statement

The importance of attendance in any organization cannot be overstated. Manually taking attendance risks missing someone or taking attendance of the same person more than once. Face recognition-based attendance system is a solution to the problem of recognizing faces for the purpose of collecting attendance by utilizing face recognition technology based on high-definition monitor video and other information technology. Face recognition is the ability of a computer system to detect and recognize human faces quickly and accurately in real time. Multiple faces can be detected and recognized automatically and instantly by the human brain. However, on the level of the human brain, it is extremely impossible to perform all of the tough activities on a computer. Face recognition is an important component of biometrics. Basic human characteristics are linked to current data in biometrics. Facial traits are retrieved and implemented using efficient algorithms.

About the project

This is a python-based application which provides an involuntary attendance marking system that operates without human intervention. It intends to serve as an efficient substitute for traditional manual attendance systems. It can be used in corporate offices, schools, and organizations where security is essential.

The system can also be used to maintain track of employees or students or other instructional activities where attendance is critical. In order to be recognized, students or employees must also register in the database. The user-friendly interface allows for on-the-spot registration. Using an admin account and password, the organization's admin can email the attendance of the day to the respective email address. Only the organization's administrators will have access to the attendance information, and no changes will be made to the attendance details.

It aims to automate the traditional attendance system where the attendance is marked manually. It also enables an organization to maintain its records like in-time, out time, break time and attendance digitally. Digitalization of the system would also help in better visualization of the data using graphs to display the no. of employees present today, total work hours of each employee and their break time. Its added features serve as an efficient upgrade and replacement over the traditional attendance system.

Application

Face recognition is one of the researches in area pattern recognition & computer vision due to its numerous practical applications in the area of **biometrics, Information security, access control, law enforcement, smart cards and surveillance system.**

Motivation

We strive to deliver a valued face recognition-based attendance service wherever attendance is critical or important for the functioning of the organisation. Reduce manual process errors by using facial recognition technology to deliver an automated and dependable attendance system. The attendance system will be more authentic as we develop a system that can recognise faces and compare them with its own database. Our main purpose is to streamline and organise the tracking and management of attendance and absenteeism. Also, increase privacy and security so that no proxy or alteration is possible.

Requirements of the project

Hardware Requirement

- Microphone
- Speaker
- Minimum 4.00 GB RAM (for the system to work smoothly)

Software Requirement

- Any python compiler like Pycharm, Spyder, Jupyter Notebook etc.
- Microsoft Visual Studio
- Any 64-bit operating system, x64-based processor

Methodology

Face detection is the act of recognizing and locating all of the faces contained in a single image or video, independent of their size, scale, orientation, age, or emotion. Face Recognition, on the other hand, is a visual pattern recognition problem in which the face, which is represented as a three-dimensional object that is sensitive to fluctuating illumination, position, and other aspects, must be identified from recorded photos. The issue of "Where is the face?" is answered through face detection. It recognizes an object in the input image as a "face" and locates it. Face Recognition, on the other hand, provides an answer to the question of who this person is Or, to put it another way, whose face is it? Based on the database of faces it uses to check this input image, it determines whether the identified face is someone recognized or unknown. As can be

observed, the output of face detection (the detected face) is the input to the face recognizer, and the output of the face recognizer is the final conclusion, i.e., face known or face unknown. Below is the flowchart and block diagram of the face recognition-based attendance system which depicts the complete working of the project.

Below is the flowchart and block diagram of the face recognition-based attendance system which depicts the complete working of the project.

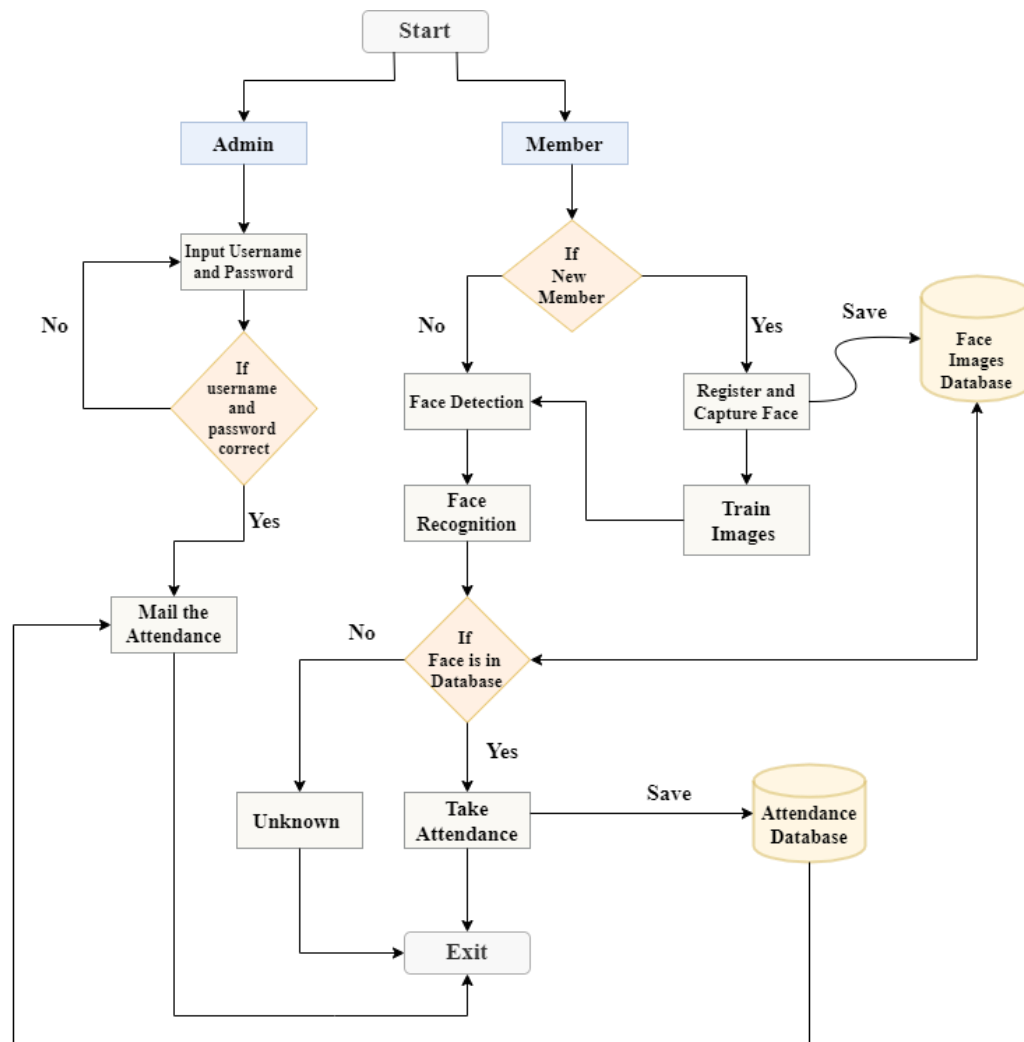


Figure 1. Flowchart of the Project

The task of face recognition can be divided into two parts. The first phase is a classification task, which accepts any image as input and returns a binary result of yes or no, indicating whether the image contains any faces. The face localization task, which takes an image as input and outputs the location of any face or faces within that picture as a bounding box with (x, y, width, height),

is the second stage. After capturing the image, the system will evaluate the pictures in its database for equality and return the most relevant result. For the above working, Local Binary Patterns Histogram (LBPH) Algorithm and Haar Cascade Classifier of OpenCV is used for capturing face images, training images, face detection and recognition for the attendance system.

OpenCV

OpenCV is a cross-platform library that can be used to create real-time computer vision applications. It focuses primarily on image processing, video capture, and analysis, with capabilities such as face detection and object detection. To find faces in an image or video or real-time webcam, OpenCV use machine learning methods because faces are so complex, there is no easy way that can tell you whether or not it identified a face. Instead, thousands of little patterns and characteristics must be matched to accurately identify the face of a human. The algorithms divide the work of recognising the face into thousands of smaller, manageable tasks known as Classifiers.

The OpenCV library has the following features:

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyse the video, including estimating motion, removing the background, and tracking objects.

Local Binary Patterns Histogram (LBPH) Algorithm

Local Binary Pattern (LBP) is a basic yet effective texture operator that labels pixels in an image by thresholding each pixel's neighbourhood and treating the result as a binary number. On some datasets, it has also been discovered that combining LBP with the histograms of oriented gradients (HOG) descriptors boosts detection performance significantly. We can represent the facial photos with a simple data vector using the LBP and histograms. LBP can be utilised for face recognition applications because it is a visual descriptor.

The LBPH algorithm has five steps:

1. **Parameters:** the LBPH uses 4 parameters:

- **Radius:** The radius represents the radius surrounding the centre pixel and is used to generate the circular local binary pattern. Normally, it is set to 1.
- **Neighbours:** The number of sample points required to construct a circular local binary pattern. Normally, it is set to 8.
- **Grid X:** The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. Normally, it is set to 8.
- **Grid Y:** The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. Normally, it is set to 8.

2. Training the Algorithm: To train the algorithm, we'll need a dataset containing the faces of the persons we're trying to recognise. We must also assign each image an ID (which might be a number or a person's name) so that the algorithm can recognise an input image and provide you with an output. The ID for all images of the same individual must be the same.

3. Applying the LBP operation: The LBPH's first computational step is to build an intermediate image that better describes the original image by highlighting the facial features. To do so, the algorithm employs a sliding window idea based on the radius and neighbours' parameters.

The image below shows this procedure:

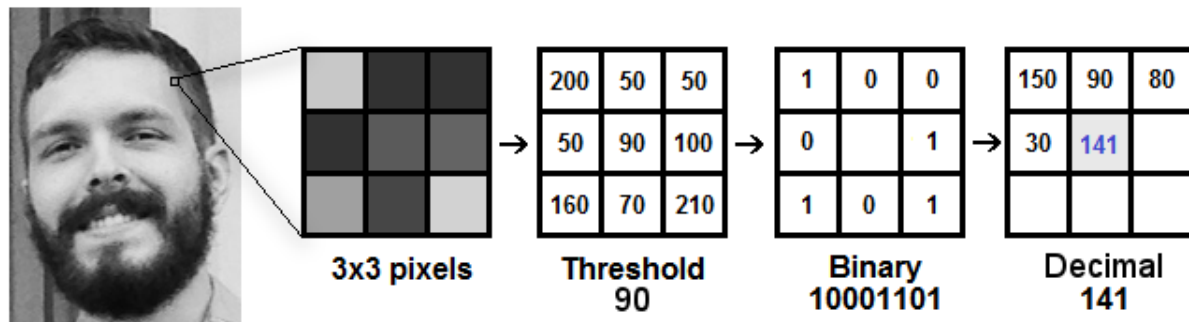


Figure 2. LBPH Operation

4. Extracting the Histograms: Using the image created in the previous phase, we can now divide the image into numerous grids using the Grid X and Grid Y parameters, as can be seen in the following image:

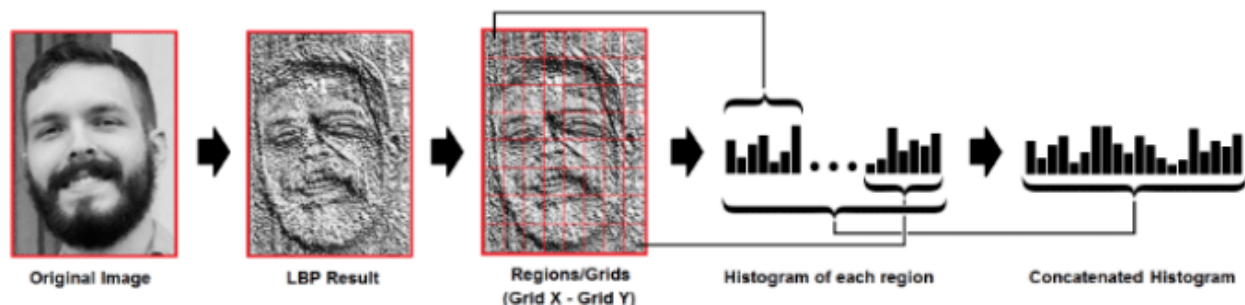


Figure 3. Extracting The Histogram

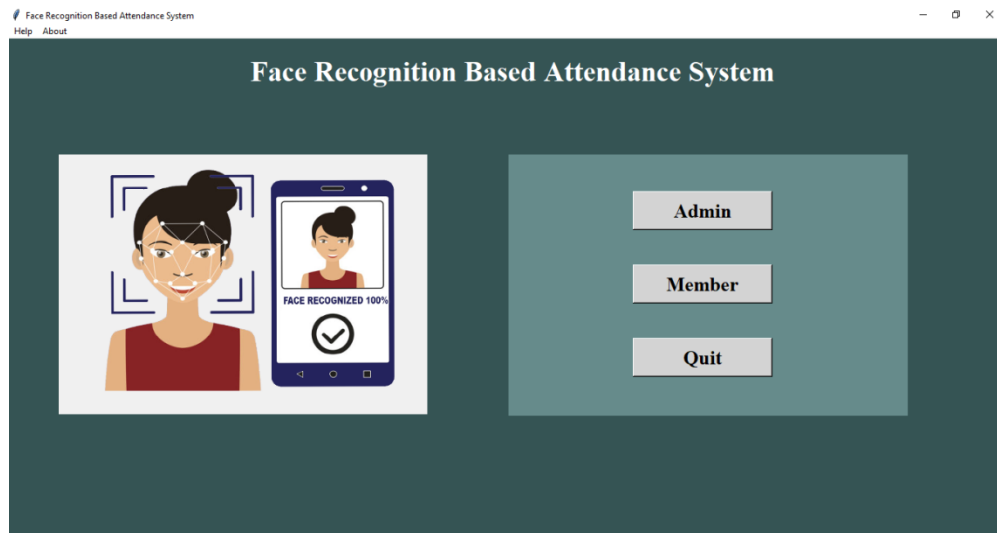
5. Performing face recognition: The algorithm has already been trained at this point. Each image from the training dataset is represented by a different histogram. As a result, given an input image, we repeat the procedures for this new image, resulting in a histogram that describes the image. So, all we have to do to identify the image that matches the input image is compare two histograms and return the image with the closest histogram. As a result, the ID from the image with the closest histogram is the algorithm's output. The calculated distance between the histograms, which can be used as a 'confidence' measurement, should also be returned by the algorithm. Note that despite the label 'confidence,' lower confidences are desirable because they indicate a closer distance between the two histograms. Then we can use a threshold and the 'confidence' to determine whether the algorithm accurately recognized the image. If the confidence is less than the stated threshold, we can assume that the algorithm has effectively recognized it.

Haar Cascade Classifier

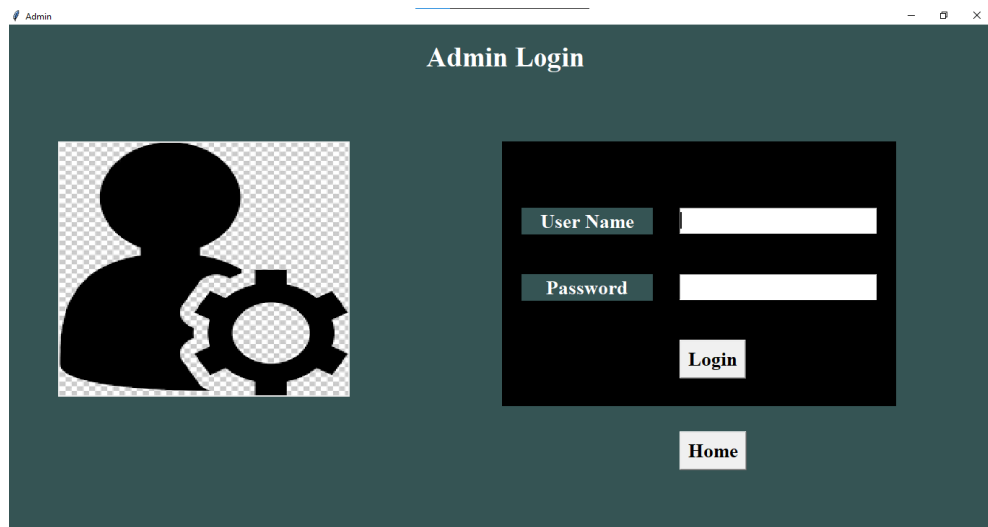
The Haar Cascade classifier is a machine learning-based object detection approach in which a cascade function is trained from a large number of positive and negative images. It is then used to detect the things in the other images based on the training. The majority of the images we encounter are RGB channel images (Red, Green, Blue). As a result, when OpenCV reads an RGB image, it stores it in the BGR (Blue, Green, Red) channel. This BGR channel must be converted to gray channel for image recognition purposes. The reason for this is because gray channel is simple to process and computationally less intensive because it just has one black-and-white channel. We use an inherent function called `detectMultiScale` with the face cascade classifier, which is an object loaded using `haar cascade frontal face default.xml` file. This function will assist us in locating the new image's features and locations. To detect the features of the new image, it will use all of the features from the face cascade classifier object. The function `detectMultiScale` returns 4 values — x-coordinate, y-coordinate, width(w) and height(h) of the identified facial feature. It will create a rectangle around the face based on these four variables. In this manner, it will detect and recognize the person's face, as well as identify the person's name and ID for the attendance.

GUI Interface

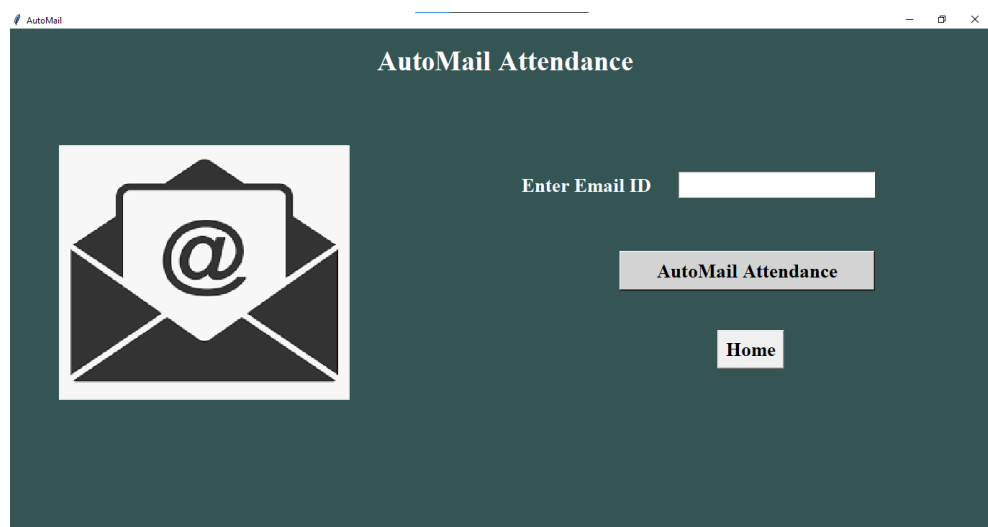
The snapshots below demonstrate the GUI Interface of the attendance system, which is divided into several sections as mentioned below.



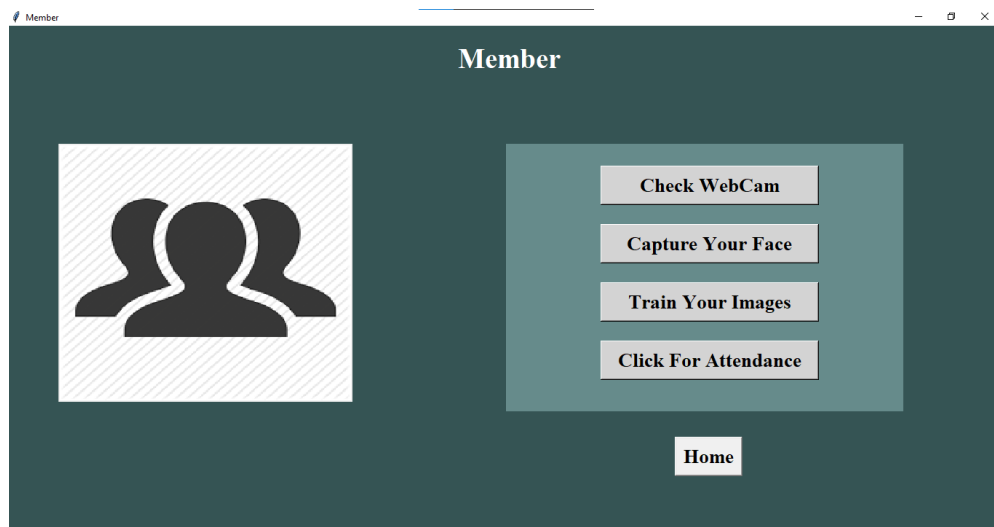
Main Window: It contains the options such as Admin, Member and Quit. The user will choose the preferred option based on its accessibility.



Admin Login Window: It is used to input the username and password of the admin account. If the username and password match with the admin account then it will login to the next window otherwise it will prompt you to re-enter the username and password.



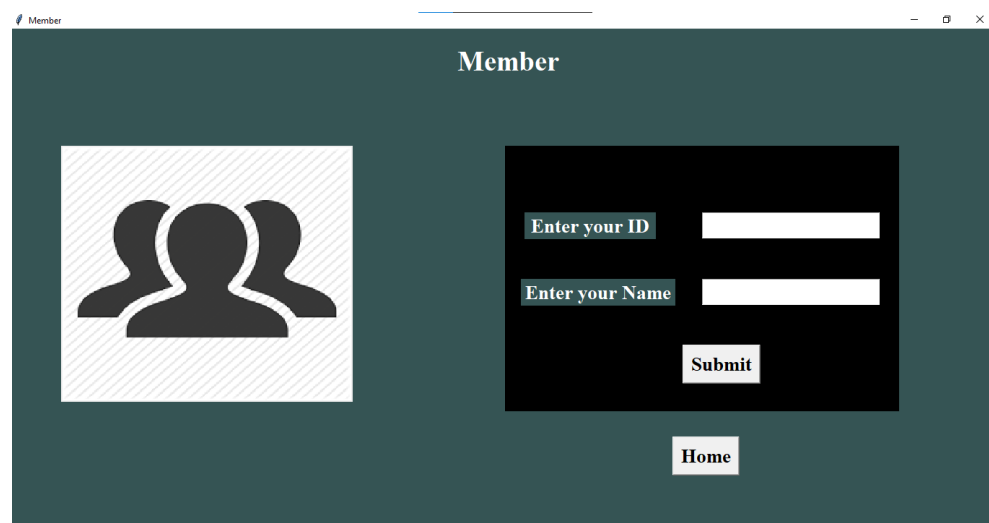
AutoMail Window: This window allows the admin to mail the attendance report to the respective mail ID from the admin's account credentials. This window will login only when the user will be able to provide the correct admin username and password.



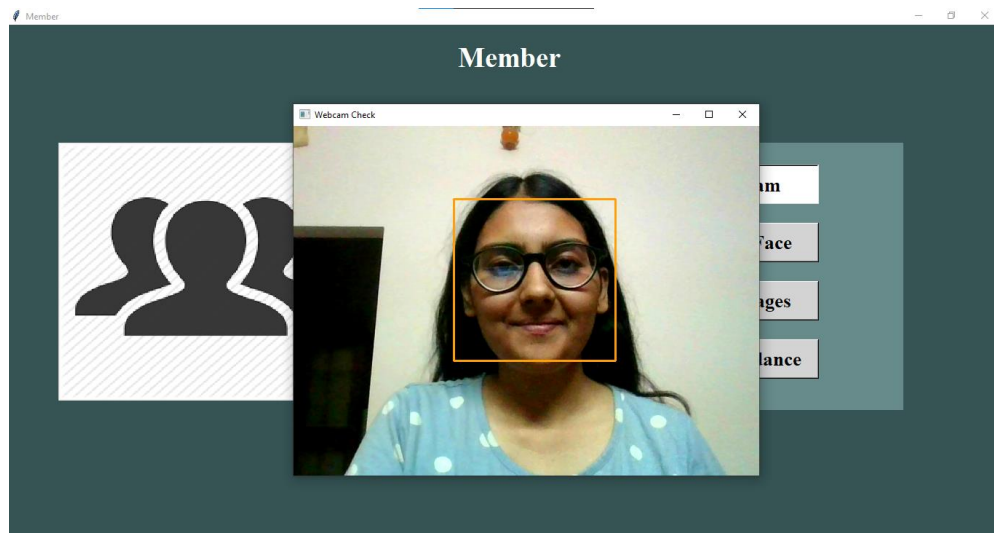
Member Window: This window allows members of the organisation to register themselves and record their attendance. It contains options such as Check Webcam, Capture Your Face, Train Your Images and Click for attendance for capturing attendance.



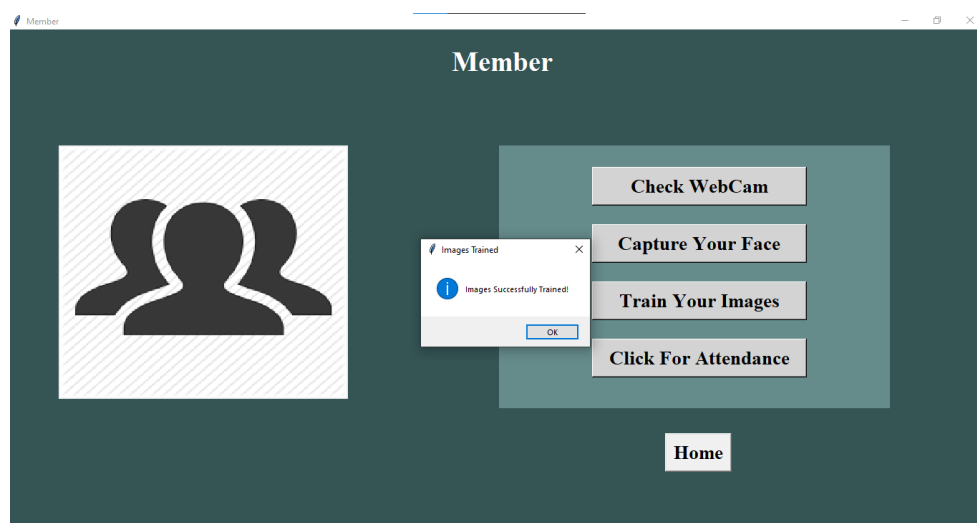
Check WebCam: It will check whether or not the system's Webcam is functioning properly.



Capture Your Face: This option will first ask for the member's ID and name in order to register them in the database, and then it will capture the member's face and save it to the Images database.



This window will capture the face of the member in order to create the Images dataset.



Train Your Images: This option will train all the images present in the database for the face recognition in order to record the attendance and will prompt the message when images are successfully trained.

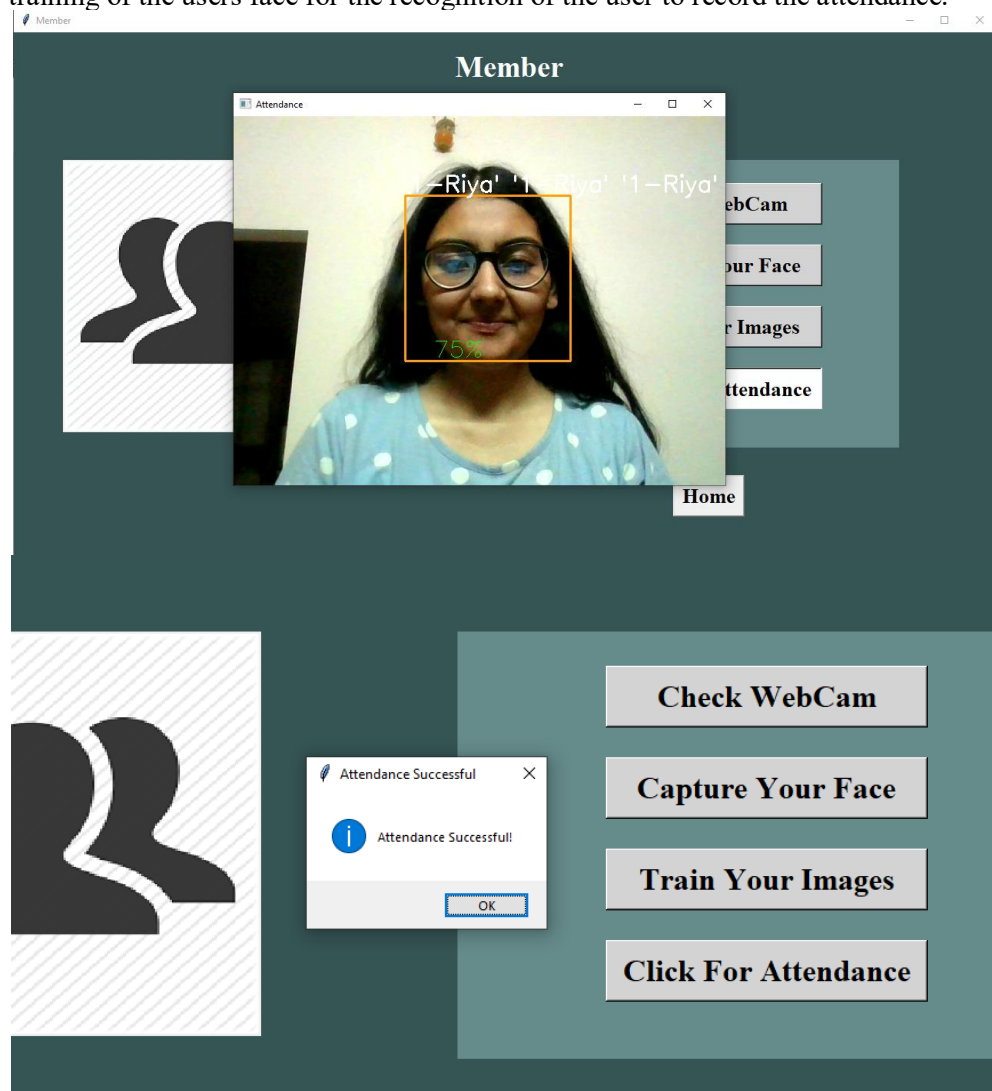
Click for Attendance: This option will launch the webcam and capture the user's face for face detection and recognition to record the attendance, as well as display the member's ID and name.

Result

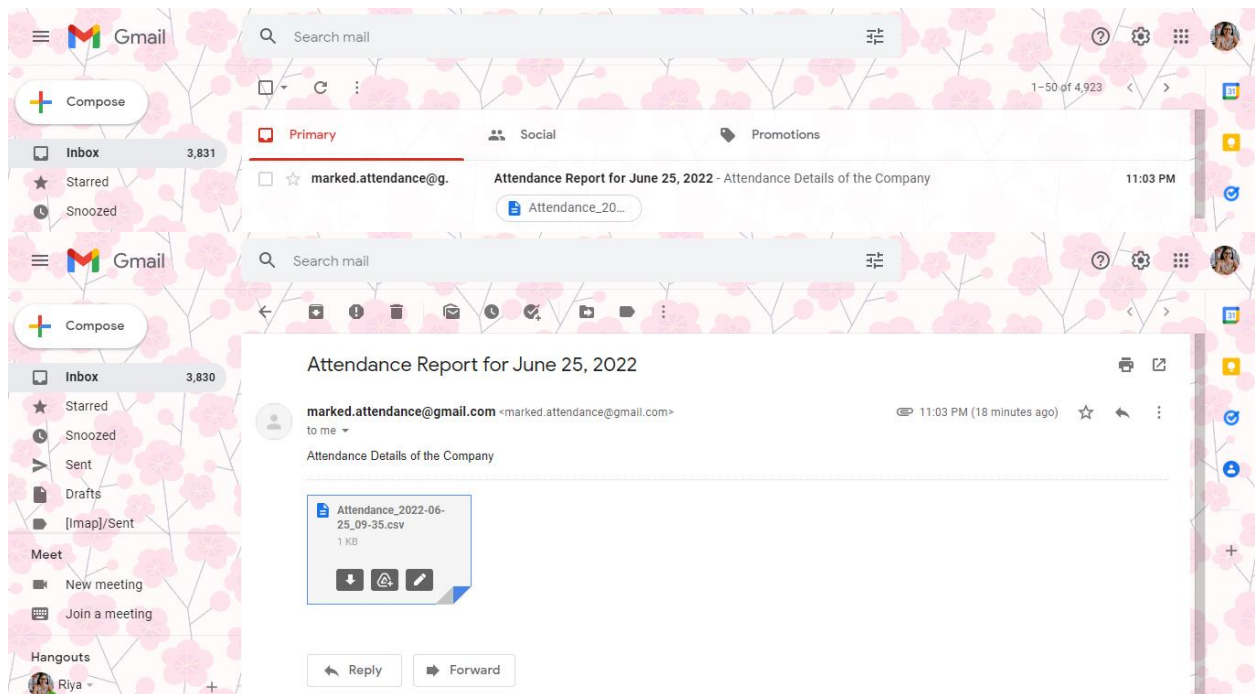
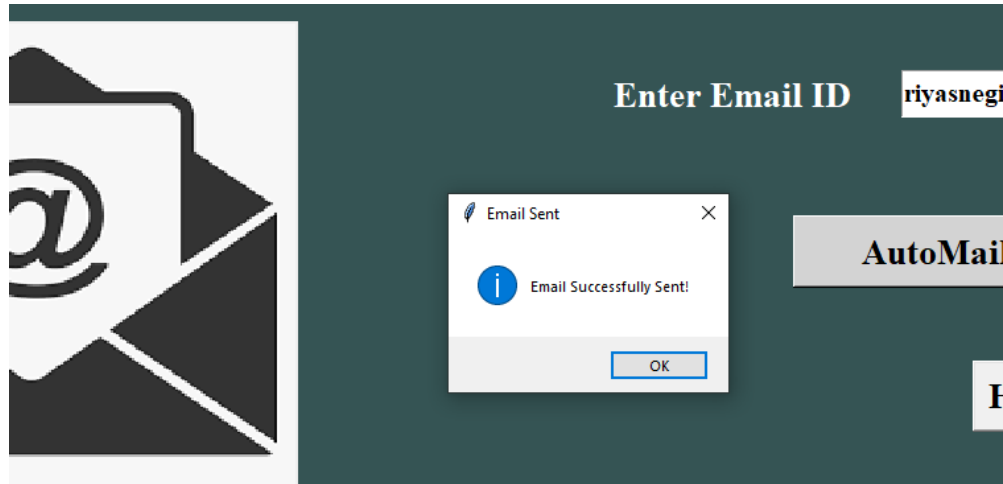
We tested the attendance-based system with a range of users, and it was able to recognise the users' faces with their names and IDs, as well as successfully recorded their attendance. The following are a few snapshots of the attendance taken. We were able to obtain a level of accuracy of around 80%.



Above are the snapshot of the images being captured and saved in the desired file location in the system for further training of the users face for the recognition of the user to record the attendance.



The above snapshots illustrate the accuracy of the user identified by the attendance-based system and above a certain threshold or accuracy it will record the attendance and will create a .csv file in the system with the details of the user with date and time.



The above snapshots demonstrate the information of the users saved in the .csv file which is the attendance recorded for the day and another snapshot demonstrates the mail being sent from the admin's account to the respective email id with the .csv attendance file attached in the mail.

Conclusion

In this system, we implemented a face recognition-based attendance system for a school, college, or any other organisation to track students and employee's attendance. It will save time and effort, especially if the organisation has a significant number of employees or pupils. The goal of the Automated Attendance Method is to eliminate the shortcomings of the traditional (manual) system. A face recognition-based attendance system is discussed in detail in this report. The suggested method identifies individuals by comparing their input image from a webcam to the train image. This suggested method could detect and localise a face from an input facial image received from a video frame being recorded. OpenCV's LBPH and Haar Cascade Classifier are used to record attendance using face detection and identification. This proposed technique has an accuracy of around 80%. The project was effective in registering the user, identifying the user's face, and recording the user's attendance. It was able to successfully send an email from the admin's account to the required email address using the admin's username and password. This technique makes it easier to keep track of attendance. It also saves time by reducing paperwork and automating the system's functioning. It's reliable and user-friendly, and it enhances productivity.