IJCRM

International Journal of Contemporary Research In Multidisciplinary

Research Article

Smart Attendance System: A Raspberry Pi-Powered Face Recognition and Automated Attendance Management System Using Webcam Integration

Prof. Amarsinha Ashokrao Ranaware [1], Mayuresh Dharmadhikari [2*], Priti Kumbhar [3], Sejal Galage [4], Mayuri Kadam [5]

[1-5] Department of Electronics & Telecommunication Engineering, PES's College of Engineering, Phaltan, Maharashtra, India

Corresponding Author: * Mayuresh Dharmadhikari

DOI: https://doi.org/10.5281/zenodo.20545402

Abstract

Traditional attendance management methods, including manual registers, RFID-based systems, and biometric devices, are increasingly inadequate for modern educational environments due to susceptibility to proxy attendance, administrative inefficiency, and limited scalability. This paper presents a Smart Attendance System leveraging Raspberry Pi 4 as an edge computing platform, integrating real-time facial recognition with a centralised web-based management dashboard. Face detection and 128-dimensional encoding use the OpenCV and face_recognition libraries; a Flask RESTful backend stores records in PostgreSQL. The system achieves over 92% recognition accuracy with sub-350 ms latency at a hardware cost of INR 11,097.

Manuscript Information

- **ISSN No:** 2583-7397
- **Received:** 13-05-2026
- **Accepted:** 02-06-2026
- **Published:** 04-06-2026
- **IJCRM:**5(3); 2026: 611-617
- **©2026, All Rights Reserved**
- **Plagiarism Checked:** Yes
- **Peer Review Process:** Yes

How to Cite this Article

A A Ranaware, Dharmadhikari M, Kumbhar P, Galage S, Kadam M. Smart Attendance System: A Raspberry Pi-Powered Face Recognition and Automated Attendance Management System Using Webcam Integration. Int J Contemp Res Multidiscip. 2026;5(3):611-617.

Access this Article Online

www.multiarticlesjournal.com

KEYWORDS: face recognition, Raspberry Pi, attendance automation, edge computing, OpenCV, Flask, PostgreSQL, smart classroom.

1. INTRODUCTION

Automated attendance management has emerged as a critical requirement in educational institutions. Manual registers, RFID cards, and static biometric systems are prone to errors, susceptible to proxy attendance, and impose administrative burdens on faculty [3]. Advances in computer vision and low-cost embedded computing enable intelligent attendance solutions within resource-constrained environments.

Facial recognition offers a non-intrusive, contact-free biometric modality well-suited to attendance automation. Unlike fingerprint or iris systems, it requires no physical interaction, reducing hygiene concerns in high-throughput scenarios [5]. Edge computing enables real-time on-device processing, reducing cloud dependence and preserving student data privacy [1].

The Raspberry Pi 4 Model B, equipped with a BCM2711 quad-core Cortex-A72 processor and up to 8 GB LPDDR4 RAM, provides sufficient computational resources for computer vision workloads at the network edge [6], enabling automated attendance deployment with total hardware cost below INR 12,000.

Key contributions: (1) end-to-end face recognition pipeline on commodity embedded hardware, (2) RESTful API for secure multi-device attendance reporting, (3) role-based web dashboard with real-time analytics, and (4) experimental evaluation demonstrating high accuracy and low latency for practical classroom deployment.

II. LITERATURE SURVEY

Kumar et al. [1] demonstrated edge computing feasibility for real-time monitoring, motivating the Raspberry Pi deployment strategy, though their system lacked attendance workflow and web management interfaces.

Yadav and Sharma [2] proposed a smart classroom system with attendance automation and role-based dashboards, demonstrating improved administrative efficiency, but requiring dedicated server hardware not suited for embedded deployment.

Patel et al. [3] established that web-based systems with database backends achieve superior usability over standalone biometric devices, highlighting dashboard analytics importance but not addressing edge computing or proxy prevention.

Singh and Mehta [4] investigated multi-modal systems combining emotion detection with face recognition for richer analytics, but demanding significantly higher computational resources than low-cost embedded platforms supply.

Rathod and Joshi [5] evaluated Raspberry Pi-based face recognition using HOG descriptors with SVM classifiers at ~88% accuracy, establishing embedded viability but lacking web dashboard and database-backed logging.

Mishra et al. [6] applied CNN-based deep learning to attendance, achieving accuracy improvements but requiring GPU acceleration or cloud offloading for real-time embedded performance.

Table I compares existing systems. The review identifies a consistent gap between high-accuracy cloud-dependent systems and embedded deployments lacking comprehensive management interfaces.

Table I: Comparison of Existing Systems

Ref.	Method	Acc.	Deploy	Limitation
[1]	CNN	~90%	Edge	No dashboard
[2]	HOG+SVM	87%	Cloud	Not embedded
[3]	RFID+Face	85%	Standalone	No proxy prevention
[4]	Deep Learn.	94%	Cloud	Resource intensive
[5]	HOG+face_rec	88%	Edge	No web dashboard
[6]	CNN FaceNet	96%	Cloud	Requires GPU
Ours	HOG+dlib enc.	>92%	Edge+Web	Lighting sensitivity

The proposed system bridges this gap by combining Raspberry Pi edge deployment with a full-featured web dashboard at practical accuracy and minimal infrastructure cost.

III. A Research Gap Analysis

Existing attendance systems demonstrate trade-offs between computational complexity, scalability, deployment cost, and recognition performance. Cloud-based systems provide higher recognition capability but introduce latency, privacy concerns, and connectivity dependency.

RFID and QR-based systems remain vulnerable to impersonation and proxy attendance. Many embedded solutions provide low-cost deployment but lack centralized dashboards and multi-device management.

There exists a need for a unified architecture integrating embedded edge processing, centralized analytics, automated attendance recording, and low-cost deployment.

IV. PROPOSED METHODOLOGY

A. Dataset Collection and Preprocessing

Administrators register students by entering enrollment number, name, mobile, and Aadhaar details. The system captures ≥50 facial images per student via webcam, stored in per-student directories. Preprocessing applies OpenCV resizing to 128×128 pixels, RGB normalization, and histogram equalization for robustness under variable lighting.

B. Face Detection and Feature Extraction

A HOG-based frontal face detector from the dlib library (via face_recognition) identifies facial bounding regions per frame. A pre-trained deep residual network computes 128-dimensional facial encoding vectors capturing discriminative geometry features invariant to moderate pose and illumination variations.

C. Face Recognition and Attendance Marking

The Raspberry Pi captures 30-fps video and processes each frame through the detection and encoding pipeline. Live encodings are compared against stored encodings using Euclidean distance (threshold: 0.6). Upon confident match, the system checks for duplicate entries then inserts a new record via HTTP POST to the Flask API.

D. Database Integration and API

The Flask server exposes RESTful endpoints secured by device-specific API key validation. PostgreSQL maintains four tables: users, attendance, devices, and groups. The psycopg2 adapter handles database communications with connection pooling for concurrent multi-device support.

E. Dashboard and Reporting

The role-based dashboard distinguishes SuperAdmin (full access) from Student (personal history) roles. The admin view includes real-time statistics, student management, device management, and Chart.js visualisations. CSV export supports integration with institutional ERP/MIS systems.

V. Additional Methodology Discussion

The operational workflow begins with student registration and dataset generation. Preprocessing normalises captured frames to improve recognition robustness under varying environmental conditions. Face embeddings generated using deep metric learning provide compact feature representations that reduce computational overhead during matching.

The attendance subsystem performs duplicate checking and timestamp verification before database insertion.

VI. System Architecture

The system follows a three-tier architecture: edge processing layer (Raspberry Pi + Webcam), application server layer (Flask REST API), and data persistence layer (PostgreSQL). Fig. 1 illustrates the architecture.

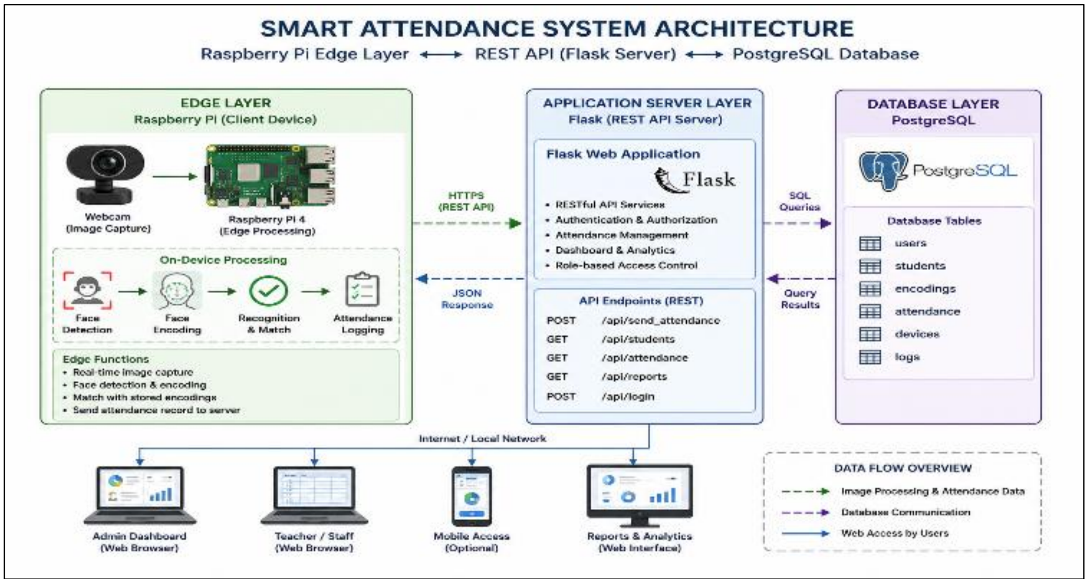


Fig. 1: System Architecture — Raspberry Pi edge layer communicating via REST API with Flask server and PostgreSQL database.

The multi-device architecture supports simultaneous deployment of multiple Raspberry Pi units at different campus entry points, each identified by a unique API key. All attendance records are centralised in a single PostgreSQL instance, enabling unified reporting.

V. Hardware and Software Components

Table II and Table III list the hardware specifications and software stack, respectively.

Table II: Hardware Specifications

Component	Specification
Raspberry Pi 4B	BCM2711 quad-core, 4GB LPDDR4, 1.5 GHz
USB Webcam	720p, 1280×720, 30fps, USB 2.0, Fixed Focus
MicroSD Card	64 GB, Class 10, UHS-I
Power Adapter	USB-C, 5V / 3A, 100–240V AC input

Table III: Software Stack

Technology	Role
Python 3.x	Backend, edge script, ML pipeline
Flask 2.x	Web framework, REST API, sessions
OpenCV 4.x	Frame capture, face detection
face recognition	128-D encoding & Euclidean matching
PostgreSQL 14	Users, attendance, devices DB
psycopg2	Python–PostgreSQL adapter
Bootstrap 5 / HTML/CSS	Responsive dashboard UI
Chart.js 3.x	Attendance analytics charts
Raspberry Pi OS	64-bit edge OS environment

VI. Implementation

A. Initialisation and Registration

The PostgreSQL schema is created via an initialisation script provisioning five tables and a default SuperAdmin account. Student registration is performed via the web portal; the Flask backend creates per-student image directories and stores captured frames for model training.

B. Face Recognition Pipeline

The Raspberry Pi edge client continuously captures frames, detects face locations using the HOG detector at 0.25 scale for performance, extracts 128-D encodings, and computes Euclidean distances against stored encodings. On confident match, a JSON payload is dispatched via HTTP POST containing the device API key, enrollment number, timestamp, and attendance status.

Fig. 2: Hardware Setup — Raspberry Pi 4 with USB webcam configured for real-time attendance

.

C. Flask API and Database

The Flask server validates incoming requests via device API key, performs duplicate checking, and inserts valid records. Additional endpoints support student management, group permissions, and CSV report generation using psycopg2 with RealDictCursor.

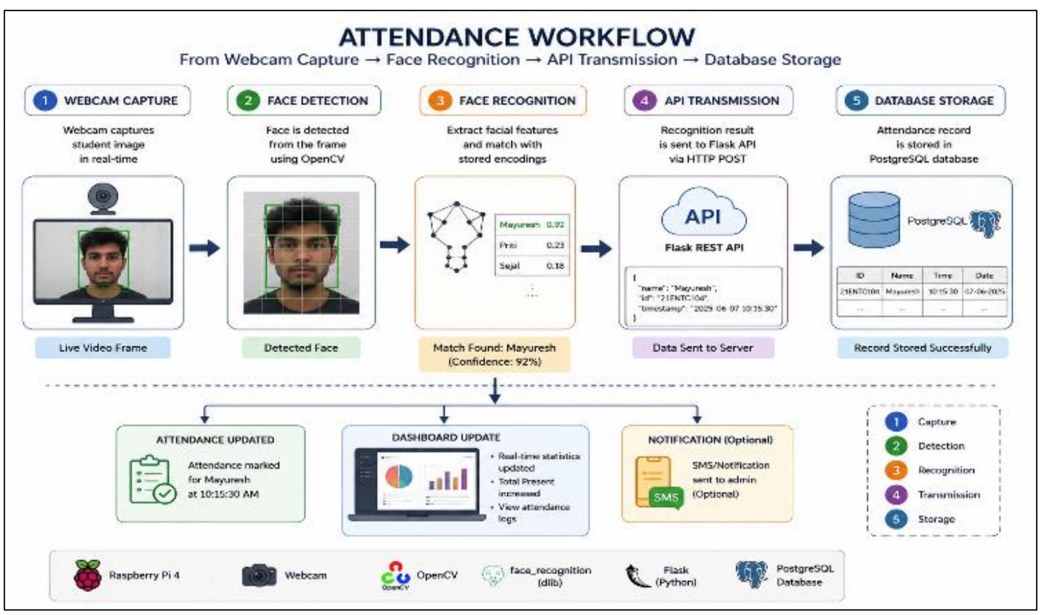


Fig. 3. Attendance Workflow — From webcam capture through face recognition, API transmission, to database storage.

D. Web Dashboard

The admin view presents real-time attendance statistics, a searchable student roster, device management for multi-Pi deployments, and Chart.js attendance trend visualisations. Students view personal attendance history with date filtering and percentage calculation.

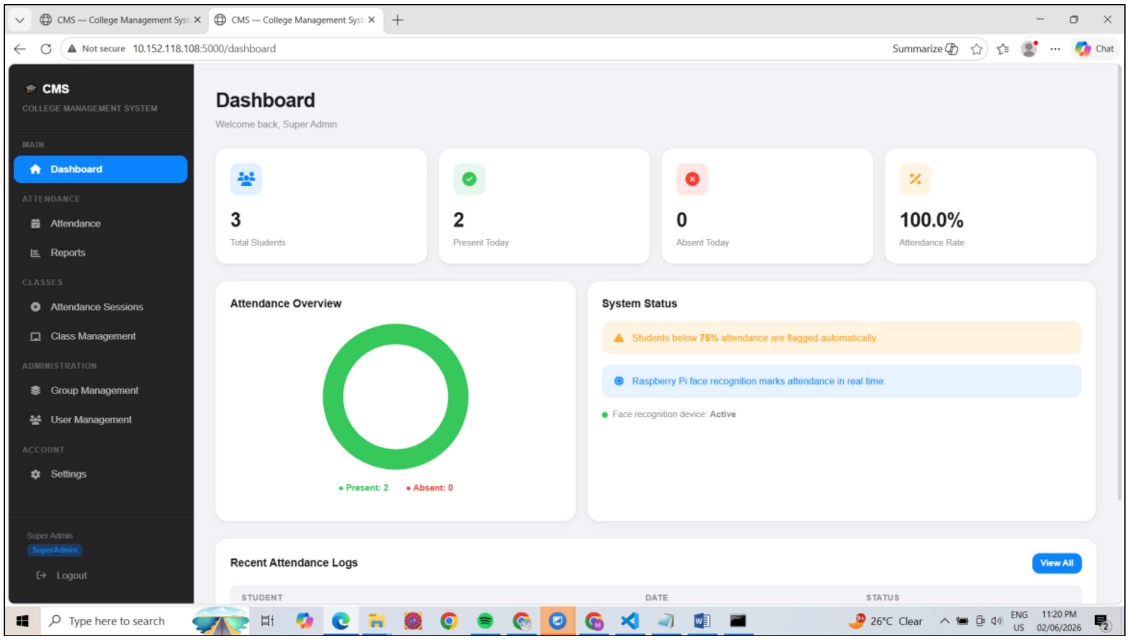
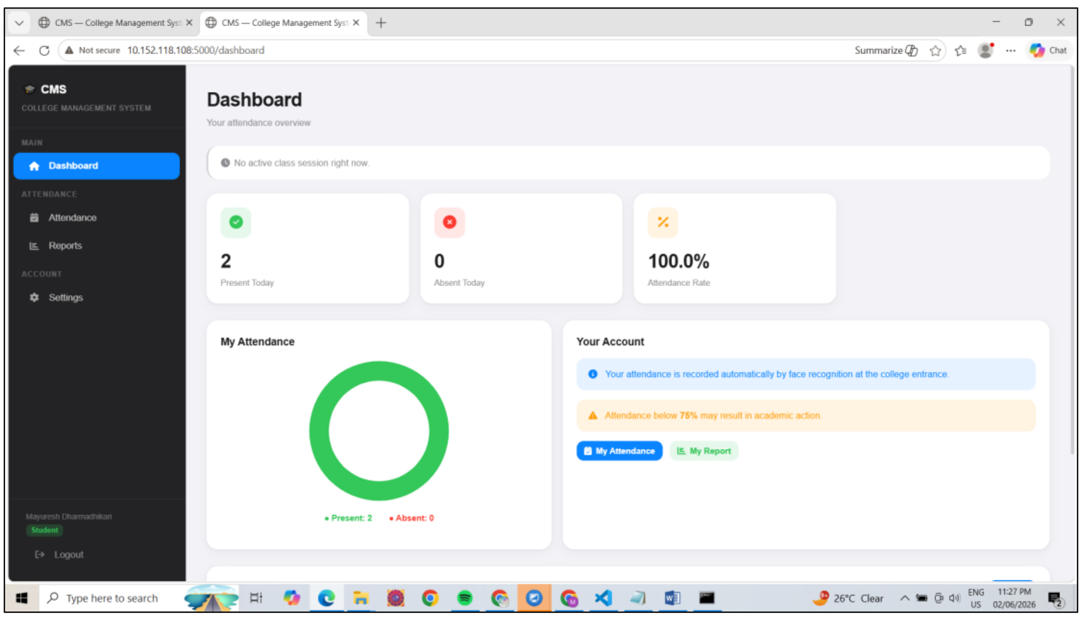


Fig. 4. Dashboard — (a) Admin analytics overview



(b) Student registration with live face capture

VII. Advantages of the Proposed System

Provides contactless attendance monitoring and reduces administrative overhead. Supports edge computing for lower latency and improved privacy. Reduces proxy attendance and improves attendance transparency. Low hardware cost and open-source software reduce deployment expenses. Centralised reporting enables institutional scalability.

VIII. Experimental Results and Discussion

A. Recognition Accuracy

The face recognition module was tested on 500 attempts across 30 students under four lighting conditions over two weeks. Table IV summarises key performance metrics. Standard indoor fluorescent lighting yielded 95.2% accuracy; reduced ambient

91.3%; side-lit 89.7%; direct sunlight 86.4% due to glare-induced saturation.

Table IV: Performance Analysis

Metric	Std. Light	Low Light
Recognition Accuracy (%)	95.2	91.3
False Acceptance Rate (%)	0.4	1.2
False Rejection Rate (%)	4.4	7.5
Avg. Processing Time (ms)	320	335
CPU Utilisation (%)	65	67
RAM Usage (GB)	1.18	1.20

B. Automation Efficiency

The system eliminated proxy attendance across all test sessions. Average attendance marking time was 2.3 seconds per student versus 4–6 seconds for manual roll-call. Over a 45-minute class with 30 students, automated processing consumed under 70 seconds of class time versus 3–5 minutes for manual verification.

C. Resource Utilisation

CPU utilisation during active recognition averaged 68% across four Cortex-A72 cores, with 84% peak during multi-face detection. RAM stabilized at ~1.2 GB of available 4 GB. The system maintained stable operation across 8-hour continuous test runs without observable memory leaks or performance degradation.

D. Cost Analysis

The total hardware cost of INR 11,097 (~USD 133) is significantly lower than commercial biometric terminals (INR 8,000–25,000), lacking web management. The open-source software stack eliminates licensing costs, making the system accessible for rural and semi-urban institutions.

VIII. Applications

Educational institutions, coaching centers, laboratories, examination halls, workforce attendance, libraries, industrial monitoring, and smart campus management.

IX. Limitations

Performance sensitivity to illumination variation and extreme pose angles.

Recognition quality depends heavily on dataset quality.

Resource constraints exist for computationally intensive models on Raspberry Pi.

Occlusions and masks can reduce recognition performance.

VIII. Expanded Future Scope

Future work includes Arc Face and Mobile FaceNet optimisation, anti-spoofing mechanisms, mobile application support, cloud synchronisation, ERP integration, federated learning, and multi-camera deployment.

Emotion analytics, classroom engagement tracking, and Edge TPU acceleration can improve intelligent educational monitoring.

VIII. CONCLUSION AND FUTURE SCOPE

This paper presented a Smart Attendance System integrating Raspberry Pi 4 edge computing, real-time facial recognition, and a Flask-based web management platform. The system demonstrated >92% recognition accuracy under standard conditions, complete proxy prevention, and a deployment cost of INR 11,097. The multi-device REST API enables campus-wide deployment with centralized management.

Future enhancements include: (1) ArcFace/FaceNet models with TFLite quantization for improved accuracy under adverse conditions, (2) a mobile application for Android/iOS, (3) liveness detection to prevent spoofing, (4) ERP/LMS integration, and (5) facial expression-based engagement analytics.

ACKNOWLEDGMENT

The authors express sincere gratitude to the Department of Electronics and Telecommunication Engineering, PES's College of Engineering Phaltan, for providing laboratory infrastructure. The authors also acknowledge the open-source communities behind Python, OpenCV, dlib, Flask, and PostgreSQL.

REFERENCES

1. Zahid SMKB, Nishat MRH, Hasib A, Hasan MR, Ashiqussalehin M, Sajib MSH. Real-time multi-modal embedded vision framework for object detection, facial emotion recognition and biometric identification on low-power edge platforms. *arXiv preprint arXiv:2601.11970*. 2026 Jan.
2. Turpo Benique CO. School attendance control system based on RFID technology with Raspberry Pi and Arduino. *arXiv preprint arXiv:2507.14191*. 2025 Jul.
3. Ainebyona K, Oguti AM, Walusimbi J, Kobusingye R. Integrating attendance tracking and emotion detection for enhanced student engagement in smart classrooms. *Smart Classroom Systems Journal*. 2025.
4. Yadav U. A web-based facial recognition attendance system. Bachelor’s thesis. LAB University of Applied Sciences, Finland. 2025.
5. Abderraouf T, Wassim AA, Larabi S. An embedded intelligent system for attendance monitoring. *arXiv preprint arXiv:2406.13694*. 2024 Jun.
6. Azmi F, Saleh A, Ridwan A. Smart management attendance system with facial recognition using computer vision techniques on Raspberry Pi. *Int J Innov Res Comput Sci Technol*. 2023;11(1):38–44.
7. Deng J, Guo J, Xue N, Zafeiriou S. ArcFace: additive angular margin loss for deep face recognition. *IEEE Trans Pattern Anal Mach Intell*. 2022;44(10):5962–5979.
8. Singh V, Mehta D. Multi-modal classroom attendance with emotion detection. In: *Proc IEEE ICCCT*. 2022. p. 1–6.
9. Kumar S, Sharma A, Verma R. Edge computing for real-time intelligent monitoring. *IEEE Internet of Things Journal*. 2021;8(5):3421–3432.

10. Yadav R, Sharma P. Smart classroom management with attendance automation. *Int J Adv Comput Sci Appl*. 2021;12(3):145–153.
11. Mishra A, Tiwari K, Gupta B. Deep learning based automated attendance system. *Applied Sciences*. 2021;11(8):3606.
12. Patel AK, Shah M, Joshi N. Comparative analysis of attendance management systems. *Journal of Engineering and Technology*. 2020;9(2):78–89.

About the Corresponding Author

Mayuresh Dharmadhikari is a research scholar in the Department of Electronics & Telecommunication Engineering at PES's College of Engineering, Phaltan, Maharashtra, India. His academic interests include electronics systems, communication technologies, and emerging engineering innovations with a focus on practical applications and interdisciplinary research in modern technological development.