FACE RECOGNITION ATTENDANCE SYSTEM

Vidushi[1], Muskan Choudhary[2], Avni Tyagi[3]
KIET Group of Institutions, Delhi-NCR, Ghaziabad[1,2,3,4]

ABSTRACT

Face recognition technology is essential in today's digital world and is used in practically every industry. One of the most common types of biometrics is face recognition. It has numerous advantages and can be used for security, identification, and authentication. Even though it performs less accurately than iris and fingerprint identification, it is nonetheless commonly used since the method is non-intrusive and contactless. Additionally, facial recognition technology can be used in businesses, schools, colleges, and other places to track attendance. The goal of this system is to create a facial recognition-based class attendance system because the current manual attendance method is time-consuming and difficult to maintain. Additionally, there's a possibility that a proxy will show up. Consequently, demand for this system rises. The four phases of this system are database building, face detection, face recognition, and attendance updating. Images of the students in class are used to develop databases. Using OpenCV and Dlib, face detection and identification are carried out. Faces are recognised and detected by the live-streaming camera. Every 24 hours, attendance will be updated, and a window will display total attendance to date, the student's name, major, year, and branch.

KEYWORDS

Face Recognition, Face Detecting, OpenCV, Dlib

INTRODUCTION

Recording attendance using the conventional method can be a time-consuming and arduous task in many educational institutions, including universities. Additionally, in order to record attendance, the faculty members need to manually call out each student's name, which might take up to five minutes for the entire session. This consumes a significant amount of time. There is always a chance that someone will show up through a proxy. As a direct consequence of this, a number of different establishments have started employing a wide range of extra techniques to monitor attendance. These techniques include radio frequency identification (RFID), iris recognition, fingerprint recognition, and a few more. However, these technologies work on a queue, which means that the process could be slower and more intrusive than usual.

The ability to recognise an individual's face has emerged as an essential component of biometric authentication since it is unobtrusive and straightforward to acquire. Systems that rely on face recognition are typically unable to differentiate between a variety of expressions on the face. A

face recognition system is comprised of two categories: verification and face identification. Both of these aspects are necessary for the system to function properly. In contrast to a 1:N problem, which compares a query face image to a template face image, face verification is a 1:1 matching technique that compares a face image to a template face image. This is done in order to verify the identity of a person.

The purpose of his system is to develop methods of facial recognition that can be used to establish an attendance system. When keeping track of who was there at this event, a person's face will be taken into consideration. The use of technologies that can recognise people's faces is becoming increasingly widespread in today's society. In the course of this study, we came up with the idea of a system that could recognise the faces of students captured by live streaming cameras. The system would then record the student's attendance if the captured face was identified in a database. The new approach will require significantly less time investment when compared to the older methods.

LITERATURE REVIEW

The fundamental objective of this paper's review is to identify the solutions offered by the authors of other works, take into account the shortcomings of the systems supplied by those authors, and recommend the most effective alternatives.

The lecture attendance system that was launched by Kawaguchi in [18] featured a newly developed approach known as continuous monitoring, and the student's attendance was noted automatically by the camera, which took a snapshot of each student as they were present in the classroom.

Since there are only two cameras installed on the classroom wall, the architecture of the system is rather straightforward. The first camera is a capturing camera, and its purpose is to take an image of a student while they are in class. The second camera is a sensor camera, and its purpose is to determine where a student is seated within the classroom. The camera that captures images will then take a picture of the student. The picture that is taken from a camera and compared to the photos and faces that have been captured in the database over a long period of time in order to perfect the attendance system.

An additional study that was suggested by [2] presented an approach for real-time computer vision use in an autonomous attendance management system. The system compared the face that was extracted from the photograph taken by the camera in the classroom with the faces that were already stored in the system. The camera, which was designed to be non-intrusive, was placed by the system.

In addition to that, this system made use of an algorithm for machine learning, which are typically employed in computer vision. Additionally, HAAR CLASSIFIERS were previously utilised to train the images that were captured by the camera. The face snap that is captured by

the camera will first be converted to grayscale, and then subtraction will be performed on the images. After that, the image will be uploaded to the server to be stored for subsequent processing.

N. Kar [19] debuted an automated attendance management system in 2012 that relied on face recognition technology and made use of the Principal Component Analysis. In order to put the system into action, you will need to make use of two libraries, namely OpenCV, which is a computer vision library, and FLTK (Light Tool Kit). Both of these libraries were helpful in the development process, particularly OpenCV's support algorithm[20] and FLTK[21], which was utilised to build the user interface. Request Matching and Adding New Fact to Database are two functions that are included in the system. After the frontal face has been extracted, the first phase of the Request Matching process is to open the camera and take a picture of the extracted face. The subsequent stage is to recognise the face using the training data, after which the extracted face will be projected onto the Principal Component Analysis. The final phase displays the image of the acquired face that is closest to the camera.

Aside from that, the process of adding a new face to the database involves taking a snapshot, then extracting the frontal face photos, and finally carrying out the Haar cascade Method in order to locate the Principal Component Analysis Algorithm. The very last thing that needs to be done is to save the information within the face XML file. The primary focus of the system is on the algorithm that will improve face detection from any photographs or videos that are acquired.

The author of [3] also suggested a method that uses face recognition to perform automatic attendance. The system that uses MATLAB and Principal Component Analysis (PCA) to extract facial features like the mouth and nose. The system [7] was created to address problems with the time-consuming attendance marking system. The experiment's findings demonstrate in this study that the system can identify faces in situations when there is a dark background or a difference in the way they are viewed.

A smart attendance marking system that integrates two different differencing methods, such as Principal Component Analysis and Artificial Neural Network, was proposed by Jyotshana Kanti [4]. The author's goal is to remedy the time-consuming and conventional attendance marking mechanism. The system uses Principal Component Analysis to extract data from the face database, find patterns, and collect photos. Artificial neural networks are used to resolve input data problems or learn from input data and predict values. The author used a back propagation method and combined it with mathematical operations to construct the system. Because of this, the author's research reveals that the system may be used to recognise in various environments.

The design comes after the way that Priyanka Thakare proposed in [22] utilising Eigenface and Principal Component Analysis. Installing a front-facing camera that can capture a student's complete face while they are in class is necessary. After the camera had taken a picture, the first step was to input the picture into the system. Sometimes the brightness or darkness of a picture captured by a camera require enhancements, such as grayscale conversion. In the following phase of this system, histogram normalisation, the contrast of the image is removed. When a

student is seated in the back row, it is obvious. If the camera is a high definition camera, the median filter is employed to eliminate noise from the image, however occasionally the noise is still present. Additionally, the author uses skin categorization, which turns every pixel black except for those that are adjacent to the skin.

[1] Samridhi Dev and Tushar Patnaik's 2020 study on a face-recognition student attendance system In this study, three alternative algorithms were used to evaluate the system, and the KNN method performed the best, with an accuracy rate of 99.27%. The system was evaluated under a variety of circumstances, including lighting, head motions, facial expressions, and students' proximity to the camera. Even when the image incorporates faces with beards and glasses and without them, the system lives up to expectations. suggested method demonstrated to be excellent at recognising faces with a two-year age difference.

[2] Kolipaka Preethi, Swathy Vodithala (2021) AUTOMATED SMART ATTENDANCE SYSTEM USING FACE RECOGNITION

The suggested approach uses many stages to record live attendance. Face Recognition in A, DataSet Creation and Training in B C. Face Recognition and Attendance Updating

[3] Sharma S, Karthikeyan Shanmugasundaram, and Sathees Kumar Ramasamy (2016) developed FAREC, a CNN-based efficient face recognition method utilising Dlib. The article used trained Convolutional Neural Network feature models, which contain all of the labels' features from face recognition systems. The

Test photos are verified using these models, and the labels and claims that the image represents the person have the highest probability value. FAREC produces 96% accuracy for FRGC and a false acceptance rate of 0.1% (1 in 100) after 20 iterations of converging learning rate from 0.01. Before the fifth epoch, the training losses are quickly and substantially reduced to zero. The learning rate convergence and accuracy of FAREC are shown in figures 9 and 10 below.

[4] Marko Arsenovic, Srdjan Sladojevic, Andras Anderla, and Darko Stefanovic (2017) developed FaceTime - Deep Learning Based Face Recognition Attendance System.

Using the suggested augmentation strategy and a small number of photos per employee, the model was trained. This resulted in the initial dataset being expanded and the overall accuracy being raised. It was possible to see how the light circumstances affected the recognition process by looking at the photographs that were saved in the database during the acquisition time. The door was open, and the majority of the inaccurately anticipated images were visible in the daylight. By adding gradient modification to the photos, this might be fixed. Only a few photos with noise from an unidentified source had predictions that were accurate. Applying time-interval automatic re-training of the embedding deep CNN together with the recently obtained images predicted by the model with the high accuracy rate could increase overall accuracy.

[5] Mayank Srivastava, Amit Kumar, Aditya Dixit, and Aman Kumar (2020) Real Time Attendance System Using Face Recognition Technique

In this study, 30 faces were used as a training set of seven people to test the system's accuracy. The Extract () function displays a sample binary image that was created using Paul-Viola's face-extracting framework detection approach. The findings indicate that as the face angle is increased, the camera's face detection and identification rate declines. The authors want to create a facial recognition-based attendance management system for colleges that includes admission and leave times. The technology continuously monitors the entry and exit points to record each student's attendance. Results from our initial experiment outperformed conventional black-and-white display systems in performance evaluation. This technique is primarily designed to recognise faces in still photos or video frames.

PROPOSED SYSTEM

The purpose of the proposed system is to take a photo of each individual student's face and store it in a database together with their attendance information. It is necessary to take a picture of the student's face in such a way that each of the features on their face may be identified in the snapshot. Because the system records a video, the face can be recognised, and the attendance database can be kept up-to-date without the need for the instructor to manually take attendance in the classroom. The system captures a video.
This system is being developed with the help of Python and its libraries, such as PpenCV, dlib, face recognition, and more.

OpenCV: which stands for Open Source Computer Vision Library, is a function library for programming that primarily concentrates on real-time computer vision. It was initially conceived and developed by Intel and later received support from Willow Garage and Itseez (both of which were subsequently acquired by Intel). The library is available on multiple operating systems and can be used without cost thanks to the open-source BSD licence. Garbage collection and dynamic typing are both features of the Python programming language. It is compatible with many different programming paradigms, including object-oriented programming, functional programming, and structured programming, particularly procedural programming. Because Python comes with such a comprehensive standard library, the phrase "batteries included" is frequently used to describe this programming language programming that is both object-oriented and functional. Because it has such a large standard library, Python is frequently referred to as a "batteries included" programming language.

Dlib: Dlib is a cutting-edge C++ toolkit that incorporates techniques and tools for machine learning. These can be used to construct complex software that can solve problems that actually occur in the real world. It is utilised in a wide number of disciplines, including as robots, embedded technology, mobile phones, and massive high performance computer environments, both in the business world and in academic institutions.

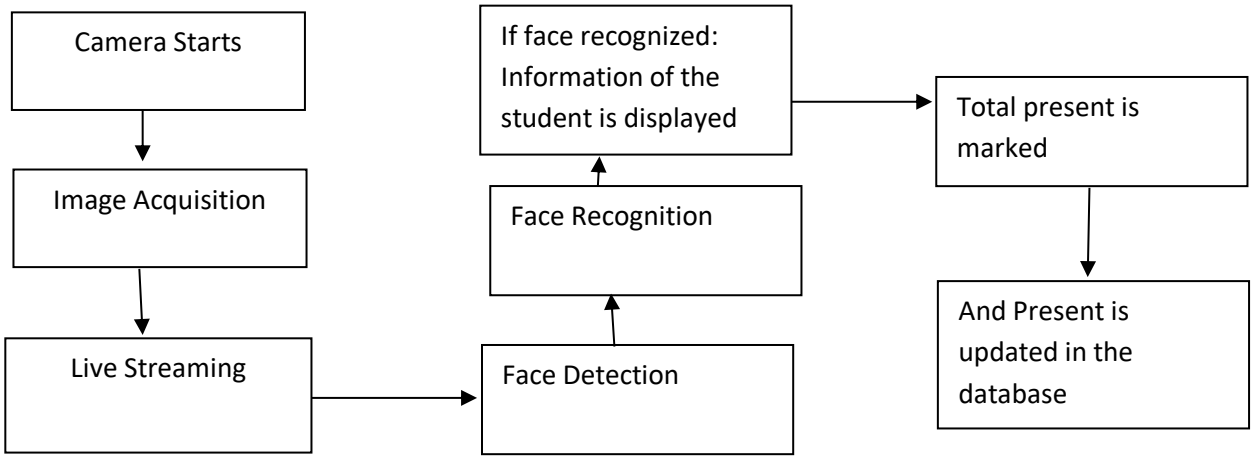
Face Recognition: Using the Python programming language or the command line, you may use the simplest face recognition library in the world to recognise and alter faces.

Built utilising the cutting-edge facial recognition capabilities of dlib, which were achieved through the use of deep learning. The model achieves an accuracy of 99.38% when compared against the benchmark of Labelled Faces in the Wild.

This also includes a straightforward command-line utility called face_recognition that allows you to do face recognition on a picture folder directly from the command line.

All the student of the class must have their images in the dataset . The images in the dataset will be analysed and compared with the faces that have been detected. In the event that a match is discovered, attendance will be recorded for the appropriate student.

The following description provides an overview of the proposed system's architecture:



These procedures can be broken down into four distinct parts:

- Dataset Creation:
  The faculty members individually generate each record in the dataset, which contains information such as the student's name, cumulative attendance up to this point, department, unique student identifier, year, and year of admission. In this proposed system, the use of Firebase database is included.

- Face Detection:
  The second phase is to detect the user's face. This occurs when a camera displays a green colour square form on the user's face. After this, the body is removed, and loading then begins.

- Face Recognition:
  This stage is to recognise the faces from the database using the faces that were uploaded. A face can be recognised by comparing it to images that have been saved in the database.

- Attendance Updation:
  Following the completion of the face recognition procedure, the recognised faces will be flagged as present. The cumulative sum of attendees up until this point is displayed, and each individual can only sign in and out once per twenty-four hours.

ALGORITHM

1. Import the required libraries:
   - `os`: for operating system-related operations
   - `pickle`: for serializing and deserializing Python objects
   - `numpy`: for numerical operations
   - `cv2`: OpenCV library for computer vision tasks
   - `face_recognition`: for face detection and recognition
   - `cvzone`: a utility library for OpenCV
   - `firebase_admin`: Firebase Admin SDK for interacting with Firebase services
   - `datetime`: for working with dates and times
2. Set up Firebase Admin SDK:
   - Initialize the Firebase app using service account credentials.
   - Specify the database URL and storage bucket.
3. Set up video capture:
   - Create a `VideoCapture` object to access the camera.
   - Set the capture properties for the video frame width and height.
4. Load the background image and mode images:
   - Read the background image from file.
   - Load the mode images into a list by reading them from a specified folder path.
5. Load the encoding file:
   - Open the encoded face data file using `pickle`.

- Deserialize the data and assign the encoding list and student IDs.

6. Initialize variables for mode type, counter, student ID, and student image.

7. Start an infinite loop for video frame processing:

- Read a frame from the video capture.

- Resize the frame for faster face recognition processing.

- Convert the frame to RGB color space.

8. Update the background image with the current frame and mode image.

9. Perform face detection and recognition on the resized frame:

- Use `face_recognition` library to locate faces in the frame.

- Compute face encodings for the detected faces.

10. Iterate over each detected face and compare it with known encodings:

- Compare the face encodings with the known encoding list using `face_recognition.compare_faces`.

- Calculate the face distance between the current face and known encodings using `face_recognition.face_distance`.

- Find the index of the best matching face based on the minimum face distance using `np.argmin`.

11. If a matching face is found:

- Update the background image with a bounding box around the detected face.

- Retrieve the student information from the Firebase Realtime Database based on the match index.

- Fetch the student image from the Firebase Storage using the student ID.

- Update the attendance data if the time elapsed since the last attendance is more than 30 seconds.

- Update the mode type and counter accordingly.

12. If the counter is not zero (i.e., a student is recognized):

- Check the counter value to determine the current mode type.

- Update the background image with the corresponding mode image.

13. If the counter is within a specific range:

- Update the background image with student information and image.

- Draw text on the image with attendance count, major, ID, standing, year, and starting year.

- Adjust the position of the student's name based on its length.

- Display the student's image within a specified region.

14. Increment the counter and check if it exceeds a threshold:

- If the counter reaches the threshold, reset the counter, mode type, and clear student information and image.

15. If no face is detected:

- Set the mode type and counter to zero.

16. Show the updated background image with overlays.

17. Wait for a key press and continue to the next iteration.

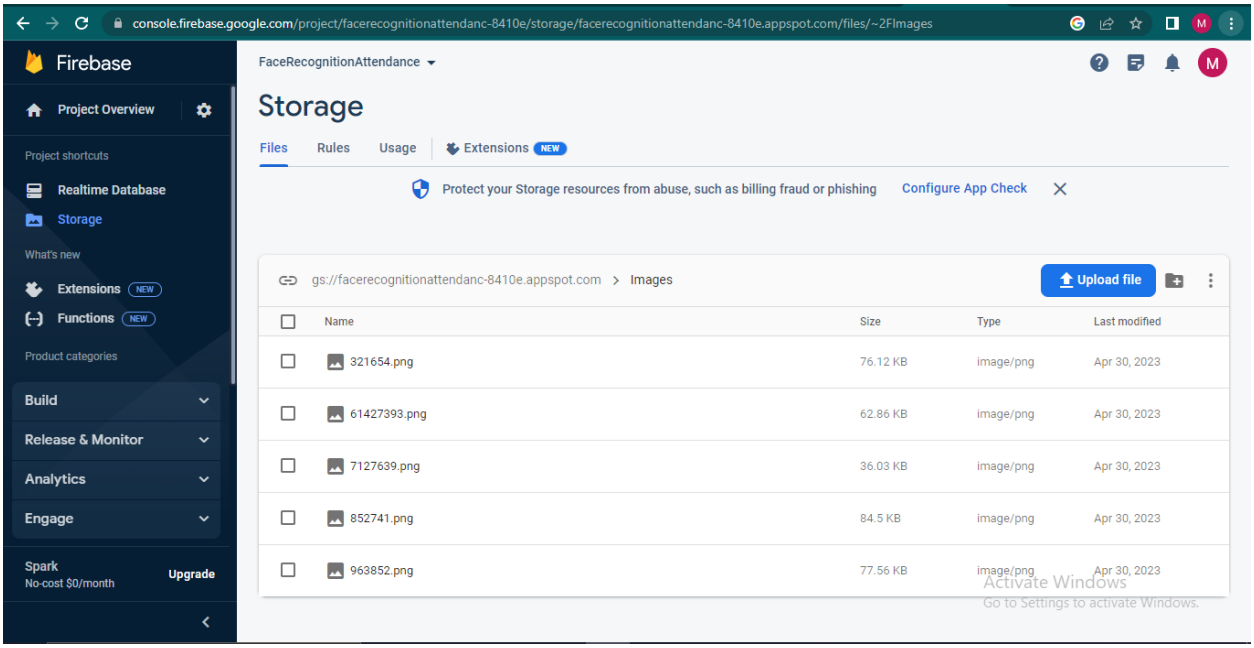
RESULT

The face acknowledgment attendance management system is incredibly user-friendly and operates effectively under short time constraints.

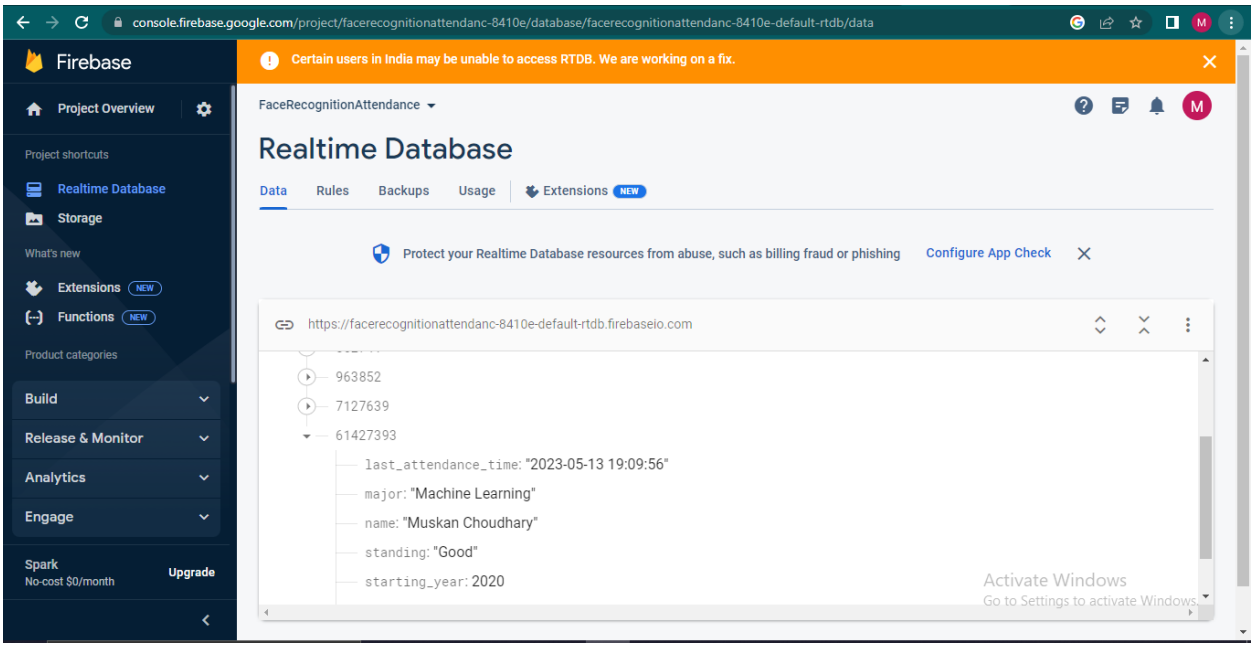
Since this is an automated system, if an administrator generated a student profile in the database just once, it would automatically use that profile as many times as necessary during the face detection and identification process.

- Firstly a database is created where the student’s information and their images with their id is stored. We have used two parts in database:

1- Storage: where all the images of the student is stored



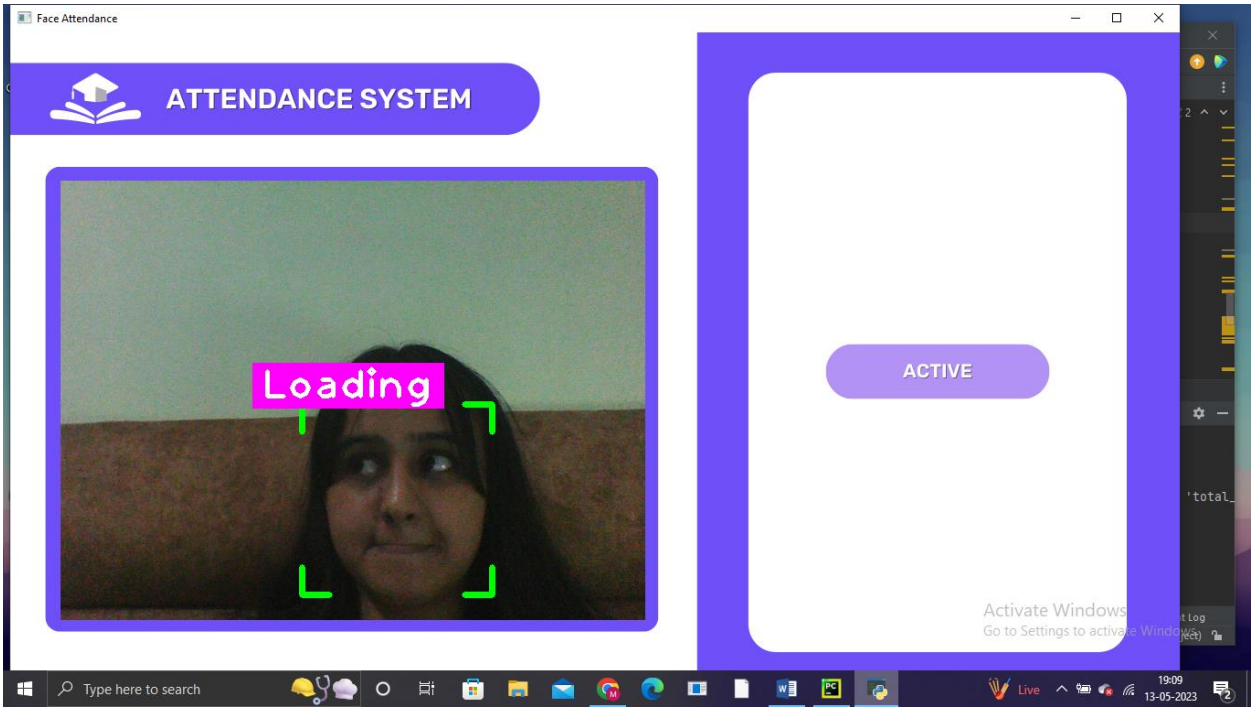
2- Realtime Database: where all the information such as students name, id, year, etc is stored.



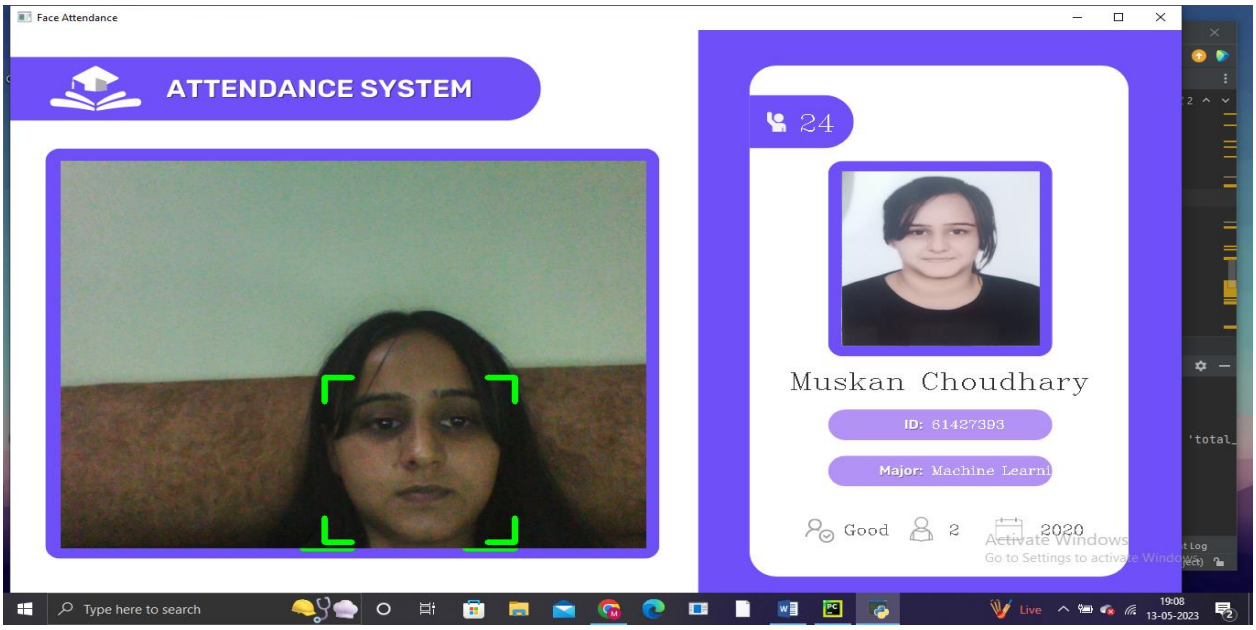
2-After creation of database the created Api is used to capture the image of the student,

After that it follows some series of steps:

- First step is where status of the student is showed as active.



- The second step where after detecting the image and recognizing it with the help of images stored in database information of the student is shown

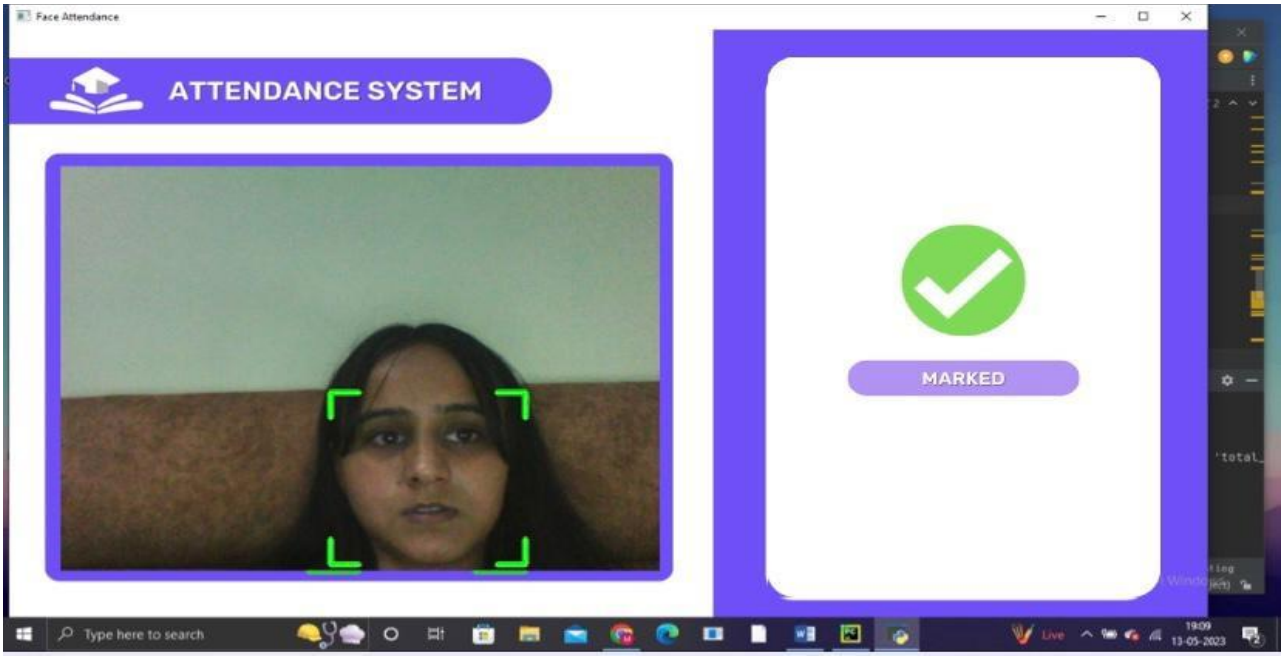


Information that is showed in this section is:

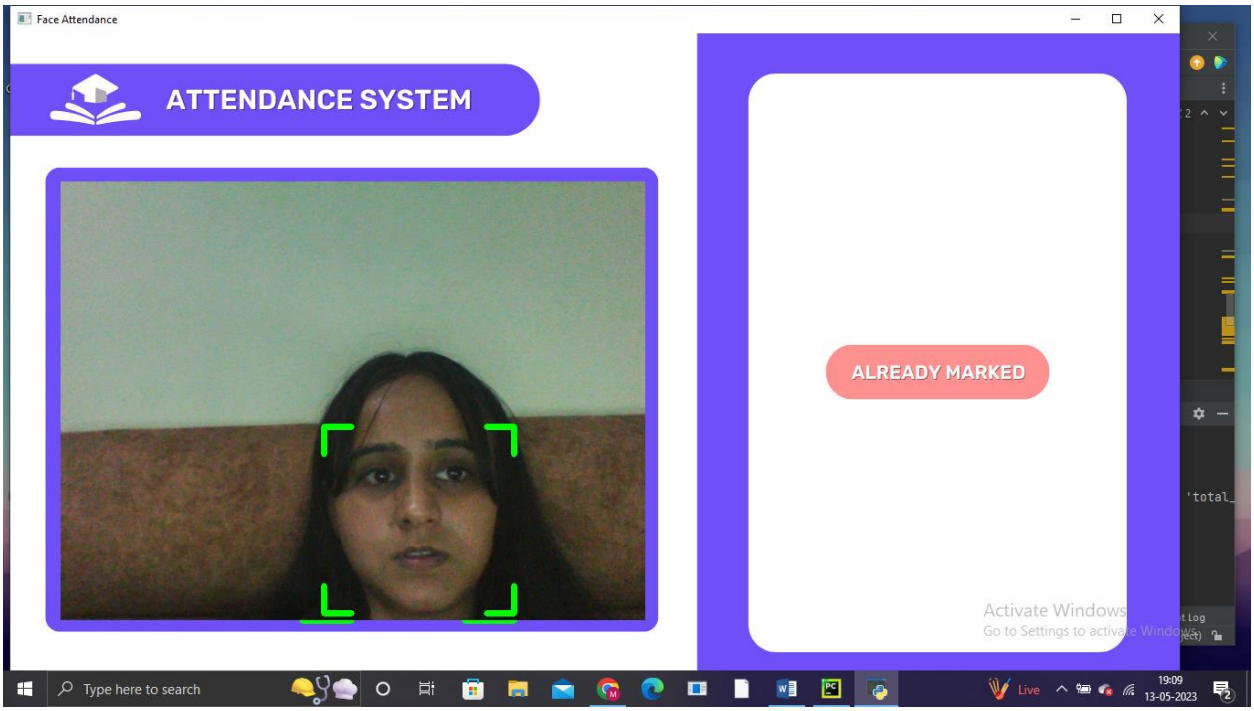
1- Total attendance
2- Name of the student
3- ID
4- Major
5- Students overall peformace

6- Year
7- Admission year

- Third step is where it is showed that whether the student attendance is marked or not



- If the student tries to get marked more than once in 24 hours than already marked will be showed



CONCLUSION

In this study, we suggested a model based on python libraries such as OpenCV, dlib, numpy, face recognition etc, for automatically taking attendance of the students for faculties. The suggested version improves on earlier methods and has been shown to be effective for both institutes and faculties . This means that it provide accuracy and reliability which helps faculties to take attendance easily and without wasting time and efforts to maintain those attendance on a paper sheet. Systems for face recognition attendance are highly scalable to handle large organisations with numerous locations. It enables centralised attendance management and gives pertinent stakeholders access to real-time data. Finally, compared to traditional attendance systems, the facial recognition attendance system has a lot to offer. It analyses facial features to swiftly and correctly record attendance using cutting-edge computer vision technology. By offering precise, effective, and secure solutions, face recognition attendance systems revolutionise attendance management. In businesses of all sizes, they improve convenience, lessen administrative hassles, and boost general productivity.

REFERENCES

[1] Jenif D Souza, Jothi S, Chandrasekar A, “Automated Attendance Marking and Management System by Facialrecognition using Histogram” in ICACCS 2019.

[2] Nandhini R, Duraimurugan N, S.P Chollalingam “ FaceRecognition Attendance System” in IJEAT in 2019.

[3]Omar Abdul Rhman Salim, Rashidha Olanrewaju , Wasiu Balogun “Class attendance management system usingface recognition” in ICCCE 2018.

[4]] Aruna katara , Sudesh V, Amar P, Nikhil D, Bhele

[5] Shreyak Sawhney, Karan Kacker, Samayak Jain, Shailendra Narayan ,Rakesh Garg “Real Time Smart Attendance system using face recognition techniques “ ininternational conference on cloud computing data scienceand engineering 2019.

[6] Shireesha Chintalapati, M.V. Raghunadh, "Automated Attendance Management System Based On Face Recognition Algorithms", IEEE International Conference on Computational Intelligence and Computing Research, 2013.

[7] Prajakta Lad, Sonali More, Simran Parkhe, Priyanka Nikam, Dipalee Chaudhari, " Student Attendance System Using Iris Detection", IJARIIE-ISSN(O)-2395-4396, Vol-3 Issue-2 2017.

[8] Dan Wang, Rong Fu, Zuying Luo, "Classroom Attendance Auto-management Based on Deep Learning",Advances in Social Science, Education and Humanities Research, volume 123,ICESAME 2017.

[9] B Prabhavathi, V Tanuja, V Madhu Viswanatham and M Rajashekhara Babu, "A smart technique for attendance system to recognize faces through parallelism", IOP Conf. Series: Materials Science and Engineering 263, 2017.

[10] Prajakta Lad, Sonali More, Simran Parkhe, Priyanka Nikam, Dipalee Chaudhari, " Student Attendance System Using Iris Detection", IJARIIE-ISSN(O)-2395-4396, Vol-3 Issue-2 2017

[11] D Midhunchakkaravarthy (2021), Blockchain embedded congestion control model for improving packet delivery rate in ad hoc networks, Machine intelligence and soft computing, 2021.

[12] . AP Gopi (2021), Secure Communication in Internet of Things Based on Packet Analysis, Machine Intelligence and Soft Computing, 2021.

[13] Automatic Attendance system through face recognition using machine learning bySumeet Kewalramani