Smart Hybrid Attendance System Using Face Recognition and IoT- Based Physical Verification

Nikhil Singh[1], P Rajeev Kumar [2], K Shilpa[3]

[1,2] B Tech final year students ,ECE department, JB Institute of Engineering and Technology,Hyderabad, Telangana
[3] Assistant professor, ECE Department ,JB Institute of Engineering and Technology,Hyderabad, Telangana

---

Abstract - *The current project aims to rectify some problems associated with the conventional attendance system such as low efficiency, proxy attendance, and reliability issues. While the face recognition-based approach automates the process of attendance tracking, it is prone to spoofing attacks and does not require physical verification. Therefore, the current paper proposes the use of a Smart Hybrid Attendance System that incorporates deep learning face recognition and hardware-based physical verification using an ESP32 board. In our approach, the Mobile Face-net (TensorFlow Lite) is utilized to obtain 128D face embedding using the cosine similarity function for verification purposes. A two-phase verification strategy was adopted by which any detected face generates a pending record of attendance, which should be validated using physical hardware within a specified time frame. The proposed solution integrates the edge AI approach by employing the Raspberry Pi device for temporal validation and hybrid verification purposes. The experiment showed an accuracy rate of up to 92%*

***Keywords**: Face Recognition, Attendance System, Edge AI, ESP32, Hybrid Verification, IoT*

1. INTRODUCTION

Attendance tracking represents one of the core administrative processes in various organizations and institutions. Conventional systems for collecting attendance data involve laborious processes and pose significant security threats. Hence, automatic attendance systems utilizing biometric have been gaining increased attention in recent times.

Facial recognition has been commonly used since it is relatively easy to implement and involves minimal privacy concerns. Yet, most attendance processes depend solely on facial recognition without any further considerations, which makes them vulnerable to infiltration via photos or video to access the system.

In the current system, facial recognition technology will be combined with hardware components. The application of the ESP32 micro-controller will allow for accurate attendance registration only upon confirmation of the person's identity and presence.

2. Problem Statement

Present day methods of tracking attendance face several obstacles such as proxy attendances through images or impersonators, inadequate means of proving physical presence, detection problems caused by lighting differences or body postures, duplication of records because of frame-by-frame processing, and unreliability when implemented in the actual world scenario. Our project is aimed at developing an attendance marking system based on the application of computer vision technologies along with the use of embedded systems.

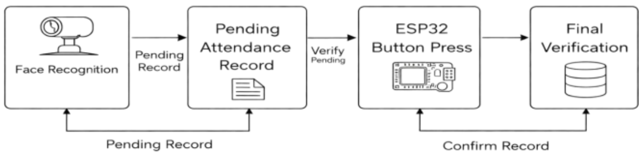

Figure 1: Outline Diagram of Proposed Smart Hybrid Attendance System

Figure 1: Figure walks you through the whole hybrid attendance process. It starts by scanning someone's face and checking if it matches any existing records. If it finds a match, the system creates a pending attendance entry. Now, the person has to press a button on the ESP32 device to prove they're actually present. When the system gets that confirmation, it saves the attendance for good in the database.

3. Proposed system

The solution makes use of a two-step authentication process:

I: Facial Recognition Authentication

Facial recognition software authenticates students based on embedding generated via deep learning.

II: Physical Confirmation Authentication

An ESP32 board demands for an active button click by the student within a given window of time.
Unlike the traditional approach, attendance is not immediately logged after facial recognition is completed. Instead, a pending-to-authentication process is employed.

4. System Architecture

The system is composed of several integrated components that work together to ensure efficient operation. The Face Monitoring Module captures frames from a video camera and performs face detection and recognition tasks. A Back end Server built using Fast API handles all API requests, manages user sessions, and performs necessary validation processes. The ESP32 module is responsible for sending HTTP requests during the authentication process. An SQLite database is used to store students' facial embedding and maintain attendance records. Additionally, a Web Dashboard provides a user-friendly interface for managing sessions and monitoring the system. Overall, the system is intentionally designed to be highly efficient and well-suited for deployment on edge devices such as Raspberry Pi.

Figure 5: Real-Time Face Detection and Recognition Output with Cosine Similarity Score

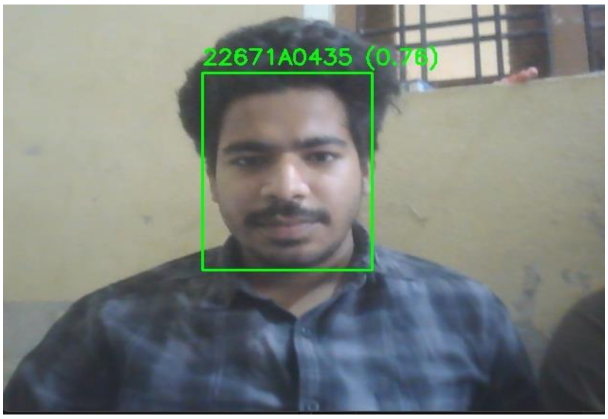

FIGURE 5.1: Real-Time Face Detection and Recognition Output with Cosine Similarity Score

Figure 5: Figure shows what the face recognition module sees in real time. You can spot the detected face inside a bounding box, and right next to it, you'll see the predicted student ID and similarity score. The system checks this score against its set threshold to decide if the face matches a registered student.

5. Methodology

5.1 Face Detection and Recognition

Face detection is performed using an Open CV Haar Cascade classifier, which efficiently identifies faces in captured frames. Once a face is detected, it is processed using the Mobile Face-net model implemented with TensorFlow Lite to extract a 128-dimensional feature vector representing unique facial characteristics. Identification is then carried out by comparing this feature vector with stored embedding using cosine similarity. If the similarity score is greater than or equal to 0.62, it is considered a valid identification; otherwise, it is treated as invalid. The threshold value of 0.62 is determined through empirical testing to achieve an optimal balance between false positives and false negatives.

5.2 Time-Based Validation Mechanism

An additional 10 seconds time limit is set in order to ensure that more than one face detection takes place during the period in question, in order to decrease the probability of a false positive because of an erroneous identification.

5.3 Hybrid Validation Technique

-The system operates in a structured sequence to ensure accurate attendance recording. Initially, face detection is performed, followed by face recognition to identify the individual. Once a successful identification is achieved, a pending record is created in the system. When the button on the ESP32 module is pressed, it triggers the transmission of an HTTP POST request to the back end server. The back end then carries out several identification processes, including session validation, verification of the pending record, and finally recording the attendance.

6. Implementation

The system is built using a combination of software and hardware components to ensure efficiency and reliability. The back-end is developed in Python using Fast-API, which handles API requests and system logic. Image processing tasks, including face detection, are carried out using Open-CV, while TensorFlow Lite is utilized for efficient model inference, enabling faster execution on edge devices. SQLite is used as the database to store facial embedding and attendance records. For hardware communication, the ESP32 module is employed to send HTTP requests and interact with the system. One of the key design decisions in this system is the separation of the face recognition component from the back-end functionality. This separation helps prevent performance issues that may arise due to the long runtime of recognition tasks, ensuring smoother and more reliable operation.

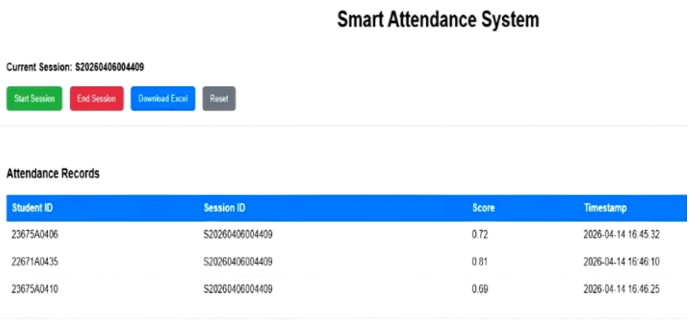

Figure 6: Web Dashboard Showing Session Control and Real-Time Attendance Records

Figure 6: Figure 5 shows the dashboard for managing attendance online. Here, you can start or end a session, reset the system, or export the attendance data when you need it. The dashboard also shows real-time attendance details, like student IDs, session IDs, confidence scores, and timestamps.

7. Result and discussions

Here's how the system comes together: The back-end uses Python with Fast-API. Image processing? That's all handled by Open-CV. For quick model inference, it taps into TensorFlow Lite. All the data lives in SQLite. And when it comes to hardware, everything connects through the ESP32. One of the main design choices was to keep face recognition separate from the back-end. By splitting them up, the back-end stays snappy and doesn't get bogged down with long-running tasks. It just works better this way.

8.Advantage and Limitation

Advantages

The system offers several key advantages that enhance its practicality and reliability. It significantly reduces instances of proxy attendance by incorporating accurate face recognition and validation mechanisms. Designed with efficiency in mind, it runs smoothly even on low-cost hardware such as edge devices. The setup process is simple, making it easy to deploy and use without requiring complex configurations. Additionally, by integrating multiple validation steps, the system outperforms traditional face-only systems in terms of overall reliability and accuracy.

Limitations

The system has a few limitations that need to be addressed for improved performance and security. Currently, it lacks liveness detection, which means it may not be able to distinguish between a real person and a spoof attempt such as a photo or video. There is also a possibility of misuse, as individuals could press the button on the ESP32 without proper supervision. Additionally, the system requires adequate lighting conditions to function effectively, as poor lighting can impact face detection and recognition accuracy. Lastly, the system is limited to operation with a single camera, which restricts its coverage and scalability

.

9. Future Scope

The system can be further enhanced through several improvements aimed at increasing security, scalability, and usability. One key upgrade is the addition of liveness detection techniques, such as blink or facial movement detection, to prevent spoofing attempts using photos or videos. Expanding the system to support multiple cameras would make it more suitable for larger classrooms by improving coverage and accuracy. Developing a mobile application interface can provide users with greater accessibility and ease of interaction with the system. Additionally, migrating the database from SQLite to a cloud-based solution would improve scalability, allowing the system to handle larger amounts of data and support multiple users or locations efficiently

.

10. Conclusion

This project brings together face recognition and a simple physical check to track attendance. It's not just all about AI—there's a hardware step in there to make everything more dependable.The system deals with daily use pretty smoothly and solves some of the problems you see in other setups.

11. References

1. Solanki K., Pittalia P., "Review of face recognition techniques", International Journal of Computer Applications, 2016, 133 (12), 20–24.

2. Deng W., Hu J., Guo J., "Face recognition via collaborative representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40 (10).

3. Pei T., Zhang L., Wang B., Li F., Zhang Z., "Decision pyramid classifier for face recognition", Pattern Recognition, 2016, 64, 305–313.

4. Shi H., Wang X., Yi D., Lei Z., Zhu X., Li S., "Cross-modality face recognition", IEEE Signal Processing Letters, 2017, 24 (1), 81–85.

5. Ding C., Tao D., "Ensemble convolutional neural networks for face recognition", IEEE Transactions on Pattern Analysis, 2016.

6. Wu D., Tang Y., Lin G., Hu H., "Robust face recognition using deep learning", Journal of Optoelectronics Laser, 2016, 27 (6), 655–661.

7. Sun Y., Zhao J., Hu Y., "Supervised sparsity preserving projections", Proceedings of SPIE, 2017.

8. Best-Rowden L., Jain A.K., "Longitudinal study of face recognition", IEEE Transactions on Pattern Analysis, 2018.
9. Duan Y., Lu J., Feng J., Zhou J., "Context-aware feature learning for face recognition", IEEE Transactions on Pattern Analysis, 2017.
10. Mudunuri S., Biswas S., "Low resolution face recognition under variations", IEEE Transactions on Pattern Analysis, 2016, 38 (5), 1034–1040.
11. Li Y., Song W., Cheng C., "Face recognition-based attendance system using Raspberry Pi", Microcontrollers and Embedded Systems, 2016.
12. Chakraborty S., Singh S., Chakraborty P., "Local gradient patterns for face recognition", IEEE Transactions on Circuits and Systems, 2016.
13. Zhu X., Liu H., Lei Z., Shi H., Yang F., Yi D., "Large-scale face recognition learning", International Journal of Computer Vision, 2019.
14. Nguyen H., Yang W., Sheng B., "Dictionary learning for face recognition", Neurocomputing, 2016.
15. Ji H., Sun Q., Yuan Y., Zhang G., "Collaborative probabilistic labels for recognition", Pattern Recognition, 2017.
16. Wang L., Chen S., "Joint representation classification for face recognition", Pattern Recognition, 2017.
17. Schroff F., Kalenichenko D., Philbin J., "FaceNet: A unified embedding for face recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.
18. Howard A.G., et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications", 2017.
19. Sandler M., Howard A., Zhu M., Zhmoginov A., Chen L., "MobileNetV2: Inverted residuals and linear bottlenecks", CVPR, 2018.
20. Viola P., Jones M., "Rapid object detection using a boosted cascade of simple features", IEEE CVPR, 2001.