



Available Online at www.hithaldia.in/locate/ECCN
All Rights Reserved

ORIGINAL CONTRIBUTION

A Portable Biometric System for Student Verification in Exam Halls using ESP32

¹Shivam Seal, ²Dipayan Biswas and ³Moumita BasakNath

^{1,2,3}*Techno Main Salt Lake, Kolkata, India,*

Email Id: ¹sealshivam09@gmail.com, ²dipayan11biswas@gmail.com, ³moumitapapai@gmail.com

ABSTRACT

Impersonation and unregistered candidates giving exams on behalf of others is becoming a growing concern, especially in institutions where security is lax and oversight is limited. This paper presents a compact, portable biometric system that enables real-time fingerprint and face-based authentication for exams. Built around the ESP32 microcontroller, the system uses the R307S fingerprint sensor and ESP32-CAM module for biometric scanning and facial recognition, with data synced to Firebase Realtime Database. Designed for flexibility and mobility, it eliminates the need for traditional roll calls or paper-based logs. Invigilators can move freely across the exam hall verifying identities, while attendance is updated to the cloud almost instantly. The added face recognition layer improves security and helps detect failed authentications, such as mismatched fingerprints or undetected faces, making the system more reliable and easier to troubleshoot during exams.

KEYWORDS: Biometric Verification, Fingerprint Authentication, ESP32-CAM, Fire- base Real-time Database, Portable Attendance System, Student Authentication, Face recognition

1. INTRODUCTION

In most schools and colleges, roll calls or signatures on attendance sheets are still used to mark attendance during exams. These methods can be slow and unreliable. However, fingerprints, biometrics, offer a much better alternative. Since everyone has a unique fingerprint, verification is faster and harder to fake. Biometric recognition has become a cornerstone in secure environments such as exam halls due to its reliability over manual methods [1]. This project uses the ESP 32-CAM board along with the R307S fingerprint module to instantly identify and log student attendance instantly, all data being updated live in the real-time database named Firebase, which is a service provided by Google. Verify and record student attendance in real time using Firebase. The

compact form factor makes it easy to carry across exam halls, and photo capture with the onboard

camera can further enhance reliability. While prior works have focused on biometric attendance, they often overlook the specific need for invigilator mobility in large exam halls. Our work addresses this gap by making the following novel contributions:

- A truly portable architecture:** We combine the ESP32-CAM and a dedicated dev board into a self-powered, handheld unit, adding a mobility factor to the verification process from fixed locations.

- Enhanced security via dual-biometrics:** By requiring both fingerprint and facial confirmation, our system significantly reduces the risk of false positives compared to single-modality systems.

•**Real-time synchronization:** We leverage the Firebase Database to provide immediate, cloud-based and real time attendance logging, accessible remotely and instantly.

This integrated approach presents a practical and robust solution for modernizing exam administration.

2. Related Work

Most fingerprint-based systems deployed in educational institutions are fixed installations, typically mounted on walls or entrance gates. While effective in controlled environments, these systems are less suitable for large exam halls where student movement can cause operational delays. Although prior studies have emphasized biometric accuracy, few have addressed the combination of mobility, cloud synchronization, and wireless real-time updates. A recent survey confirms the growing adoption of fingerprint and facial authentication in academic settings [2]. To address existing limitations, our system enables invigilators to carry the authentication device and verify students individually at their seats while maintaining continuous online synchronization. Chakraborty and Bhowmick proposed a stationary IoT-based dual-biometric attendance framework [5], whereas our work extends this concept by introducing portability and Firebase-based real-time cloud integration.

3. Proposed System Architecture

We used the ESP32 Dev Board as the main microcontroller, which connects to Wi-Fi and communicates with Firebase. The R307S fingerprint sensor handles biometric input, while an LCD displays prompts such as 'Scan Finger' or 'Face Found'. For facial recognition, we utilized the Dlib embedding model to extract facial feature encodings, offering real-time accuracy on constrained hardware [4]. The student's name, along with the date and time of entry, is instantly uploaded to Firebase for secure cloud storage via the Realtime Database [9]. Since biometric systems demand low-latency integration between

hardware and software layers, our design reflects a co-engineered architecture for real-time performance [6].

3.1. Hardware Overview

•**ESP32 Development Board:** Microcontroller with built-in Wi-Fi. Controls the system and handles data transmission to Firebase.

•**ESP32-CAM:** Microcontroller with built-in Wi-Fi and camera support. Controls image capture and face detection with Python Flask and Open CV. The ESP32-CAM board, used for both Wi-Fi and image capture, reduces component count and improves device portability [7].

•**R307S Fingerprint Module:** The R307S optical sensor offers reliable fingerprint matching suited for mobile exam environments [8]. Used to register and match fingerprints. Can store more than 100 fingerprint templates.

•**LCD 16x2 Display:** Gives real-time instructions and feedback to users.

•**18650 Batteries:** The 18650 lithium-ion rechargeable battery with a capacity of 2000mAh used to power the system.

•**4x4 Keypad:** The keypad helps the student enter the roll number and the section in which he studies.



Fig. 1: Complete Circuit of Biometric Attendance System

3.2 Software Overview

- Google Firebase:** A real-time database, all attendance records (timestamp + ID) are stored here.
- Arduino IDE:** Used for coding and uploading the firmware onto the ESP32 Development Board and ESP32-CAM.
- Python Flask:** used to handle HTTP requests from ESP32 devices. When a student is identified by fingerprint or face matching, ESP32 sends a request to the Flask server.
- Open CV:** Open-Source Computer Vision Library is used for real-time face detection and image processing. The system leverages OpenCV along with deep learning models (like DeepFace or face recognition) to encode, compare, and verify student faces.

3.2 System Components Table

Table 1: Hardware Components and Their Functions

Component	Function
ESP32DevBoard	Maincontroller,handlesdatatransmissiontoserverandbackfromit
ESP32CAM	HandlesWi-
R307SFingerprintSensor	Fiandimagecaptureusedforfacerecognition
LCD 16x2 Display	Captures and verifies fingerprint data of student
18650Batteries 4x4	Showsverificationstatusouser
Keypad	Powersthedeviceandaddsportabilityfactor
MT3608BoostConverter	Helpstoenterdatatothedevicewhichspeedsupverification
JumperWires	This module steps up the voltage from the 3.7V-4.2V 18650 batteries to a stable 5V output
Breadboard	Helpstomakeconnectionbetweenthecomponent sandmodules
	Serves as a base on which the connection between components and wires are made

4 System Workflow

The project workflow follows a structured process of data input, biometric verification, back-end processing, and output feedback. The detailed sequence of operations is as follows.

1. The student initiates the process by entering their Roll Number, typically a four-digit unique identifier, and selects their Section (A, B, C, or D) using the 4x4 matrix keypad. The entered details are displayed on the 16x2 I2C-based LCD screen for confirmation.
2. Once the roll number and section are entered, the fingerprint authentication process begins. The R307S fingerprint sensor is activated and prompts the student to place their finger on the sensor. The scanned fingerprint is compared to previously enrolled templates stored in the sensor’s onboard memory. If it does not match, an error message is displayed, and the student is prompted to retry. 3 retries are allowed after which the whole process restarts from entering the roll number.
3. Upon successful fingerprint verification, the system proceeds to facial recognition. The ESP32-CAM module captures a real-time image of the student’s face. OpenCV-based facial detection and recognition algorithms process the image and compare it with the stored data set (in flash memory or via Firebase). If the face is not recognized, the student is prompted to try again. For lightweight face detection, the Haar cascade technique introduced by Viola and Jones is still effective on embedded platforms [3].
4. After dual biometric verification, the ESP32 Dev Board connects to the Firebase via Wi-Fi and retrieves the student’s name and section using the roll number as a unique key. This step ensures that the correct biometric data are matched with academic records.
5. Once authenticated, a final attendance log is prepared that contains the student’s name, roll number, section, date, and verification time. This log is sent using HTTPS requests to a Python Flask server hosted on a PC. The server writes the data to an Excel file (.xlsx) using the openpyxl library.
6. The LCD provides immediate feedback with messages like 'Marked Present', along with the student’s name and timestamp. If any step fails, messages such as 'Invalid Fingerprint', 'Face Not Recognized', or 'Database Error' are shown, and the process pauses until the issue is resolved.

This procedure ensures that attendance is captured accurately, securely, and in real time without

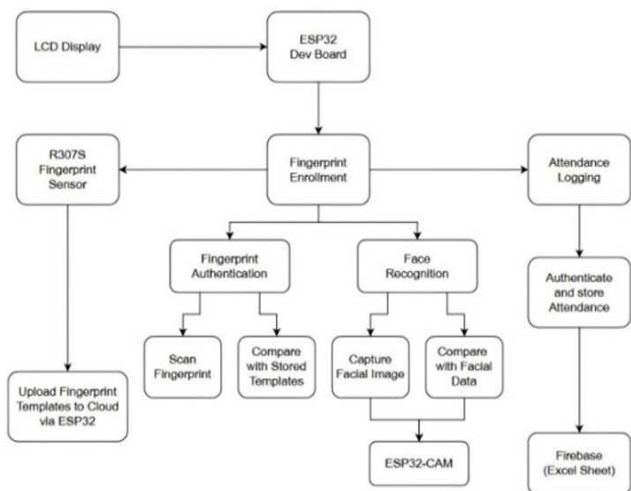


Fig.2: System Workflow of the Attendance System

manual supervision or paperwork. Increase transparency, prevent impersonation, and ensure that only authenticated students can enter the examination hall.

5 Implementation

5.1 Device Boot-Up

When powered on, the device shows two menu options to select. One for checking the battery percentage to ensure that the device has sufficient power before being used for attendance purposes in the exam hall. The other option is to start the student verification

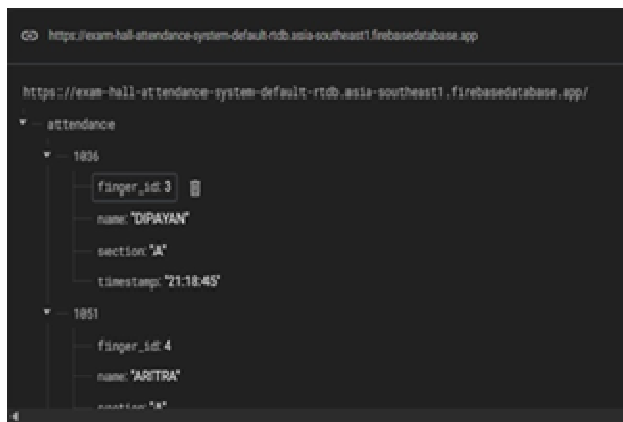
5.2 Fingerprint Capture and FaceRecognition

When powered on, the ESP32-CAM connects to Wi-Fi. The fingerprint sensor checks if the scanned fingerprint matches any stored template. If it does, then it moves to facial recognition using the Open CV and Python face recognition script

5.3 Cloud Sync and Real-Time Feedback

Firestore is updated in real time. This helps teachers view attendance logs remotely, even during the exam. There is no need for manual record keeping. In addition, an Excel file is saved

with all attendance data for easy data transfer from Fig. 3 Real time update in Firestore



one system to another.Ensuring secure data transmission and storage is paramount to protect sensitive biometric information from unauthorized access or breaches. [12]

5.4User Interaction via LCD

The LCD shows messages such as 'Place Finger', 'Try Again', or 'Marked Present'. This helps users understand what is happening and reduces confusion.



Fig. 4: LCD showing message for user

5.5 Image Capture Feature

The ESP32-CAM includes a camera which is used to scan the student’s face and used for facial recognition by comparing to the already stored photo of the student in the server. Local Binary Patterns (LBP) were also considered for their compact and effective texture descriptors [10]. Despite these advancements, facial recognition systems face challenges related to lighting

conditions, facial expressions, and occlusions, which can impact recognition accuracy. Additionally, concerns about data privacy and the potential for misuse of facial data have been raised, necessitating robust data protection measures. [11]

6. Results and Discussion

6.1 System Response

The whole process takes about 10–15 seconds. Once matched, the data show up in Firebase almost immediately. Issues mainly occur when:

- Wi-Fi signal is weak or disconnected.
- The finger is dirty or not placed properly.
- ESP32 camera is not held at the face-level position, causing the face to not be detected.
- Memory usage is high, which causes occasional lag.

Despite that, we achieved more than 90% success during the testing.

6.2 Benefits

- Portable and wireless, easy to carry by invigilators during exams.
- Reduces impersonation risks.
- Real-time attendance recording without paperwork.

6.3 Drawbacks

- An internet connection is required.
- Face recognition accuracy is heavily affected by ambient lighting.
- The current setup is designed for a fixed number of students.

7. Conclusion and Future Scope

This project shows how portable biometric-based systems can help maintain fairness in exams. Since it is wireless and cloud-connected, it works well in large rooms. Future upgrades could include:

- Offline support with local storage.
- Dedicated mobile application for live monitoring of attendance data.
- Upgrade the ESP32-CAM module to a higher-resolution image sensor such as the OV5640 to

improve facial recognition under challenging lighting conditions

- Improve the user interface by incorporating touch screen displays with a graphical interface.
- Scale it to support hundreds of simultaneous users in multiple exam halls by making enhancements to networking, cloud storage, and database management.

References:

- [1] Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14(1), 4-20.
- [2] Alam, M., Ahmad, M., Khan, R., Alazab, M., & Imran, M. (2020). A survey on biometric recognition systems and their applications. *Multimedia Tools Appl.* 79(9), 6067–6102.
- [3] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In: *Proc. IEEE CVPR*, pp. 511–518.
- [4] King, D. E. (2009). Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10, 1755-1758.
- [5] Chakraborty, S., & Bhowmick, P. (2022). IoT-based student authentication system using face and finger- print recognition. In: *Proc. ICSTCEE*, pp. 1–5.
- [6] Maio, D., Maltoni, D., Cappelli, R., Wayman, J.L., & Jain, A.K. (2004). *Biometric systems: Technology, design and performance evaluation*. Springer.
- [7] Espressif Systems: ESP32-CAM Datasheet. https://www.espressif.com/sites/default/files/documentation/esp32-cam_datasheet_en.pdf.
- [8] Next Biometrics: R307S Fingerprint Sensor Datasheet. <https://nextbiometrics.com/wp-content/uploads/2021/04/R307-datasheet.pdf>.
- [9] Google: Firebase Realtime Database. <https://firebase.google.com/products/realtime-database>.
- [10] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(12), 2037–2041.

[11] Rao, A. (2022).AttenFace: A Real Time Attendance System using Face Recognition. arXiv preprint arXiv:2211.07582.

[12] Baral, M. (2019) Biometric Attendance System Using Arduino. GitHub repository, <https://github.com/MonalisaBaral/BiometricAttendance-System-Using-Arduino>.