Real Time Face Attendance using Open Computer Vision

Neeraj Panwar[1], Akash Nautiyal[2]

[1]*Associate Professor, Department of Computer Application*

[2]*Student, Department of Computer Application*

[1,2] *Graphic Era Hill University, Dehradun, Uttarakhand, India*

[1]neeraj.pan28@gmail.com

[2]anautiyal92@gmail.com

***Abstract*— A face recognition attendance system is a biometric technology that uses artificial intelligence to identify and verify people based on their facial characteristics. It's a fast, high-accuracy system that's used in various sectors, including finance, retail, government, and industry.**

Attendance systems have evolved significantly over the years, with the advent of technology providing innovative solutions to streamline the process. Facial recognition has emerged as a promising technology in this domain due to its accuracy, efficiency, and non-intrusiveness. The system operates by capturing facial images using a camera and employing Machine learning algorithms to detect, extract, and analyse facial features. Utilizing Machine Learning Algorithm, we develop Face encoding of the captured images and then compare it with encoding present in database of the stored images. The system's architecture encompasses both hardware and software components, including camera modules, processing units, and a centralized database.

The proposed system has been evaluated through extensive testing and validation, demonstrating promising results in terms of accuracy, speed, and reliability. Furthermore, the system's adaptability and scalability make it suitable for various industries, including education, corporate environments, and public institutions.

***Keywords* — Face recognition, face detection, Face Encoding, Computer vision, Firebase Database and Recognition mechanisms**

I. INTRODUCTION

Facial recognition technology has become a common biometric feature in laptops, cell phones, attendance tracking systems, and other devices in recent years. Many organizations employ facial recognition systems for surveillance, home security automation, security access, and criminal identification procedures because of recent technological breakthroughs. In essence, face recognition software compares an individual's geometric traits with their facial features. The conventional approach stores user images in a database, which is then used to compare collected photographs with user images. A variety of traits were retrieved and applied as facial features, including corners, edges, and texture descriptors. For categorization, these features were used with machine learning methods.

The system utilizes computer vision, a field of artificial intelligence, to automate the attendance process by recognizing and verifying individuals based on their facial features in real-time [1]. The primary components and functionalities of the system include:

Face Detection: Employing robust algorithms to identify and locate human faces within images or video streams. This initial step is crucial for subsequent processing.

Encoding Generation: Utilizing computer vision techniques to extract unique facial features, such as the distance between eyes, shape of the nose, and contours of the face. These features are translated into mathematical representations for recognition which is know as Encoding of an Image which is useful in comparing the images.

Real-Time Processing: The system operates in real-time, enabling immediate recognition and verification of individuals as they enter or interact with the system. This ensures swift attendance recording without causing delays or disruptions.

Database Management: Storing and managing a database of known faces or templates against which incoming faces are compared for identification and attendance tracking.

User Interface and Integration: Designing a user-friendly interface for administrators and users to interact with the system seamlessly. Integration with existing attendance management systems or APIs enhances its usability and compatibility.

Accuracy and Performance Evaluation: Conducting rigorous testing and evaluation to assess the system's accuracy, reliability, and performance under various conditions, including varying lighting, angles, and facial expressions.

II. LITERATURE REVIEW

Automated facial recognition systems are commonly employed in corporations, educational institutions, and government agencies to monitor employee attendance.

Compared to manual methods, it is quicker, more efficient, and completely removes the need for physical labour [2].
The objectives of a real-time face attendance system using computer vision typically revolve around improving efficiency, accuracy, and convenience in attendance management processes. Here are the primary objectives:

Automated Attendance Tracking: To automate the attendance recording process by using facial recognition technology. This eliminates the need for manual attendance marking, reducing errors and time spent on traditional methods.

Real-Time Monitoring: To provide instantaneous attendance updates. As individuals are recognized in real-time, the system can immediately log their attendance status without delays.

Accuracy and Reliability: To ensure a high level of accuracy in identifying individuals. The system aims to minimize false positives or negatives, accurately associating individuals with their attendance records.

Efficiency and Timesaving: To streamline attendance management for both administrators and attendees. By eliminating manual processes, the system saves time and effort, allowing staff to focus on other essential tasks.

Integration and Compatibility: To integrate seamlessly with existing attendance management systems or databases. Compatibility ensures smooth implementation without disrupting established workflows.

User-Friendly Interface: To offer an intuitive and user-friendly interface for both administrators and users interacting with the system. This promotes ease of use and acceptance among users.

Security and Access Control: To enhance security by monitoring entry points and ensuring that only authorized individuals are granted access based on their recognition in the system.

Scalability and Adaptability: To design a system that can be easily scaled to accommodate varying numbers of users and locations. This adaptability allows the system to be implemented across different environments.

Compliance and Ethical Considerations: To adhere to ethical standards and legal requirements concerning data privacy, consent, and the responsible use of facial recognition technology.

III. Data collection & preprocessing

Data preprocessing is a crucial step in data analysis and machine learning that involves transforming raw data into a more usable format. It aims to clean, normalize, and prepare the data to enhance its quality and make it suitable for further analysis or model training. It also involves storing of pre-processed data in database so that it can be used further.

The data of an individual is collected in the form of Name (Name of the student), Major (in which subject user is doing Course), Standing (Garde of User), Starting Year (Starting year of the Course), Year (Course Duration). Last Attendance and Total Attendance is added by Code. All these variables are bound up in a folder named Roll Number of User and saved in the Firebase database.

The image which is taken from the webcam is of high pixel range and it can be a hectic task for our computer to generate encoding of that image and to store that high pixel image in our database can also be wastage of storage. The image captured from the webcam has extra information, which is not necessary for the face recognition algorithm, so to remove that extra information we first convert the high pixel image into 256 * 256 and pass the image using various filter for the optimization of the image so that the image used for encoding can have more information regarding facial features. When the image is read from open CV, its default format is BGR and our model uses RGB format, so with the help of Open CV we convert BGR into RGB. After that the image is named after the Roll Number of the user.

IV. Methodology

The methods used in this research were trained and tested with PyCharm with some basic libraries installed. These algorithms are trained with a normal CPU, so there is no necessity for high-power GPUs.

For storing data, Firebase Database is used which update information in Real-Time. This database can store user information in Real time and updating it on real time. Data is stored in JSON format in Firebase Realtime Database, a cloud-hosted NoSQL database. With its real-time synchronization feature, clients can instantly sync data changes across one another in milliseconds. The database is accessible from various platforms, including web, iOS, Android, and server environments. Changes made by one client are instantly reflected on all connected devices, making it ideal for collaborative or multi-user applications. Firebase also provides offline capabilities, enabling applications to remain functional even without an internet connection. Once reconnected, the database syncs any pending changes automatically update. Firebase offers robust security rules to control access to the database. Developers can define rules based on authentication, user roles, and data structures to ensure data security and privacy. It seamlessly integrates with other Firebase services such as Authentication, Cloud Functions, Analytics, and Storage, allowing developers to create comprehensive and scalable applications. Firebase Realtime Database is designed to handle scalability requirements, automatically scaling to accommodate increasing data and user loads [3]. The workflow of system is shown in Fig.1.

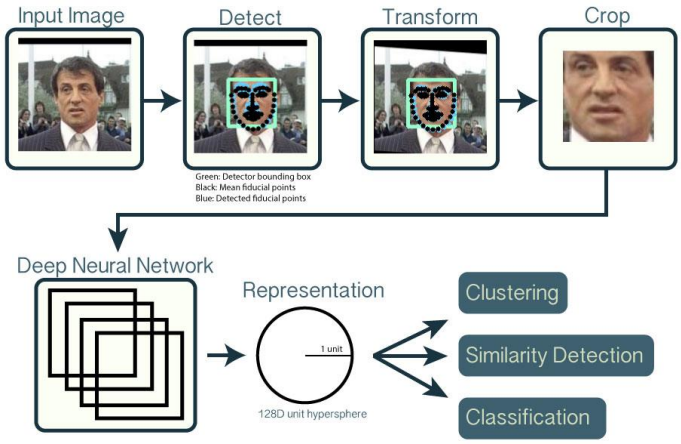

Fig.1 Workflow of System

V. PROPOSED SYSTEM

The proposed workflow for Real-Time Face Attendance System encompasses: -

A. Data Preprocessing

Data preprocessing is a crucial step in data analysis and machine learning that involves transforming raw data into a more usable format. It aims to clean, normalize, and prepare the data to enhance its quality and make it suitable for further analysis or model training. It also involves storing of pre-processed data in database so that it can be used further. Individual data is gathered in the following formats: Name (student name), Major (subject area in which the user is enrolled in the course), Standing (user's grade), Beginning Year (course year beginning), and Year (length of the course). Code adds the Last Attendance and Total Attendance. All these variables are stored in the Firebase database and organized into a folder called Roll Number of User which is basically a 6-digit number.

The image captured by the webcam has a high pixel range, which means that encoding it and storing it in our database could be a laborious process for our computer and result in storage waste. The webcam image contains extra information that the face recognition algorithm does not need. To get rid of this extra information, we first convert the high-pixel image to 256 by 256 pixel and then optimize the image using different filters. This way, the image used for encoding can contain more information about facial features. Since our model uses the RGB format and the default BGR format for images read from OpenCV, we use OpenCV to help us convert BGR to RGB. The image is then given a name based on the user's roll number.

B. Face Detection

Face detection is a computer vision technology used to identify and locate human faces within images or video frames. It's a foundational step in various applications, including facial recognition, security systems, photo editing software, and more. The technique of identifying a face in a picture is called detection. Facial recognition, made possible by computer vision, can recognize, and identify individual faces from a picture that contains the faces of one or more people [4]. A face detection example is shown in Fig2.

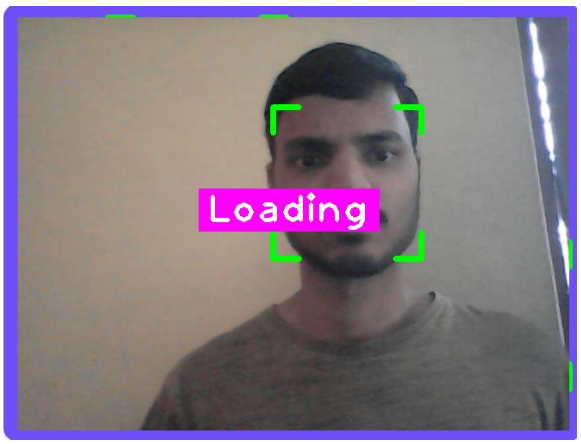

Fig 2. Face Detection using face recognition.

C. Generate Face Encodings

Generating face encodings involves converting facial images into numerical representations, enabling easy comparison and recognition of faces. One popular method for face encoding is using deep learning models, particularly face recognition models, to extract facial embeddings [5]. Dlib comes with pre-trained landmark detector that detects 68 different landmark features as shown in Fig.3.

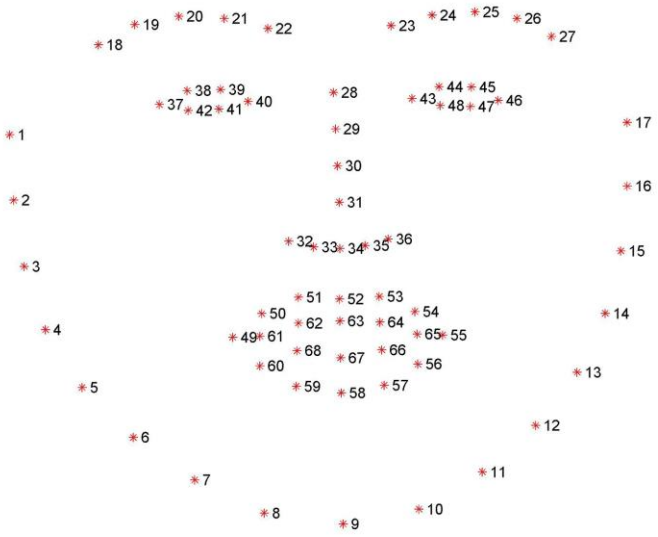

Fig 3. Dlib facial landmark detector

A CNN that has been previously trained is used to generate the 128 encoding values. Due to its prior training on millions of training photos, the model is able to produce trustworthy encodings for faces it has never seen before. Images of the same individual ought to yield around the same 128-valued

encoded vector. An extremely basic interface for extracting the encodings from an image and the set of rectangles that depict the faces in the image is offered by the facial_recognition library. Finding out which person the face belongs to is as easy as measuring the distance between the new facial image and those in the encodings and choosing the one with the lowest distance value. This is done once you have the 128 valued encoded vector and the person's name as the label [6]. Below you can see the face encoding (Fig.5 and Fig.7) of two different images Fig.4 and Fig.6 taken from the webcam.

Fig 4. Input Image 1 from camera Model.

Encoding Started ...

array([-1.90755725e-01, 1.27633125e-01, 2.92013586e-02, -3.49817239e-02,
-1.23655237e-02, -9.81059074e-02, -2.48256847e-02, -1.01783261e-01,
1.08596422e-01, -3.11100110e-02, 2.03311846e-01, -1.00102231e-01,
-2.21952349e-01, -1.36154726e-01, -2.54909489e-02, 9.77982506e-02,
-1.09247833e-01, -1.99613065e-01, 1.42976791e-02, -1.01944096e-01,
1.17541626e-01, 5.11706993e-03, 1.54069178e-02, 1.08624332e-01,
-2.86817849e-01, -2.77195096e-01, -9.48177874e-02, -2.17730090e-01,
3.68094258e-03, -1.19397134e-01, -2.11995188e-02, 4.86608967e-02,
7.48404413e-02, 3.25713865e-02, 7.43889362e-02, -4.79312614e-02,
9.41099226e-02, -2.00933248e-01, 1.03238538e-01, 1.53080150e-01,
1.45939484e-01, 5.49700409e-02, 1.52193904e-01, -1.54745445e-01,
3.05712204e-02, 5.61157838e-02, -2.23978162e-01, 1.62887692e-01,
6.50427639e-02, 1.80913117e-02, -6.47907481e-02, -1.45804305e-02,
2.25748748e-01, 7.31183663e-02, -1.49839759e-01, -8.68962184e-02,
1.26961440e-01, -1.57963574e-01, -4.08169851e-02, 1.21845007e-01,
-8.39499384e-02, -1.63634926e-01, -2.68825203e-01, 7.65522718e-02,
-1.59626573e-01, 1.03154860e-01, -1.08275175e-01, -2.94140391e-02,
9.88917053e-02, -2.83227175e-01, 1.47268504e-01, 1.50552467e-01,
-8.42809603e-02, 8.90252069e-02, 7.27110654e-02, 1.72209013e-02,
2.99694594e-02, 1.84677541e-04, -9.40349102e-02, -9.36298594e-02,
6.18245490e-02, -7.97352046e-02, 5.16808107e-02, 5.50850760e-03)]

Encoding Complete
File Saved
Process finished with exit code 0

Fig 5. Encoding of Fig1

Fig 6. Input Image 2 from camera Model.

Encoding Started ...

array([-2.20260203e-01, 1.30789548e-01, 7.59498700e-02, -4.84002829e-02,
-3.05678025e-02, -3.31506915e-02, -5.93391657e-02, -1.11219533e-01,
1.63894996e-01, -7.84125775e-02, 2.31201515e-01, 2.20352970e-02,
-9.91328359e-02, -7.81775936e-02, -6.82642609e-02, 1.59182370e-01,
-1.73202962e-01, -1.30004942e-01, -5.82705997e-02, -1.26322702e-01,
-9.83849168e-05, 2.29360834e-02, -1.96711738e-02, 5.66161275e-02,
-8.76095295e-02, -3.29574496e-01, -9.54316184e-02, -1.23355776e-01,
6.16217889e-02, -6.14502057e-02, 5.96307330e-02, 3.92164253e-02,
-1.53687090e-01, -2.27859914e-02, 6.19656444e-02, 7.00810701e-02,
3.96575592e-02, 4.29849550e-02, 1.62563711e-01, 3.10022812e-02,
3.73470504e-03, 9.08966362e-02, -1.83375955e-01, 1.45096555e-01,
1.32592246e-01, 1.29765749e-01, ...3731e-01, 1.41869143e-01,
-6.02923557e-02, -6.05847128e-03, -3.53222303e-02, 3.23826298e-02,
9.01911873e-03, 1.09084390e-01, 9.74849835e-02, 4.25669998e-02,
1.69138983e-02, -2.45250817e-02, -9.32487175e-02, -1.09036803e-01,
1.60095356e-02, -6.40238002e-02, 1.25130966e-01, 9.43907164e-03

Encoding Complete

File Saved

Process finished with exit code 0

Fig 7. Encoding of Fig3

D. Compare Encodings

In this process we generally compare both encoding of the database images and the captured images and the matched result is stored in list. The least value in the list is then identified by ID and is shown in the desired Graphics. We can see the least value in the array, and that value is then converted into Boolean as minimum value gets true and rest gets false [7]. After we get the index value of matched index, we update attendance in Database with last attendance time as shown in Fig 8.

```
Loading Encode File ...
Encode File Loaded
matches [False, False, True, False, False, False]
faceDis [0.99285519 0.65350092 0.34120175 0.66463729 1.05193736 0.89501874]
Match Index 2
Known Face Detected
250999
{'last_attendance_time': '2024-04-11 12:19:41', 'major': 'Computer', 'name': 'Akash Nautiyal'
55.754506
matches [False, False, True, False, False, False]
faceDis [0.99040352 0.66188263 0.30666434 0.67987125 1.04448264 0.88833328]
Match Index 2
Known Face Detected
250999
```

Fig 8. Comparison of Encoding

E. Update Attendance in Database

After showing desired image in the graphics, the attendance of the identified ID is then updated in the database in real time (Fig. 9). If the attendance is already marked so it doesn't affect the database and the result "Already marked" is shown in the Output, otherwise the attendance is increased by 1 and last attendance time is also updated.

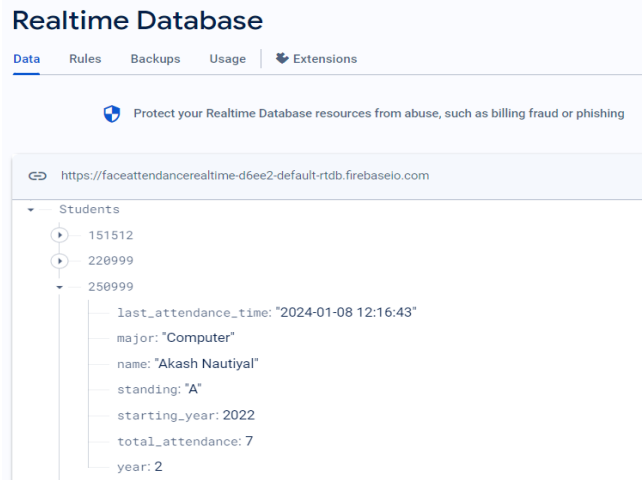

Fig 9. User Information in the database

VI. Result

The world's most straightforward face recognition library allows you to recognize and work with faces from Python or the command line developed with dlib's cutting-edge deep learning-based face recognition technology. On the Labeled Faces in the Wild benchmark, the model's accuracy is 99.38%. A database of face photos called Labeled Faces in the Wild was created with the purpose of researching the issue of unrestrained face recognition. More than 13,000 facial photos that were gathered from the internet are included in the data collection. The name of each person in the photo is written on each face.

The image provided to face_encoder has its (128,1) dimension encoding obtained using face_encoder. It is a 29-convolution-layer pretrained ResNet network model. A dataset containing roughly 3 million faces is used to train the model. This system performs well in each and every aspect and perform with same accuracy for every user whenever it is used for face attendance. As it uses Face Recognition library whose accuracy is 99.38% so because of that this system is able to recognize face with the same accuracy and gives positive result every time. As we are using Google Firebase as a database and it is known for its smooth operation or updating data in real time so as face is recognized by model its attendance is marked in no time. We have tried this system with the information of 10 people and it has marked attendance of them every time.

VII. Conclusions

Applications for facial image processing include face recognition systems, and in recent years, their importance as a research topic has grown. The system's implementations include person verification, video monitoring, crime prevention, and other related security measures. University implementations of face recognition systems are possible. The goal of the Face Recognition Based Attendance System is to decrease the errors that happen with the manual, traditional method of recording attendance. Automating and creating a system that benefits a company, like an institute, is the goal. This approach is workable, dependable, and sufficiently safe. The suggested method can identify several faces, and the system's performance yields respectable, decent outcomes.

References

[1] Modern Face Recognition with Deep Learning [Online] Avaliable: Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning | by Adam Geitgey | Medium

[2] Empowering Students and Teachers: The Role of Face Recognition Attendance System in Educational Institutions[Online] Available: https://www.rydotinfotech.com/blog/benefits-of-face-recognition-attendance-system-for-schools-and-colleges/

[3] Rosebrock, Adrian. "*Opencv face recognition*." PyImageSearch. Dostopno na: https://www. pyimagesearch. com/2018/09/24/opencv-face-recognition/[14. 4. 2019] (2018).

[4] Facial Recognition — a visual step by step [Online] Avaliable: https://medium.com/swlh/facial-recognition-a-visual-step-by-step-d679289bab11

[5] [Hangaragi, Shivalila, Tripty Singh, and N. Neelima. "*Face detection and Recognition using Face Mesh and deep neural network*." Procedia Computer Science 218 (2023): 741-749.]

[6] Step by Step Face Recognition Code Implementation From Scratch In Python [Online]Available:https://towardsdatascience.com/step-by-step-face-recognition-code-implementation-from-scratch-in-python-cc95fa041120

[7] How to compare two images and get an accuracy level with Python [Online]Available:https://medium.com/@gowtham180502/how-to-compare-two-images-and-get-an-accuracy-level-with-python-61d8d953a942