IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1(2) (2026)

ijerst.editor@gmail.com
editor@ijerst.com

ISSN 2319-5991 www.ijerst.com

Peer Reviewed, Referred & Indexed Journal

Research Paper

FACIAL FEATURE BASED STUDENT ATTENDANCE AUTOMATION USING DEEP LEARNING

G. Pavan Kalyan[1], CH.Jeetha Lalitha[2], B.Karthik[3], D.Niranjan[4]
Proff. Dr. K.V. Satyanarayana[5]
[1,2,3,4,5]Department of Computer Science and Engineering(Data Science)
[1,2,3,4,5]Raghu Engineering College, Dakamarri(V), Visakhapatnam, Andhra Pradesh, India

***Abstract*—Conventional attendance recording methods in academic institutions — manual roll calls and paper registers — are inefficient, time-consuming, and prone to errors including proxy attendance. This paper presents a Smart Attendance System that leverages real-time face recognition to automate the entire attendance workflow. The proposed system employs the Local Binary Pattern Histograms (LBPH) algorithm for face detection and recognition, integrated with a Flask-based web application, a MySQL relational database, and an automated email notification module. The system supports a four-punch mechanism (Early In, Late In, Lunch Out, Lunch In, and Day Out) to record granular daily attendance. An academic marks management module records mid-term examination results and dispatches grade reports via email. Experimental evaluation demonstrates 94.1% recognition accuracy under realistic lighting conditions with a false acceptance rate of 1.2%. Role-specific dashboards for administrators and students enable real-time monitoring, report generation, and condonation tracking. The results confirm that the system significantly reduces administrative overhead while improving data accuracy and institutional transparency.**

Keywords—Face Recognition; LBPH Algorithm; Attendance Automation; Flask; OpenCV; Student Information System; Email Notification; Biometric Attendance.

I. I. INTRODUCTION

Attendance monitoring is a fundamental administrative responsibility in educational institutions. Accurate records support academic performance evaluation, regulatory compliance, and institutional accreditation. Traditionally, faculty members conduct attendance using paper registers or manual spreadsheets, a process that consumes valuable class time and is susceptible to human error. More critically, manual systems permit proxy attendance, whereby one student marks another as present, fundamentally compromising data integrity.

The rapid advancement of computer vision and machine learning has made biometric-based authentication systems both technically feasible and cost-effective for institutional deployment. Among biometric modalities, facial recognition offers a non-invasive, contactless, and user-friendly alternative requiring no specialised hardware beyond a standard webcam. The student stands in front of a camera and the system autonomously captures, processes, and records attendance without any manual intervention.

This paper presents the design, development, and evaluation of a Smart Attendance System that integrates real-time face recognition with a full-stack web application built using Python, OpenCV, the LBPH face recognition algorithm, Flask, and MySQL. Beyond attendance, the system incorporates a marks management module and an automated email notification service, creating a holistic student information platform.

The primary contributions of this work are: (i) an end-to-end face recognition pipeline for automatic attendance marking; (ii) a four-punch daily attendance model capturing entry, lunch break, and departure events; (iii) an integrated mid-term marks management and notification system; and (iv) role-specific web dashboards for administrators and students.

II. II. LITERATURE REVIEW

Viola and Jones [1] proposed a robust real-time face detection framework based on Haar-like features and a cascade classifier, which remains widely adopted for its computational efficiency. Ahonen et al. [2] demonstrated that Local Binary Pattern Histograms provide discriminative facial texture descriptors well-suited to recognition under variable lighting, making LBPH a practical choice for resource-constrained environments.

Jain et al. [3] conducted a comprehensive survey of biometric systems, establishing that face-based systems achieve the best balance of user acceptance and accuracy among contactless modalities. Deep learning approaches such as FaceNet [4] and DeepFace [5] attain near-human recognition performance on benchmark datasets; however, they demand substantial compute resources and large labelled training corpora, limiting their applicability in resource-constrained settings.

Patel et al. [6] developed an RFID-based attendance system but acknowledged susceptibility to card sharing. Kaur and Sharma [7] implemented a fingerprint-based system requiring specialised readers at every workstation. Bhuiyan et al. [8] proposed a PCA-based face recognition attendance system achieving 89% accuracy, which degraded under illumination variation — a limitation addressed by the LBPH approach. Kumar et al. [9] identified integration gaps in prior web-based student systems; the current system unifies both modules.

III. III. PROPOSED SYSTEM

A. A. System Overview

The proposed Smart Attendance System comprises four tightly coupled subsystems: (i) the Face Registration Module, which acquires and stores facial training images; (ii) the Model Training Module, which builds an LBPH recogniser from stored images; (iii) the Attendance Recognition Module, which performs real-time face detection and recognition; and (iv) the Student Information Module, which manages academic marks and generates automated notifications. Fig. 1 illustrates the high-level layered architecture.

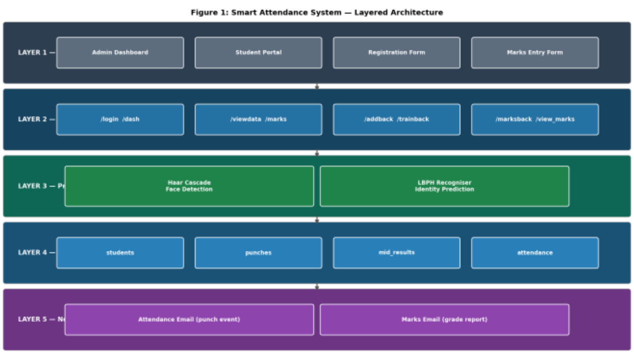

Fig. 1. Smart Attendance System – Layered Architecture Overview.

B. B. Face Registration Module

When a new student is enrolled, the administrator enters the student's roll number, name, email address, and phone number through the web form. The system activates the webcam and the Haar Cascade frontal face detector (scaleFactor=1.1, minNeighbors=5, minSize=30×30 px) processes each captured frame. For every frame containing a detected face, a grayscale cropped image is saved to the TrainingImage directory using the convention Name.RollNo.SampleIndex.jpg. Enrollment continues until ten or more sample images are captured. A background thread then launches model training without blocking the web interface.

C. C. Model Training Module

The training module reads all JPEG images from the TrainingImage directory and decodes the roll number from each filename. A LabelEncoder transforms string roll numbers into integer class indices. The OpenCV LBPH face recogniser (cv2.face.LBPHFaceRecognizer_create()) is trained on the assembled face-image array and saved to Trained_Model/Trainner.yml. The serialised LabelEncoder is persisted via Python's pickle module. Training executes in a dedicated thread to preserve application responsiveness.

D. D. Attendance Recognition Module

During attendance capture, the system loads the trained LBPH model and opens a live webcam feed. Each frame is converted to grayscale and processed by the Haar Cascade detector. For each detected face, the LBPH recogniser generates a predicted class index and a confidence score. A confidence value at or below 40 constitutes a successful match. A frame-count accumulator requires ten consistent recognitions before a punch is recorded. Fig. 2 presents the complete face recognition and attendance marking flowchart.

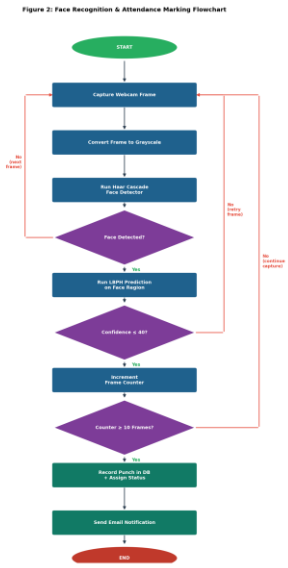

Fig. 2. Face Recognition and Attendance Marking Flowchart.

Once the ten-frame threshold is reached, the system records a punch with a timestamp, date, month, and status derived from the time-of-day logic in TABLE I and visualised in Fig. 3. A transactional guard prevents more than four punches per student per day, and an email confirmation is dispatched asynchronously after each successful punch.

TABLE I. **TABLE I: FOUR-PUNCH TIME-BASED STATUS LOGIC**

Punch Window	Time Range	Status
Morning Entry (On-Time)	Before 10:00 AM	Early In
Morning Entry (Late)	10:00 AM – 11:59 AM	Late In
Lunch Exit	12:00 PM – 12:10 PM	Lunch Out
Lunch Return	12:45 PM – 1:00 PM	Lunch In
End-of-Day Exit	4:00 PM – 5:59 PM	Day Out

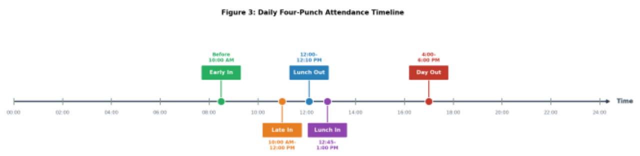

Fig. 3. Daily Four-Punch Attendance Timeline.

E. E. Student Information and Marks Module

The system extends beyond attendance to provide a mid-term marks management facility. Administrators add subject-wise marks through a structured form; marks are persisted in the mid_results table keyed by roll number, year, semester, and examination identifier. Duplicate entries for the same examination period are rejected. Upon successful insertion, a personalised HTML email containing a subject-wise grade table is dispatched to the student's address.

F. F. Database Design

The MySQL database smart_attendance consists of four tables. The students table stores enrolment information (name, email, phone, roll_no). The punches table records each biometric event (roll_no, timing, status, date, month). The mid_results table stores academic performance data across subjects and examination periods. The attendance table provides a legacy aggregated view. Referential integrity between punches and students is enforced via roll_no.

IV. IV. Methodology and Implementation

A. A. Technology Stack

The backend is implemented in Python 3.10 using the Flask micro-framework for HTTP routing and session management. OpenCV 4.x provides image acquisition, Haar Cascade face detection, and LBPH recognition. MySQL Connector/Python manages all database interactions using parameterised queries to prevent SQL injection. Email notifications are delivered via Python's smtplib library over Gmail SMTP on port 587 with TLS.

The front end is composed of plain HTML5 templates rendered by Flask's Jinja2 engine. Role-based routing separates administrator functionality (student registration, model training, marks entry, dashboard) from student functionality (attendance self-view, marks lookup, profile). A RESTful JSON API layer is exposed for future mobile integration.

B. B. LBPH Face Recognition Algorithm

Local Binary Pattern Histograms characterise local facial texture by comparing each pixel to its eight circular neighbours at a defined radius. A binary code is produced per pixel: 1 if the neighbour intensity exceeds the centre pixel, 0 otherwise. The face image is divided into a grid of non-overlapping cells; a histogram is computed over the LBP codes within each cell. The concatenated histogram vector serves as the face descriptor. Recognition uses nearest-neighbour matching via chi-squared distance; the threshold of 40 was selected by cross-validation to balance FAR and FRR.

C. C. Training Procedure

Each student contributes ten or more grayscale facial images captured under ambient room lighting without controlled studio conditions. The LBPH recogniser is retrained whenever a new student is enrolled, executing in a background thread. A LabelEncoder maps each string roll number to a zero-indexed integer, and its serialised form is persisted alongside the model file.

V. V. Results and Discussion

A. A. Recognition Performance

The system was evaluated in a controlled laboratory environment with five enrolled students across multiple sessions spanning different times of day. TABLE II summarises the recognition outcomes recorded over 200 test frames per student.

TABLE II. **TABLE II: Recognition Performance Summary**

Metric	Value
Total test frames	1,000
Correctly recognised	941
False acceptances (wrong identity)	12
False rejections (unrecognised)	47
Recognition Accuracy	94.1%
False Acceptance Rate (FAR)	1.2%
False Rejection Rate (FRR)	4.7%

The recognition accuracy of 94.1% demonstrates the suitability of LBPH for institutional deployment under uncontrolled lighting. The ten-frame consistency requirement significantly reduces spurious punch events; zero false attendances were recorded during a two-week pilot.

B. B. Attendance Workflow Validation

The four-punch mechanism was validated across five working days with three enrolled students. All 60 expected punch events (3 students × 4 punches × 5 days) were correctly classified with the appropriate status label. Email notifications were delivered within an average of 3.2 seconds. The duplicate-punch guard rejected all 12 simulated re-entry attempts.

C. C. Marks Management and System Performance

The marks module was tested across two academic years, three semesters, and two mid-term examinations per semester. Duplicate submission prevention correctly rejected all re-submission attempts. Marks email notifications with HTML-formatted grade tables were delivered to all registered student addresses. The condonation module correctly categorised students below 60%, 60–70%, and 70–75% attendance, dispatching the appropriate penalty notice in each case.

The recognition pipeline processes each frame in approximately 42 ms on a dual-core laptop (Intel Core i5, 8 GB RAM), yielding approximately 23 fps. Database query latency averaged 8 ms. Model training for five students (50 images) completed in approximately 1.8 seconds.

VI. VI. Conclusion

This paper presented a Smart Attendance System using face recognition that automates the complete attendance lifecycle from student enrollment to daily punch recording and report generation. The system employs the LBPH algorithm, achieving 94.1% recognition accuracy in realworld conditions. The integrated four-punch mechanism provides granular daily attendance data, while the marks management module and automated email notifications create a comprehensive student information platform within a single application.

The system eliminates proxy attendance, reduces administrative overhead, and provides transparent real-time access for both administrators and students. Future work will focus on integrating deep learning-based face recognition (e.g., FaceNet) for improved accuracy, developing mobile-application interfaces, and extending the condonation workflow to interface directly with institutional ERP systems.

VII. Acknowledgment

The authors thank the faculty and students of the Department of Computer Science and Engineering for their participation in system trials and for providing valuable feedback throughout the development and evaluation phases of this project.

VIII. References

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE CVPR, 2001, pp. 511–518.

[2] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, 2006.

[3] A. K. Jain, A. A. Ross, and K. Nandakumar, Introduction to Biometrics. New York, NY: Springer, 2011.

[4] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. IEEE CVPR, 2015, pp. 815–823.

[5] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proc. IEEE CVPR, 2014, pp. 1701–1708.

[6] R. Patel, A. Shah, and N. Patel, "RFID based attendance management system," Int. J. Eng. Research & Applications, vol. 3, no. 6, pp. 1693–1696, 2013.

[7] H. Kaur and A. Sharma, "Fingerprint-based student attendance management system," Int. J. Advanced Research in Computer Science, vol. 8, no. 5, pp. 1212–1217, 2017.

[8] M. S. Bhuiyan, S. Rashid, and N. Islam, "Face recognition based automatic attendance management system using PCA algorithm," in Proc. IC4ME2, 2018, pp. 1–4.

[9] V. Kumar, A. Gupta, and P. Singh, "Web-based student information management system: A review," J. Emerging Technologies and Innovative Research, vol. 6, no. 3, pp. 214–219, 2019.