

FACE RECOGNITION-BASED ATTENDANCE SYSTEM

Sunitha H D¹, *Kaushik Kumar^{2*}, Kushagra Maheshwari³

¹Department of Electronic & Communication Engineering, R R Institute Of Technology, Karnataka, India.

²Department of Electronic & Communication Engineering, R R Institute Of Technology, Karnataka, India.

³Department of Electronic & Communication Engineering, R R Institute Of Technology, Karnataka, India.

*Corresponding Author Email: Kaushikjnv2004@gmail.com

Abstract

This paper presents the design and implementation of a real-time face recognition-based attendance system for automated identity verification and attendance logging using computer vision and machine learning. The system captures live video through a webcam, detects faces in real time, and matches facial embeddings with a stored user database. It integrates image acquisition, preprocessing, face encoding, recognition, and attendance logging into a single workflow, eliminating manual registers and fingerprint-based methods. The system is implemented using Python, OpenCV, dlib, Flask, and the face_recognition library. Preprocessing techniques such as grayscale conversion, noise reduction, and CLAHE are used to improve recognition under varying lighting conditions. Once a valid match is found, the user's name, date, and timestamp are automatically stored in CSV format.

The proposed system is low-cost, contactless, scalable, and suit-able for educational institutions and workplaces. Experimental results show reliable performance under indoor conditions with support for multi-face detection and future expansion through cloud and IoT integration.

Keywords- Face Recognition, Attendance System, OpenCV, dlib, Flask, Computer Vision, Machine Learning

1. Introduction

Attendance management is a routine but important administrative task in schools, colleges, workplaces, and training centers. Although it appears simple, it consumes valuable time every day and is usually carried out through manual registers, signature sheets, RFID cards, barcode scans, or biometric fingerprint devices. Each of these methods has limitations. Manual marking is slow and depends on human supervision. Fingerprint scanners require physical contact and may fail due to moisture, dirt, or poor maintenance. RFID cards can be forgotten, shared, duplicated, or misused. Barcode systems are easy to exploit and do not reliably stop proxy attendance.

Face recognition offers a practical alternative because it verifies identity using natural facial features rather than physical tokens or direct touch. A person only needs to stand in front of a camera, and the system can detect, identify, and log attendance automatically. This makes the process faster, cleaner, and more convenient. It also

improves accountability because the actual face of the person must be present at the time of

marking. In institutional environments where attendance data is used for evaluation, discipline, payroll, or reporting, a more reliable and automated solution is clearly valuable.

Implementing a real-time face recognition-based attendance system that is affordable, easy to use, and capable of running on standard hardware. The project demonstrates how computer vision and machine learning can be combined to solve a practical problem in an efficient manner. It also provides a useful mini-project platform for understanding facial encodings, recognition pipelines, and web-based user interaction. The proposed approach emphasizes ease of deployment, modular design, and future scalability.

2. Related Work

Several recent studies have highlighted the rapid advancement of automated attendance systems using face recognition, deep learning, and real-time image processing techniques.

S. Mehta *et al.* [1] analyzed the use of Convolutional Neural Networks (CNNs) for automated attendance systems and reported high recognition accuracy under controlled environments. However, their study observed reduced performance under varying lighting conditions and facial pose changes, indicating the need for stronger preprocessing techniques.

Intel OpenVINO Team [2] discussed real-time face detection on low-power and edge devices. Their documentation emphasized the tradeoff between inference speed and model complexity, showing that lightweight optimized models are more suitable for practical deployment where latency is critical.

R. Sharma and P. Yadav [3] proposed a classroom attendance system using Haar Cascade face detection. Although computationally efficient, the system showed poor performance in crowded scenes, side-angle faces, and partial occlusions, highlighting the limitations of traditional approaches.

H. Li *et al.* [4] explored face-recognition-based identity authentication systems and noted that many existing solutions lacked protection against spoofing attacks. Their research recommended liveness detection and motion analysis to improve system security.

Singh and A. Kapoor [5] developed a smart attendance platform using Python and OpenCV for educational institutions. Their results showed that digital attendance systems reduce manual effort and simplify record management, but performance depended on image quality and camera placement.

J. Patel *et al.* [6] investigated deep metric learning for biometric recognition and concluded that embedding-based models outperform conventional image comparison methods in both speed and recognition accuracy.

Based on previous studies, the proposed system combines real-time face detection, image preprocessing, robust facial embeddings, and automatic attendance logging to improve practical usability, speed, and reliability.

3. Materials and Methods

The first stage of the methodology is data collection. Multiple images of each registered person are captured through a webcam or uploaded through the interface. These images are stored with labels that identify the person. Collecting more than one image is important because a single face image cannot represent the many variations that occur in real use, such as changes in pose, distance from the camera, or expression. A richer dataset improves recognition accuracy later. The proposed system is designed as a modular attendance platform with four main stages: face capture, preprocessing and encoding, recognition, and attendance logging.

During enrollment, the user registers by uploading one or more face images. These images are stored in a dataset and processed to generate face encodings. Multiple encodings per person can be kept so that recognition remains reliable even if the face is seen from different angles or under different lighting.

During runtime, a live camera stream is opened and analyzed frame by frame. The system detects visible faces, extracts the facial region, preprocesses the image, and generates a new encoding. That encoding is compared with the stored encodings using a similarity threshold. If a match is found, attendance is marked automatically and the record is saved with a timestamp. If no valid match is found, the face is treated as unknown and not logged.

The architecture is intentionally simple but effective. The processing logic is separated from the user interface, and the storage of encodings is separated from attendance without rewriting the whole project.

It converts the input image to grayscale because color information is not needed for the recognition pipeline and grayscale reduces computational load. CLAHE is then applied to improve contrast in local regions of the face, which is especially helpful under uneven lighting.

Noise reduction is used to remove unwanted artifacts produced by the camera, compression, or motion blur. These steps produce a cleaner and more stable input for feature extraction.

The third stage is face encoding. The face recognition library, which is built on top of dlib, extracts facial land-marks and converts the face into a 128-dimensional numerical embedding. This encoding represents the facial identity compactly and allows fast comparison. Rather than comparing raw images, which would be slow and unreliable, the system compares embeddings. This makes the

matching process efficient and suitable for real-time use.

The fourth stage is live recognition. As the camera stream continues, each frame is processed to detect faces. Every face is encoded and compared against stored encodings. If the distance between the live encoding and a stored encoding is below the threshold, the face is recognized.

The threshold is selected carefully so that it avoids false matches while still allowing small changes in facial appearance. Recognition occurs multiple times per second, making the system responsive and practical.

The final stage is attendance logging. Once the face is recognized, the system stores the person's name, date, and time in a CSV file. Before adding a new entry, the system checks whether the same person has already been marked for that day. This duplicate-checking step prevents repeated attendance entries. The log file can later be used for analysis, reporting, or conversion to other formats if necessary.

The workflow figure is placed here because it explains the system from registration to attendance marking. It gives a clear visual understanding of how the input moves through preprocessing, recognition, and logging.

In an IEEE paper, this type of figure belongs in the methodology section because it supports the description of the internal working.

3.1. System Architecture

The system architecture contains four layers: input, processing, storage, and interface. The input layer captures live video from the webcam or IP camera.

The processing layer performs face detection, preprocessing, encoding, and matching. The storage layer contains the registered facial encodings and the attendance log. The interface layer provides a web-based front end for registration and monitoring.

This layered design improves maintainability because each part has a clearly defined function. It also makes the system easier to scale and debug. If a future version needs cloud synchronization or additional authentication features, those changes can be integrated without disturbing the whole pipeline. The architecture is therefore suitable not only for the current mini-project but also for further development into a larger platform.

The use of a browser-based interface is especially helpful because it avoids complex desktop-only

interaction. Users can open the system in a browser, upload a face, trigger recognition, and review attendance records. This makes the project simple enough for day-to-day use and flexible enough for future expansion into a smarter attendance dashboard.

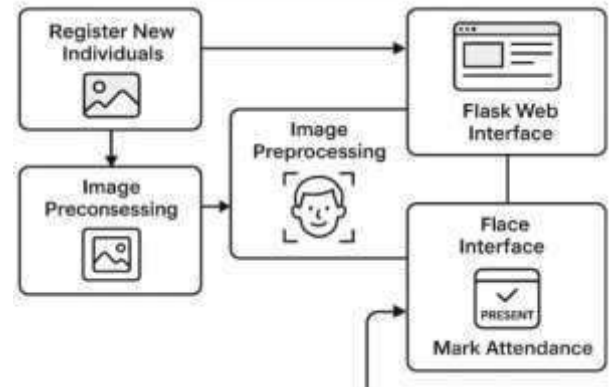


Fig. 1: Functional workflow of the face recognition attendance system.

3.2. Implementation Details

The implementation is developed in Python because Python has a strong ecosystem for image processing and machine learning. OpenCV handles frame capture, image manipulation, and display. dlib and face_recognition are used for facial detection and embedding generation. Flask provides the web interface that connects user actions to backend processing. Pickle is used to serialize and store facial encodings efficiently.

At registration time, the user submits a name and uploads a facial image. The system detects the face, preprocesses the image, and generates an encoding. That encoding is stored in a serialized file so that it can be loaded quickly during recognition. If the same user is registered with multiple images, the system stores multiple encodings, which improves robustness when the face is seen under different conditions.

At runtime, the webcam continuously captures frames. For each frame, the system detects faces and draws bounding boxes and labels around the detected regions. The encoding for each face is compared against the stored encodings. If a match is found, attendance is recorded automatically. If not, the face is labeled as unknown and ignored for logging purposes.

The attendance module stores the records in CSV format. This format is simple, transparent, and easy to open in spreadsheet software. The system also ensures that a person is not marked more than once on the same day. This prevents duplicates and keeps

the attendance record clean and trustworthy. The images stay aligned, readable, and professional. It also avoids the messy look that happens when screenshots are placed randomly inside the text.

4. Results and Discussion

The prototype was tested under normal indoor lighting using a standard webcam. The system detected registered faces and marked attendance automatically without manual intervention. Recognition was fast enough for live use and worked well when the face was clearly visible to the camera. Multiple faces could be detected in a single frame, which is useful in classroom and office environments. The prototype contains a web-based front page, a student dashboard, and the overall workflow diagram. The front page gives the system a clear entry point and shows how the application is intended to be used. The student dashboard presents timetable and attendance-related data in a structured visual format. These prototype images are not decorative; they explain the visible behavior of the system and show how the user interacts with the application.

The first prototype image belongs in the implementation section because it shows the homepage design of the attendance platform. The second image belongs here because it shows the student dashboard view, which is the main user-facing output after login or access. Together, these two images demonstrate that the system is not just an algorithmic prototype but also a usable application with a proper interface.

The results show that the system is reliable in controlled environments. When lighting is reasonable and the face is front-facing, recognition is accurate and immediate. Performance decreases somewhat under dim light or when the face is turned away from the camera, which is expected in any camera-based biometric system.



(a) Prototype homepage interface



(b) Student dashboard interface

Fig. 2: Prototype implementation of the face recognition attendance system.

TABLE I: Observed System Behavior

Condition	Observed Behavior	Result
Normal indoor light	Stable recognition	Good
Low light	Slight accuracy reduction	Acceptable
Multiple faces	Simultaneous detection	Good
Duplicate entry	Blocked correctly	Successful
CSV logging	File updated correctly	Successful

The preprocessing stage helps reduce these problems but cannot eliminate them entirely.

The main strength of the prototype is that it removes manual work while keeping the system simple. The user does not need to touch a device or carry a card. The attendance record is generated automatically and can be reviewed later. This is especially useful in classes, where attendance must be taken quickly at the start of a lecture. The system also produces a more transparent log because it records the exact timestamp of recognition.

Another advantage is simplicity. The browser-based interface makes the system easy to use even for non-technical users. The attendance data is stored digitally, which simplifies tracking, reporting, and analysis. The modular architecture also makes the system easy to improve in future versions.

Conclusion

This paper presented a real-time face recognition-based attendance system that automates attendance marking using computer vision and machine learning. The system captures live video,

detects faces, generates embeddings, matches them with stored identities, and logs attendance automatically. The design is contactless, affordable, and suitable for academic as well as workplace environments. The current prototype still has limitations. It depends on lighting quality and camera placement, so recognition can drop when the environment is too dark or the face is partially hidden. It does not yet include

TABLE I: Observed System Behavior

Condition	Observed Behavior	Result
Normal indoor light	Stable recognition	Good
Low light	Slight accuracy reduction	Acceptable
Multiple faces	Simultaneous detection	Good
Duplicate entry	Blocked correctly	Successful
CSV logging	File updated correctly	Successful

biometric system. It does not yet include anti-spoofing or liveness detection, which means a printed photo might still be a problem in some cases. The data is stored locally, so large-scale centralized monitoring is not yet available. The preprocessing stage helps reduce these problems but cannot eliminate them entirely.

The main strength of the prototype is that it removes manual work while keeping the system simple. The user does not need to touch a device or carry a card. The attendance record is generated automatically and can be reviewed later. This is especially useful in classes, where attendance must be taken quickly at the start of a lecture. The system also produces a more transparent log because it records the exact timestamp of recognition.

Another advantage is simplicity. The browser-based inter-face makes the system easy to use even for non-technical users. The attendance data is stored digitally, which simplifies tracking, reporting, and analysis. The modular architecture also makes the system easy to improve in future versions.

Future work should focus on improving these areas. Live-ness detection can be introduced using blink detection, motion analysis, or texture-based spoof checking. Cloud based dashboards can be used for centralized attendance storage. Mobile app integration can provide notifications and remote access. The system can also be ported to IoT hardware such as ESP32-CAM or Raspberry Pi for standalone deployment. More advanced recognition models such as FaceNet or Arc-Face can be explored to improve accuracy and robustness.

The developed prototype shows that face recognition can be used effectively for attendance

management when combined with preprocessing, compact facial encodings, and digital logging. The system performs reliably in normal indoor conditions and supports multi-face recognition in live video streams. Although the current version has limitations in lighting robustness and anti-spoofing, it provides a strong foundation for future enhancement. With further improvements, it can evolve into a more secure and scalable attendance management platform.

References

- [1] S. Mehta *et al.*, "CNN-Based Automated Attendance System," *IJCSIT*, vol. 15, no. 2, pp. 45–50, 2025.
- [2] Intel OpenVINO Team, "Real-Time Face Detection on Edge Devices," Intel Documentation, 2024.
- [3] R. Sharma and P. Yadav, "Classroom Attendance System Using Haar Cascade Face Detection," *IEEE Xplore*, pp. 112–116, 2023.
- [4] H. Li *et al.*, "Face Recognition-Based Identity Authentication and Anti-Spoofing Techniques," *Springer*, vol. 10, pp. 201–210, 2022.
- [5] M. Singh and A. Kapoor, "Smart Attendance Platform Using Python and OpenCV," *IJERT*, vol. 13, no. 4, pp. 78–84, 2024.
- [6] J. Patel *et al.*, "Deep Metric Learning for Biometric Recognition," *Elsevier*, vol. 18, no. 3, pp. 55–63, 2023.