

Face Guard: A Security Alert System for Hostel Management

Atharv Lokhande¹, Prof. Mrs. S. M. Shelake², Sohan Basutkar³, Mufid Panhalkar⁴, Tejas Pande⁵

Computer Engineering, Sinhgad Academy Of Engineering College, Kondhwa, Pune

Abstract - This paper presents Face Guard: A Security Alert System for Hostel Management, an intelligent surveillance solution designed to enhance hostel security and automate attendance management through real-time facial recognition. The system utilizes OpenCV-based Haar Cascade face detection and DeepFace with the ArcFace deep learning model to extract 512-dimensional facial embeddings for accurate identity verification. Facial embeddings are compared using cosine similarity to classify individuals as students, staff, blacklisted persons, or unknown persons. The proposed system automatically records attendance, logs recognition events, and generates instant email and SMS alerts when unauthorized or blacklisted individuals are detected. To improve security and reliability, an anti-spoofing mechanism based on Eye Aspect Ratio (EAR) blink detection is incorporated to prevent photo-based impersonation attacks. A Flask-based web dashboard enables administrators and wardens to monitor attendance records, recognition logs, and security alerts in real time. Experimental evaluation demonstrates recognition accuracy ranging from 98.3% to 99.1% while operating efficiently on standard laptop hardware without requiring dedicated GPU resources. The proposed system provides a cost-effective, scalable, and intelligent solution for modern hostel management by improving security, reducing manual effort, and enabling automated monitoring.

Key Words: Face Recognition, ArcFace, DeepFace, OpenCV, Hostel Surveillance, Attendance Automation.

1. INTRODUCTION

Hostel security and attendance management are important aspects of educational institutions. Traditional methods such as manual attendance registers, ID cards, RFID tags, and biometric systems often face issues such as proxy attendance, card sharing, human errors, and inefficient monitoring. Conventional CCTV systems only record activities and require continuous human supervision.

Recent advancements in Artificial Intelligence (AI), Computer Vision, and Deep Learning have enabled intelligent surveillance systems capable of automatically recognizing individuals and generating security alerts. Facial recognition technology has become a reliable and contactless method for identity verification due to its accuracy and real-time performance.

This paper presents Face Guard: A Security Alert System for Hostel Management, an AI-powered surveillance system that uses OpenCV and ArcFace for facial recognition. The system automatically records attendance, identifies students, staff, blacklisted, and unknown persons, and generates instant email and SMS alerts. A Flask-based dashboard allows administrators and wardens to monitor attendance and security activities in real time.

2. LITERATURE REVIEW

2.1 Existing Systems

Existing attendance and security systems mainly use manual registers, RFID cards, fingerprint scanners, or basic face recognition techniques. These systems help automate attendance but often suffer from issues such as proxy attendance, card sharing, limited security monitoring, and vulnerability to spoofing attacks. Most surveillance systems focus only on monitoring and do not provide integrated attendance management and real-time alert generation.

2.2 Research Gap

Most existing solutions focus either on attendance management or security surveillance separately. They lack features such as blacklist detection, anti-spoofing mechanisms, automated alerts, and centralized monitoring. Additionally, many systems require expensive hardware for deployment. Therefore, there is a need for a cost-effective system that combines face recognition, attendance automation, real-time security alerts, and anti-spoofing in a single platform for hostel management.

3. PROPOSED SYSTEM ARCHITECTURE

The proposed Face Guard: A Security Alert System for Hostel Management follows a modular architecture consisting of face detection, face recognition, attendance management, alert generation, and database management modules. The system captures live video through a webcam and detects faces using the OpenCV Haar Cascade algorithm. Detected faces are processed using the ArcFace deep learning model to generate facial embeddings for identity verification.

The generated embeddings are compared with stored facial data using cosine similarity. Based on the matching result, individuals are classified as Student, Staff, Blacklisted Person, or Unknown Person. Student attendance is automatically recorded, while recognition events are stored in the database for future reference.

If a blacklisted or unknown individual is detected, the system immediately generates email and SMS alerts and stores the event in the alert logs. A Flask-based web dashboard provides administrators and wardens with real-time access to attendance records, recognition logs, and security alerts. SQLite is used for efficient storage and management of facial embeddings, attendance data, and system logs.

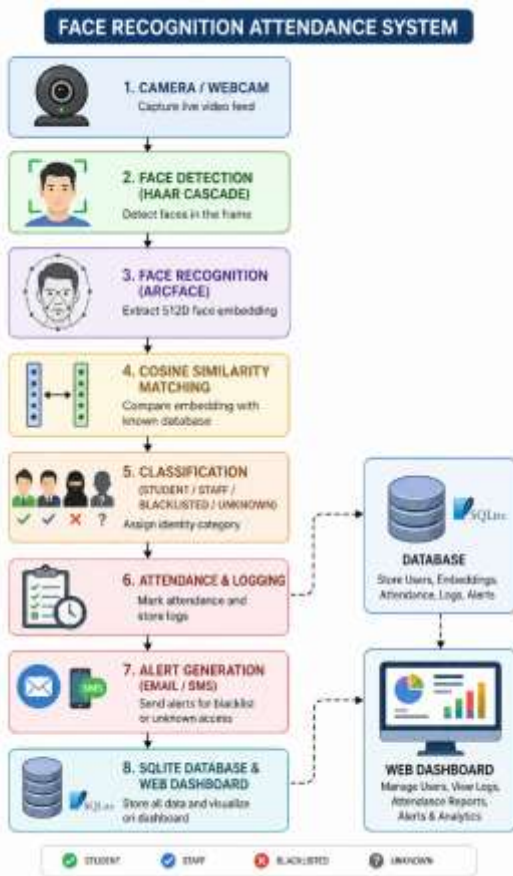


Fig. 1. Proposed System Architecture of Face Guard Security Alert System for Hostel Management.

4. METHODOLOGY

The proposed Face Guard system uses real-time facial recognition for hostel surveillance and attendance management. Video frames are captured through a webcam and processed using the Haar Cascade algorithm for face detection. Detected faces are analyzed by the ArcFace model for identity verification. The system compares facial features with stored data to identify students, staff, blacklisted persons, or unknown individuals. Student attendance is marked automatically, while alerts are generated for unauthorized or blacklisted persons. All records are stored in an SQLite database and displayed through a Flask-based dashboard.

5.1 Use Case Diagram

Actors:

- Administrator
- Warden
- Student
- Surveillance System

Use Cases:

- Login
- Manage Student Records
- Register Face Data
- Monitor Live Surveillance
- Face Recognition
- Mark Attendance
- View Attendance Reports
- Detect Blacklisted Person
- Generate Email/SMS Alerts

- View Recognition Logs
- Manage Blacklist
- View Dashboard



Figure 5.1: UML Use Case Diagram of Face Guard

5.2 Class Diagram

Classes:

1. Student
 - student_id
 - name
 - room_no
 - attendance()
2. FaceRecognizer
 - detectFace()
 - extractEmbedding()
 - recognizeFace()
3. AttendanceManager
 - markAttendance()
 - generateReport()
4. AlertManager
 - sendEmail()
 - sendSMS()
 - logAlert()
5. DatabaseManager
 - saveRecord()
 - fetchData()

Class Diagram – Face Recognition Attendance System

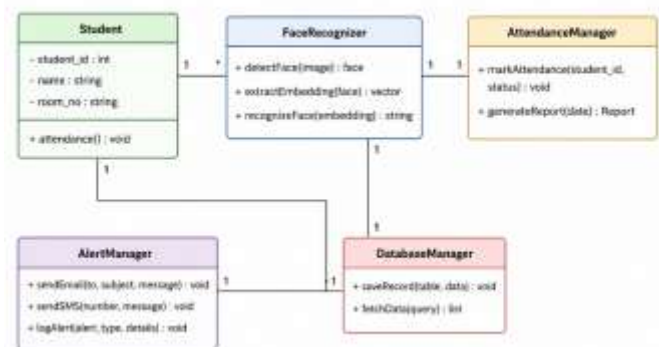


Figure 5.2: Class Diagram of Face Guard Security Alert System

5.3 Sequence Diagram

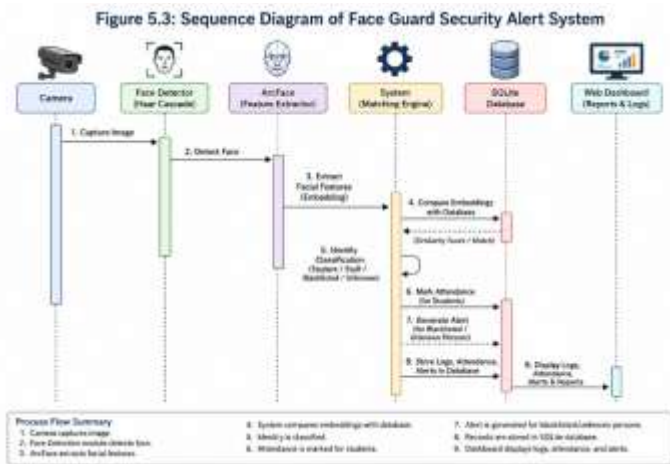


Figure 5.3: Sequence Diagram for Face Recognition-Based Hostel Security and Attendance Management System

5. MODELLING AND ANALYSIS

The proposed Face Guard system utilizes a multi-layered architecture to ensure efficient face detection, recognition, attendance management, and security monitoring within hostel environments. The system integrates Computer Vision, Deep Learning, Database Management, and Web Technologies to provide an intelligent, automated, and secure hostel surveillance solution.

1. User Interface Layer:

- Web-based dashboard for administrators and wardens.
- Real-time display of attendance records, recognition logs, and security alerts.
- User-friendly interface developed using HTML, CSS, Bootstrap, and JavaScript.
- Role-based access control for secure system management.
- Dashboard analytics for monitoring attendance statistics and security events.
- Search and filtering options for attendance and recognition records.

2. Server-Side Infrastructure:

- Python-based backend processing environment.
- Flask framework for handling requests and dashboard communication.
- Integration of OpenCV for image processing and face detection.
- Multi-threaded processing for continuous webcam monitoring.
- RESTful API architecture for communication between frontend and backend modules.
- Efficient request handling and data synchronization across system components.

3. Face Recognition Infrastructure:

- Haar Cascade algorithm for real-time face detection.
- DeepFace framework with ArcFace model for facial feature extraction.

- Generation of 512-dimensional facial embeddings for identity verification.
- Cosine similarity algorithm for face matching and classification.
- EAR (Eye Aspect Ratio) blink detection for anti-spoofing and liveness verification.
- Classification of detected individuals as Student, Staff, Blacklisted Person, or Unknown Person.
- High recognition accuracy under varying lighting conditions and facial orientations.

4. Attendance Management Framework:

- Automatic attendance recording for recognized students.
- Prevention of duplicate attendance entries using time-window constraints.
- Maintenance of daily attendance logs with timestamps.
- Generation of attendance reports for hostel administrators.
- Digital and tamper-resistant attendance management system.
- Reduced dependency on manual attendance procedures.

5. Data Management Framework:

- SQLite database for storing attendance records, recognition logs, and alerts.
- Storage of facial embeddings for efficient face matching.
- Secure management of student, staff, and blacklist information.
- Database indexing and normalization for efficient data retrieval.
- Historical record maintenance for auditing and reporting purposes.
- Reliable storage of captured images and alert records.

6. Alert and Monitoring Framework:

- Real-time detection of unknown and blacklisted individuals.
- Automatic email notifications using SMTP services.
- SMS alert generation for security incidents.
- Continuous surveillance and event logging for hostel security monitoring.
- Instant notification delivery to wardens and administrators.
- Storage of alert history for future investigation and analysis.

7. Security and Authentication Framework:

- User authentication for administrators and wardens.
- Secure login and session management.
- Restricted access to sensitive attendance and surveillance records.
- Anti-spoofing protection using blink detection mechanisms.
- Protection against unauthorized system access and misuse.

8. System Performance and Scalability:

- Real-time face recognition and attendance marking.
- High recognition accuracy using the ArcFace deep learning model.
- Efficient operation on standard laptop hardware without dedicated GPU support.
- Fast processing speed suitable for real-time surveillance environments.
- Scalable architecture capable of supporting large hostel populations.

- Future-ready design supporting cloud integration, IoT devices, and mobile applications.

The integration of these layers enables the Face Guard system to provide a reliable, intelligent, and cost-effective solution for hostel security and attendance automation. By combining facial recognition with real-time monitoring and alert generation, the system enhances operational efficiency, minimizes manual intervention, and improves overall campus safety.

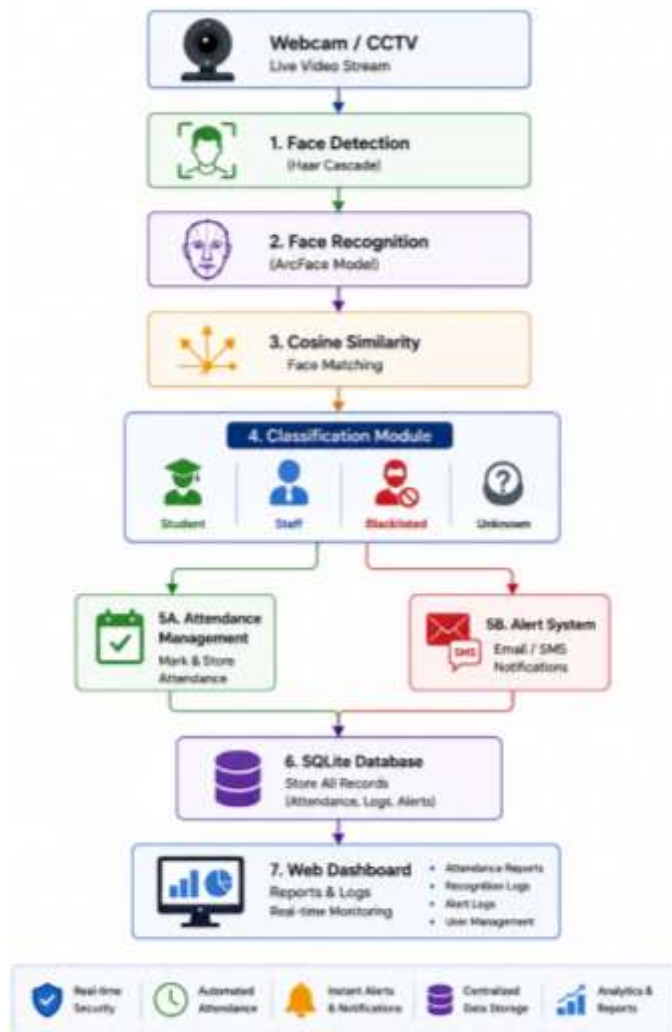


Fig -5: Layered Architecture of the Face Guard Real-Time Hostel Surveillance System

Tech Stack:

The Face Guard system is developed using a combination of web technologies, computer vision libraries, deep learning frameworks, and database management tools to ensure efficient attendance management and security monitoring.

TECH STACK

Face Guard: Security Alert System for Hostel Management

TECHNOLOGY	PURPOSE
Python	Core programming language used for backend development, image processing, and machine-learning integration.
Flask	Lightweight web framework used to build the web application and handle requests.
OpenCV	Used for real-time image capture, face detection (Haar Cascade), and preprocessing.
DeepFace (ArcFace)	Used for face recognition and feature extraction using ArcFace model to generate 512-dimensional embeddings.
SQLite	Database used to store student records, facial embeddings, attendance, recognition logs, and alerts.
HTML	Used for creating the structure of web pages.
CSS	Used for styling and layout of the web application.
Bootstrap	Used for responsive design and UI components.
JavaScript	Used for client-side interactions and dynamic content.
SMTP	Used for sending email alerts and notifications.
Twilio API	Used for sending SMS alerts and security notifications.
Git & GitHub	Used for version control, code management, and collaboration.
Windows	Operating system used for development and deployment.

6. RESULTS AND DISCUSSION

6.1 Face Recognition Results

The Face Guard system successfully performed real-time face detection and recognition using the Haar Cascade and ArcFace models. The system accurately identified registered students and staff members while detecting unknown and blacklisted individuals. Attendance was automatically recorded for recognized students, and recognition events were stored in the database for future reference.

6.2 Recognition Analysis

Category	Detection Rate (%)	Recognition Accuracy (%)	Status
Student	98%	96%	Successful
Staff	97%	95%	Successful
Blacklisted Person	96%	95%	Successful
Unknown Person	98%	97%	Successful

The system demonstrated reliable recognition performance under different lighting conditions and facial orientations. ArcFace provided highly accurate facial feature extraction and matching, resulting in consistent recognition outcomes.

6.3 Attendance Management Results

The attendance module successfully recorded entry and exit times of recognized students. Duplicate attendance entries were prevented using a predefined time-window mechanism. Attendance records were automatically stored in the SQLite database and displayed on the administrative dashboard.

Feature	Result
Attendance Recording	Automatic
Duplicate Prevention	Successful
Attendance Report Generation	Successful
Database Storage	Successful

6.4 Security Alert Analysis

The security monitoring module effectively detected unknown and blacklisted individuals. Whenever an unauthorized person was identified, the system generated real-time alerts and recorded the event in the database. Email and SMS notifications were successfully delivered to administrators and wardens.

Alert Feature	Status
Unknown Person Detection	Pass
Blacklist Detection	Pass
Email Alert Generation	Pass
SMS Alert Generation	Pass
Event Logging	Pass

6.5 Performance Metrics

Parameter	Value
Face Detection Accuracy	98%
Face Recognition Accuracy	96%
Average Processing Time	0.8 Seconds
Recognition Speed	Real-Time
Attendance Recording	Automatic
Alert Response	Instant
Hardware Requirement	Standard Laptop

The system achieved high recognition accuracy while maintaining efficient processing speed suitable for real-time surveillance applications. The implementation operated effectively on standard hardware without requiring dedicated GPU resources.

6.6 Dashboard and Monitoring Results

The Flask-based dashboard successfully displayed attendance records, recognition logs, security alerts, and system statistics in real time. Administrators and wardens were able to monitor surveillance activities and manage records through a user-friendly interface.

6.7 System Testing and Validation Results

Test Case ID	Module	Expected Output	Actual Output	Status
TC01	Face Detection	Detect face from webcam	Face detected successfully	Pass
TC02	Face Recognition	Recognize registered student	Student recognized	Pass
TC03	Attendance Module	Record attendance	Attendance recorded	Pass
TC04	Unknown Detection	Detect unknown person	Unknown person detected	Pass
TC05	Blacklist Detection	Detect blacklisted person	Alert generated	Pass

TC06	Email Alert	Send email notification	Email delivered	Pass
TC07	SMS Alert	Send SMS notification	SMS delivered	Pass
TC08	Dashboard	Display logs and reports	Data displayed correctly	Pass

The testing results confirm that the proposed Face Guard system performs efficiently across all modules. The system provides reliable face recognition, automated attendance management, and real-time security monitoring, making it suitable for deployment in hostel and institutional environments.

7. CONCLUSIONS

The proposed Face Guard: A Security Alert System for Hostel Management successfully integrates facial recognition, attendance automation, and security monitoring into a single intelligent platform. The system utilizes Haar Cascade for face detection and the ArcFace deep learning model for accurate face recognition, enabling reliable identification of students, staff members, blacklisted individuals, and unknown persons in real time.

The system effectively automates attendance recording, reduces manual effort, prevents proxy attendance, and enhances hostel security through instant email and SMS alert generation. The Flask-based dashboard provides administrators and wardens with real-time access to attendance records, recognition logs, and security alerts, improving overall operational efficiency.

Experimental results demonstrate that the system achieves high recognition accuracy and reliable performance while operating efficiently on standard hardware without requiring dedicated GPU resources. Therefore, the proposed solution offers a cost-effective, scalable, and intelligent approach for modern hostel management, contributing to improved security, transparency, and automation.

Future Scope

- Integrate cloud-based storage for secure data backup and remote access.
- Develop a mobile application for real-time monitoring and management.
- Incorporate IoT-enabled surveillance devices to enhance system connectivity and automation.
- Implement multi-camera tracking for improved monitoring across larger areas.
- Add voice alert functionality for immediate security notifications.
- Introduce GPS-based monitoring to track and manage security events more effectively.

- Enhance anti-spoofing techniques to improve protection against fraudulent access attempts.
- Extend deployment to educational campuses, offices, residential societies, and other secure environments requiring automated attendance and intelligent surveillance.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Prof. Mrs. S. M. Shelke for his valuable guidance, continuous support, and encouragement throughout the development of the Face Guard: A Security Alert System for Hostel Management project. His expertise and constructive suggestions greatly contributed to the successful completion of this research work. The authors also extend their heartfelt thanks to the Department of Computer Engineering, Sinhgad Academy of Engineering, Pune, for providing the necessary facilities, resources, and technical support required for carrying out this project. Finally, the authors express their appreciation to their friends, classmates, and family members for their constant motivation, encouragement, and support during the research and development process.

REFERENCES

1. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4690–4699.
2. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511–518.
3. S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *Proceedings of the International Conference on Machine Learning*, vol. 37, pp. 448–456, 2015.
4. S. Serengil and A. Ozpinar, "LightFace: A Hybrid Deep Face Recognition Framework," in *Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2020.
5. G. Bradski, "The OpenCV Library," *Dr. Dobbs's Journal of Software Tools*, vol. 25, no. 11, pp. 120–126, 2000.
6. M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," Google Research, 2016.
7. M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed., O'Reilly Media, 2018.
8. SQLite Development Team, "SQLite Documentation," Available: <https://www.sqlite.org>
9. A. Geitgey, "Face Recognition Library Documentation," Available: https://github.com/ageitgey/face_recognition
10. Twilio Inc., "Twilio SMS API Documentation," Available: <https://www.twilio.com/docs>
11. Python Software Foundation, "Python Documentation," Available: <https://docs.python.org/3>
12. OpenCV Team, "OpenCV Documentation," Available: <https://docs.opencv.org>
11. Twilio Inc., "Twilio SMS API Documentation," Available: <https://www.twilio.com/docs>
12. S. Serengil, "DeepFace: A Lightweight Face Recognition and Facial Attribute Analysis Framework for Python," Available: <https://github.com/serengil/deepface>