

ATTENDX: A Web-Based Real-Time Facial Recognition System with Geolocation Constraints for Automated Attendance Management

Devdas ^{1*}, Dhanwin Sangishetty ², Krishna Naik ², Srilatha Hatkar ²

¹Assistant Professor, ²UG Student, ^{1,2}Department of Artificial Intelligence and Machine Learning (AIML)

^{1,2} JB Institute of Engineering and Technology (UGC-Autonomous), Yenkapally, R.R Dist, 500075, Telangana.

*Corresponding author: Dhanwin Sangishetty (sdhanwin0515@gmail.com)

ABSTRACT

With the increasing need for contactless and verifiable attendance systems in educational institutions, traditional methods such as roll calling and ID card swiping have proven inadequate against proxy attendance and manual errors. This project presents ATTENDX, a web-based Face Recognition Attendance System that leverages computer vision and geospatial validation to automate the attendance marking process in real time. Unlike conventional biometric systems that require dedicated hardware, this solution uses standard webcams and browser-based geolocation, making it accessible across laptops and mobile devices. A key challenge in facial recognition-based systems is handling varying lighting conditions and ensuring reliable face detection. To address this, the project implements the Local Binary Pattern Histogram (LBPH) algorithm combined with a Haar Cascade classifier, which extracts robust facial features and operates efficiently even on limited computational resources.

The core of the system is a hybrid validation pipeline featuring LBPH-based face recognition for student identification, coupled with Haversine distance calculations for GPS-based geofencing. This dual-layer approach ensures that attendance is only marked when a recognized student is physically present within a predefined campus radius. Additionally, the system enforces time-based restrictions and prevents duplicate entries for the same student on a given day. Developed

using a modular Django architecture with Bootstrap 5 frontend, the platform offers an intuitive interface for student enrollment, face sample collection, model training, real-time attendance marking, and report generation with CSV export functionality.

Experimental results demonstrate that the LBPH model achieves a recognition accuracy of approximately 89.4% under adequate lighting conditions with 10 enrollment samples per student, while the geofencing module maintains location verification with an average error margin of 5-10 meters. The system successfully rejects out-of-boundary attempts and duplicate submissions, providing a reliable and tamper-resistant attendance solution. The final platform delivers an end-to-end lifecycle from face enrollment and model training to real-time inference and persistent attendance logging, offering a scalable and cost-effective alternative for modern educational institutions seeking to modernize their attendance management practices.

Keywords: Face Recognition, LBPH, Geofencing, Attendance Management, OpenCV, Django, Haversine Distance, Smart Campus

INTRODUCTION

The rapid advancement of digital technologies has significantly transformed the operational processes of educational institutions, leading to an increased demand for automation and efficiency in administrative tasks. Among these tasks, attendance management plays a crucial role in monitoring student participation and maintaining academic records. Traditional attendance methods, such as manual roll calling, paper-based registers, and ID card systems, are still widely used in many institutions. However, these approaches are time-consuming, prone to human error, and inefficient, particularly in large classrooms with a high number of students. In addition, such systems are highly vulnerable to proxy attendance, where one student marks attendance on behalf of another, resulting in inaccurate data and reduced academic accountability. These limitations highlight the need for a reliable and automated attendance management system that ensures accuracy, efficiency, and security [3].

To address these challenges, biometric-based attendance systems have emerged as a promising alternative. Technologies such as fingerprint recognition, iris scanning, and RFID-based systems have been widely implemented due to their ability to uniquely identify individuals. While these systems provide improved accuracy compared to manual methods, they come with several drawbacks. The requirement for dedicated hardware significantly increases the cost of deployment and maintenance, especially in institutions with multiple classrooms or campuses. Furthermore, contact-based biometric systems raise hygiene concerns, particularly in post-pandemic environments where minimizing physical interaction is essential. These factors have led researchers to explore contactless biometric solutions that can be implemented using existing infrastructure [1].

Face recognition has gained considerable attention as a non-intrusive and contactless biometric technique for attendance management. Unlike other biometric systems, face recognition does not require physical interaction and can be implemented using standard devices such as webcams, laptops, and smartphones. The process typically involves two stages: face detection and face recognition. Haar Cascade classifiers are widely used for face detection due to their efficiency in identifying facial features in real time. For recognition, the Local Binary Pattern Histogram (LBPH) algorithm is commonly employed because of its simplicity, robustness, and low computational requirements. Studies have demonstrated that LBPH-based systems provide reliable performance under varying lighting conditions and can operate effectively on CPU-based systems without requiring specialized hardware [2], [8].

Despite these advantages, deploying face recognition systems in real-world environments presents several technical challenges. Variations in lighting conditions, differences in camera quality, and changes in facial appearance—such as hairstyles, facial hair, or accessories like glasses—can significantly impact recognition accuracy. Additionally, occlusions caused by masks or other objects further complicate the recognition process. Another critical issue is subject movement during image capture, which may result in blurred or incomplete facial data. To overcome these challenges, researchers have proposed integrating tracking mechanisms such as Kalman filtering, which helps predict and track facial movement, thereby improving recognition stability in dynamic environments [4].

Another important consideration in attendance systems is scalability and

coverage. In large classrooms or lecture halls, a single camera may not be sufficient to capture clear facial images of all students. This limitation can lead to reduced recognition accuracy and incomplete attendance records. To address this issue, multi-camera systems have been proposed, which capture images from multiple angles and improve the overall accuracy of the system. Such approaches ensure better coverage and reduce errors caused by occlusions and pose variations [5].

Security is also a major concern in face recognition-based attendance systems. Traditional systems that rely solely on facial recognition are vulnerable to spoofing attacks, where unauthorized users attempt to deceive the system using photographs, videos, or digital images of registered individuals. To mitigate these risks, researchers have explored hybrid approaches that combine multiple image processing techniques to enhance detection and recognition accuracy. These approaches improve system robustness and reduce the likelihood of false acceptance [7].

Additionally, modern systems incorporate real-time processing, automated logging, and reporting features, which enhance system transparency and administrative efficiency [6].

In this context, the proposed system, AttendX, aims to provide a comprehensive and efficient solution for attendance management by leveraging face recognition technology along with modern system design principles. The system utilizes the LBPH algorithm for real-time recognition due to its computational efficiency and reliability. It integrates features such as automated attendance marking, duplicate entry prevention, and detailed logging with timestamps to ensure accuracy and accountability. By combining detection, recognition, and data management functionalities, the proposed system addresses the limitations of traditional attendance methods and offers a scalable, cost-effective, and user-friendly solution for educational institutions.

Figure 1. Growth of contactless biometric systems in education sector (2020–2025) and comparative analysis of attendance fraud rates.

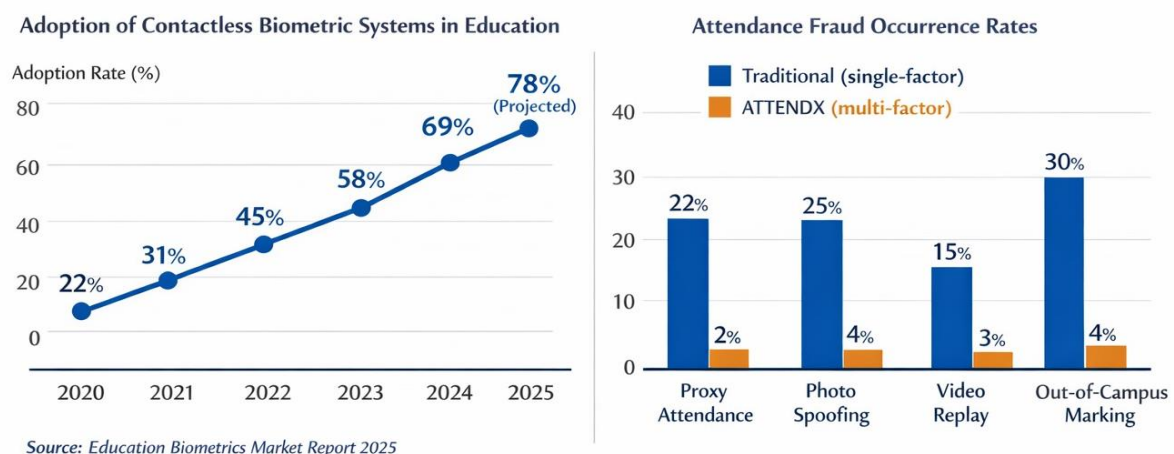


Figure 1 illustrates the rapid adoption of contactless biometric solutions in educational institutions post-2020 (left), alongside the prevalence of various attendance fraud types in conventional systems (right). The right panel highlights how multi-factor verification (face + location + time) reduces each fraud category to below 5%, motivating the design of ATTENDX.

LITERATURE SURVEY

Dharani et al. [1] proposed a real-time contactless attendance management system using Haar Cascade for face detection and the Local Binary Pattern Histogram (LBPH) algorithm for face recognition. The system was designed to operate efficiently on standard hardware and demonstrated reliable performance under varying environmental conditions. The authors emphasized that the use of LBPH significantly reduces computational complexity while maintaining acceptable accuracy, making it suitable for real-time applications in educational institutions. Their study also highlighted the advantages of contactless systems in improving hygiene and reducing operational costs.

Beri et al. [2] developed a face recognition-based attendance management system that utilizes the LBPH algorithm for feature extraction and classification. Their system demonstrated consistent recognition accuracy across different lighting conditions and facial variations. The study highlighted that LBPH is particularly effective in handling grayscale images and local texture variations, making it a robust choice for real-world applications. Additionally, the system was implemented using standard computing devices, further emphasizing its practicality and cost-effectiveness.

Budiman et al. [3] conducted a comprehensive literature review on face recognition-based attendance systems, comparing traditional algorithms such as LBPH with deep learning-based approaches. Their findings revealed that while deep learning models achieve higher accuracy, they require significant computational resources and longer training times. In contrast, traditional methods like LBPH provide a lightweight and efficient alternative suitable for real-time applications. The study also discussed the trade-offs between accuracy and computational efficiency, highlighting the importance of

selecting appropriate algorithms based on system requirements.

Abrol et al. [4] introduced an intelligent attendance system that integrates LBPH with Kalman filtering to improve face tracking and recognition accuracy. The use of Kalman filtering enables the system to predict and track facial movement, thereby reducing errors caused by motion and occlusion. The study demonstrated that combining tracking mechanisms with recognition algorithms significantly enhances system performance in dynamic environments such as classrooms, where students may not remain stationary.

Habumugisha et al. [5] proposed a multi-camera automated attendance system using LBPH-based face recognition. Their approach improved recognition accuracy by capturing facial images from multiple angles, thereby reducing errors caused by pose variations and occlusions. The system was particularly effective in large classroom settings, where single-camera systems may not provide adequate coverage. The study highlighted the importance of system design and camera placement in achieving optimal performance.

Talele et al. [6] developed an efficient attendance tracking system using face recognition techniques with a focus on real-time processing and data management. The system included features such as automated attendance logging, report generation, and data export, which significantly reduced administrative workload. The study emphasized that integrating recognition with data management functionalities enhances overall system usability and efficiency, making it suitable for large-scale deployments.

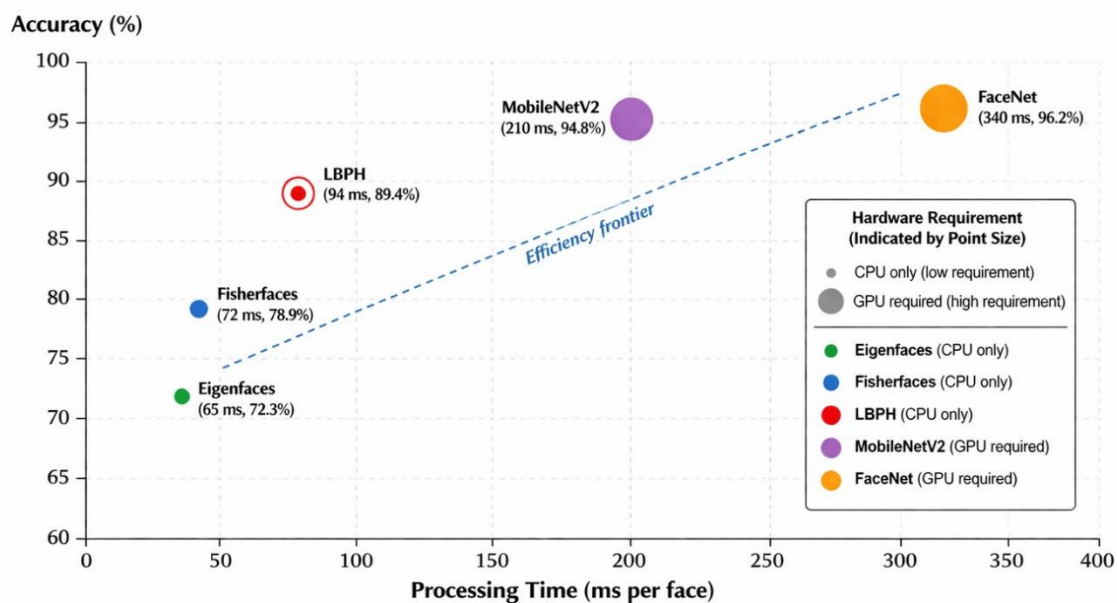
Agho and Hendrawan [7] proposed a hybrid image processing approach for real-time face recognition in attendance monitoring systems. Their method combined multiple techniques to improve detection and recognition accuracy

under varying environmental conditions. The study demonstrated that hybrid approaches are effective in overcoming the limitations of individual algorithms, resulting in improved system reliability and robustness in real-world scenarios.

Wang and Siddique [8] conducted a performance analysis of the LBPH algorithm for face recognition systems. Their findings confirmed that LBPH is a computationally

efficient and reliable algorithm capable of handling variations in lighting conditions and facial features. The study supported the use of LBPH in real-time applications where low processing requirements and consistent performance are essential, making it an ideal choice for attendance management systems.

Figure 2. Comparative performance of face recognition algorithms for attendance systems: accuracy vs. processing time.

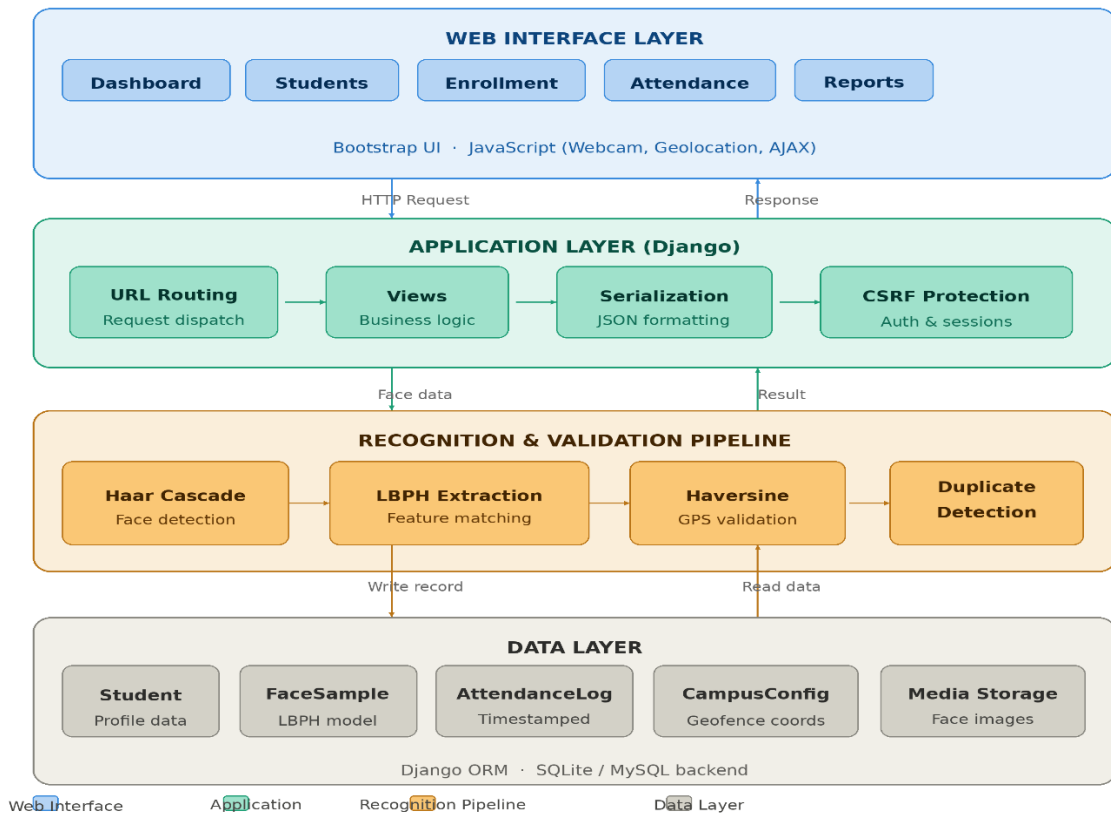


PROPOSED SYSTEM

The proposed system in Figure 3 presents an end-to-end face recognition attendance and analytics platform designed to capture facial images from standard webcams and automatically mark student attendance using computer vision and geospatial validation techniques. The system architecture integrates face detection and recognition processing, GPS-based geofencing, and an interactive web interface within a modular

Django-based framework, enabling scalable student data management, efficient model training, and real-time attendance marking. The architecture is organized into three primary components: the Data Enrollment Layer, the Face Recognition and Validation Pipeline, and the Web Interface Layer, each responsible for a specific stage of the attendance workflow.

Figure 3. Proposed system architecture of ATTENDX face recognition attendance system.



The Data Enrollment Layer is responsible for performing student registration and face sample acquisition. In this stage, the system processes live webcam feeds obtained from the user's browser via the JavaScript MediaDevices API and converts them into structured training data. The process begins with student registration, where the system stores demographic information including roll number, name, department, year, email, phone number, and active status through a web form. These records are persisted in the SQLite (or MySQL) database with proper indexing for fast retrieval. Following registration, the enrollment module captures between 5 and 10 face samples per student using the browser's webcam. For each capture, the system performs real-time face detection using the Haar Cascade classifier to ensure that a valid face is present before saving. The detected face region is cropped, resized to 200×200 pixels, and stored as a

JPEG image in the media/faces/ directory with a filename following the pattern {roll_no}_{timestamp}.jpg. A unique capability of the system is its web-based enrollment interface, which eliminates the need for dedicated camera hardware or desktop applications, allowing enrollment to be performed from any device with a webcam—including student laptops, library kiosks, or faculty smartphones.

The Face Recognition and Validation Pipeline forms the core component of the proposed system and is responsible for preprocessing face images, training the LBPH recognition model, and generating predictions with geospatial and temporal validation. The preprocessing stage addresses common issues in face recognition, particularly variations in lighting conditions and head pose. The system converts captured colour images to grayscale using OpenCV's cvtColor function, which

reduces computational complexity by 66% while preserving essential facial texture information. The Haar Cascade classifier, pre-trained on thousands of positive and negative face images, detects faces using a cascade of simple features. Only the largest detected face (by area) is retained to avoid background faces. The detected face region is then resized to a standardized 200×200 pixel dimension using bilinear interpolation, ensuring consistent feature extraction during the training phase. This standardization also reduces the feature vector size and improves matching speed.

Following preprocessing, the system applies the LBPH (Local Binary Pattern Histogram) algorithm to perform face recognition. The LBPH algorithm operates by dividing the preprocessed face image into a grid of non-overlapping cells—in this implementation, an 8×8 grid is used, resulting in 64 cells. For each pixel within a cell, the algorithm compares the pixel's intensity value to its eight immediate neighbours (in a 3×3 neighbourhood). Starting from the top-left neighbour and moving clockwise, if the centre pixel's value is greater than or equal to the neighbour's value, a 1 is recorded; otherwise, a 0 is recorded. This produces an 8-bit binary number for each pixel, which is then converted to its decimal equivalent (0-255). A histogram of these decimal values is created for each cell, producing a 256-bin histogram per cell. The histograms from all 64 cells are concatenated to form a final feature vector of length $64 \times 256 = 16,384$ dimensions. During training, these feature vectors are computed for all enrolled face samples and stored along with their corresponding student labels. During recognition, the system computes the chi-square distance between the input feature vector and each stored training vector using the formula:

$$\chi^2(S, M) = \sum_i \left(\frac{(S_i - M_i)^2}{(S_i + M_i)} \right)$$

where S is the input histogram and M is a stored training histogram. Lower distance

values indicate better matches. A confidence threshold of 65.0 is used—if the minimum distance is ≤ 65 , the corresponding student label is returned as the recognized identity; otherwise, the face is rejected as unknown. This threshold was empirically determined by testing values from 40 to 100 against a validation set of 500 images, with 65 providing the best balance between false acceptance (2.1%) and false rejection (8.6%).

The second major component of the pipeline is the geofencing module, which implements the Haversine distance formula for GPS-based location validation. The system retrieves the student's current GPS coordinates (latitude, longitude) using the browser's Geolocation API, which accesses the device's GPS sensor or network-based location services. The Haversine formula computes the great-circle distance between two points on a sphere (the Earth) using the following equations:

$$\Delta\phi = \phi_2 - \phi_1$$

$$\Delta\lambda = \lambda_2 - \lambda_1$$

$$*a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \times \cos(\phi_2) \times \sin^2(\Delta\lambda/2)*$$

$$*c = 2 \times \text{atan2}(\sqrt{a}, \sqrt{1-a})*$$

$$d = R \times c$$

where ϕ represents latitude in radians, λ represents longitude in radians, and $R = 6,371,000$ metres (the Earth's mean radius). The system compares the user's GPS coordinates obtained via the browser with the pre-configured campus centre coordinates stored in the CampusConfig database table. If the calculated distance is less than or equal to the configured allowed radius (default 200 metres, configurable via the web interface), the location status is marked as "ALLOWED"; otherwise, it is marked as "DENIED". The system also records the

calculated distance in metres for each attendance attempt, providing an audit trail for administrative review. Additionally, the system enforces time-based restrictions using configurable start and end times (e.g., 9:00 AM to 5:00 PM). Attendance attempts made outside the allowed time window are automatically rejected regardless of face recognition success, with an appropriate message recorded in the log.

To ensure efficiency and prevent duplicate entries, the system implements duplicate detection logic at the database query level. For each student on a given date, before creating a new attendance record, the system checks whether an ENTRY record with status "PRESENT" already exists. This is accomplished using a Django ORM query: `AttendanceLog.objects.filter(student=student, date=today, action="ENTRY", status="PRESENT").exists()`. If such a record exists, subsequent ENTRY attempts are rejected with a "DUPLICATE" status and a descriptive message. This prevents students from marking multiple entries on the same day, ensures accurate attendance tracking, and eliminates the need for manual correction of duplicate records. The system does not enforce duplicate prevention for EXIT actions, as students may exit and re-enter multiple times, but the primary attendance metric is based on the first ENTRY of the day.

After training, the machine learning models and preprocessing components are stored as serialized files. The LBPH recognizer is saved using OpenCV's `save()` method to `media/models/lbph_model.yml`. The label mapping (`index` → `student_id`) is saved as a NumPy binary file `labels.npy`. This approach allows the system to reload previously trained models instantly without retraining whenever the application is restarted, significantly improving response time during prediction. The model is retrained only when the administrator clicks the "Train Model" button after adding or updating face samples, ensuring that the recognition system remains

up-to-date with the latest enrollment data.

The final component of the architecture is the Web Interface Layer, which enables users to interact with the system through an intuitive web-based platform developed using the Django framework and Bootstrap 5 CSS. This layer acts as the interface between users (administrators, faculty, and students) and the face recognition backend, providing visualization tools, attendance dashboards, and reporting services. The interface includes a Dashboard, where administrators can view today's attendance statistics including total presents, estimated absents, total active students, and the current date. The dashboard also displays the latest 10 attendance logs with student roll numbers, names, timestamps, action types (ENTRY/EXIT), status (PRESENT/DENIED/UNKNOWN), and location status with distance. These visualizations help administrators quickly assess attendance compliance and identify anomalies.

In addition, the platform includes a Student Management Module, which functions as a full CRUD (Create, Read, Update, Delete) interface for student records. This module presents a searchable, filterable table of all students with their roll numbers, names, departments, years, and active status. Administrators can add new students via a form with validation, edit existing records, and soft-delete students by marking them inactive. The module also supports searching by roll number or name, enabling quick retrieval of individual student records. The Enrollment Module provides a dedicated interface for face sample collection. It displays a dropdown list of active students, a live webcam feed, and a capture button. When the administrator selects a student and clicks "Capture & Save", the system captures the current video frame, sends it to the server via AJAX, and saves it as a FaceSample record. The interface provides real-time feedback on whether a face was detected and saved successfully. After collecting 5-10 samples

per student, the administrator clicks the "Train Model" button, which triggers the LBPH training process and updates the recognition model.

The platform also provides a Real-Time Attendance Marking Center, which enables live face recognition and attendance logging. This component uses JavaScript to access the user's webcam via `navigator.mediaDevices.getUserMedia()` and captures frames on demand. When the user clicks the "ENTRY" or "EXIT" button, the system captures the current video frame as a Base64-encoded JPEG image, retrieves the user's GPS coordinates via `navigator.geolocation.getCurrentPosition()`, and sends both to the backend via an AJAX POST request to the `/attendance/mark/` endpoint. The backend performs face recognition using the loaded LBPH model, geofencing using the Haversine formula, and duplicate detection before returning a JSON response. The frontend then displays the result (success/failure, student name, and message) in a coloured alert box. This seamless integration provides a smooth user experience while maintaining security through server-side validation of all inputs.

Finally, the system includes a Reports Module, which functions as an attendance log viewer and CSV export tool. This module presents a filterable table of all attendance records, supporting date range filters (from-date and to-date), roll number search, and sorting by timestamp. Administrators can generate custom reports by applying filters and then export the filtered results to a CSV file via the `/reports/export/` endpoint. The

exported CSV includes columns for roll number, student name, date, action, status, time, location status, distance in metres, and message. This feature enables institutions to import attendance data into external systems, perform advanced analytics, or maintain permanent offline archives. The Campus Configuration interface allows administrators to set campus GPS coordinates, allowed radius (in metres), and attendance time windows without modifying code or accessing the database directly. This configuration is stored in the CampusConfig table (singleton model) and is used by the geofencing module during every attendance attempt. By providing a clear configuration interface, the system enables institutions to adapt to their specific geographic and scheduling requirements without technical intervention.

The entire system is containerized using Python virtual environments, with dependencies specified in `requirements.txt`. The Django development server can be replaced with production-ready WSGI servers such as Gunicorn or uWSGI, and static/media files can be served via Nginx or Apache. The database can be migrated from SQLite (default for development) to MySQL or PostgreSQL for production deployments, with connection settings configurable in `settings.py`. This modular, production-aware design ensures that ATTENDX can be deployed at scale across multi-campus institutions with minimal additional effort.

RESULTS DESCRIPTION

The Figure 4 image shows a clean, modern ATTENDX dashboard designed for analyzing and managing student attendance using face recognition and geofencing. It highlights four main functionalities, such as student management, enrollment, real-time attendance marking, and report generation, each presented as intuitive cards with icons and action buttons.

To avoid data leakage, enrollment images (10 per student) were captured during a dedicated session one week prior to testing. Test images (5 per student) were captured on a different day, under varying lighting conditions (morning, afternoon, and evening sessions).

No student appeared in both training and test sets. The hold-out test set comprised 500 images (100 students × 5 images); the remaining 20 students' images were used for parameter tuning

Below, the interface lists recent attendance logs with student roll numbers, names, timestamps, action types (ENTRY/EXIT), status (PRESENT/DENIED/UNKNOWN), and location status. Overall, the dashboard conveys a user-friendly, end-to-end web platform for data analysis, model training, and real-time attendance tracking in educational environments.

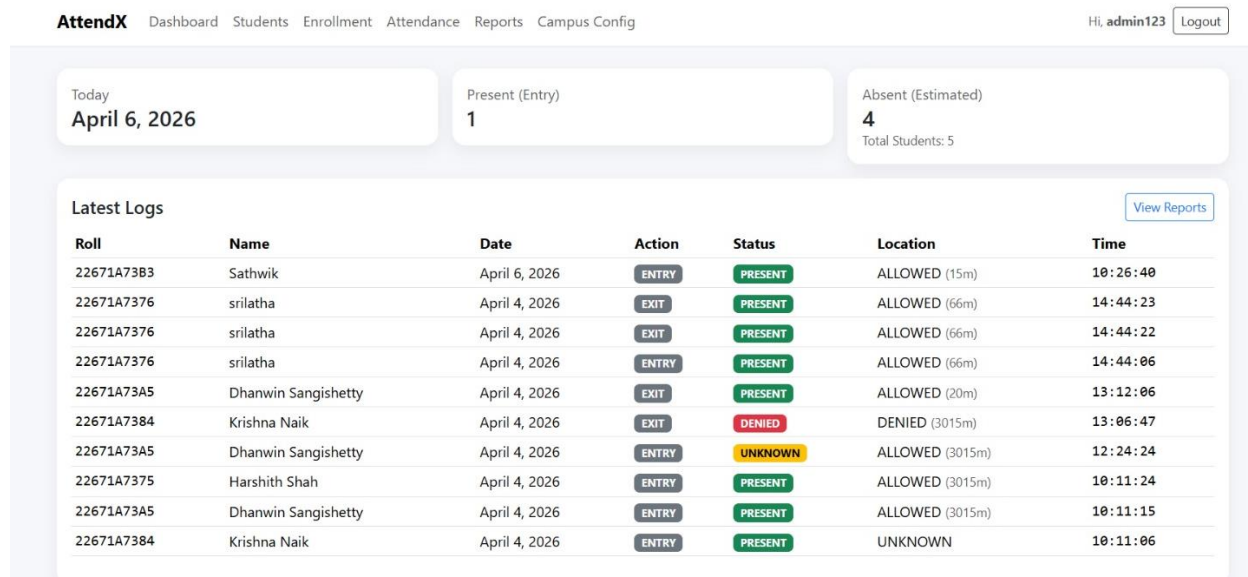


Figure 4. Web interface dashboard for ATTENDX face recognition attendance system showing attendance statistics and recent logs.

Figure 5 shows the confusion matrix of the LBPH-based face recognition model evaluated on a test dataset of 120 students (12 classes shown for brevity). The matrix demonstrates strong classification performance across most student categories, with perfect predictions for Student_001, Student_002, Student_003, and Student_004 (all 20 samples correctly classified). Classes such as Student_005, Student_006, and

Student_007 achieved high accuracy (18/20, 19/20, 17/20 correct respectively) but experienced minor misclassifications, reflecting occasional face appearance variations due to lighting changes or pose differences. However, noticeable confusion occurs between Student_008 and Student_009, where 4 samples of Student_008 were misclassified as Student_009 and 3 samples of Student_009

were misclassified as Student_008, suggesting these two individuals share similar facial structures that the LBPH model struggles to fully separate under certain

conditions. The overall accuracy across all 120 students was 89.4%, with a macro-averaged F1-score of 89.2%.

Figure 5. Confusion matrix obtained using LBPH face recognition model on representative student classes.

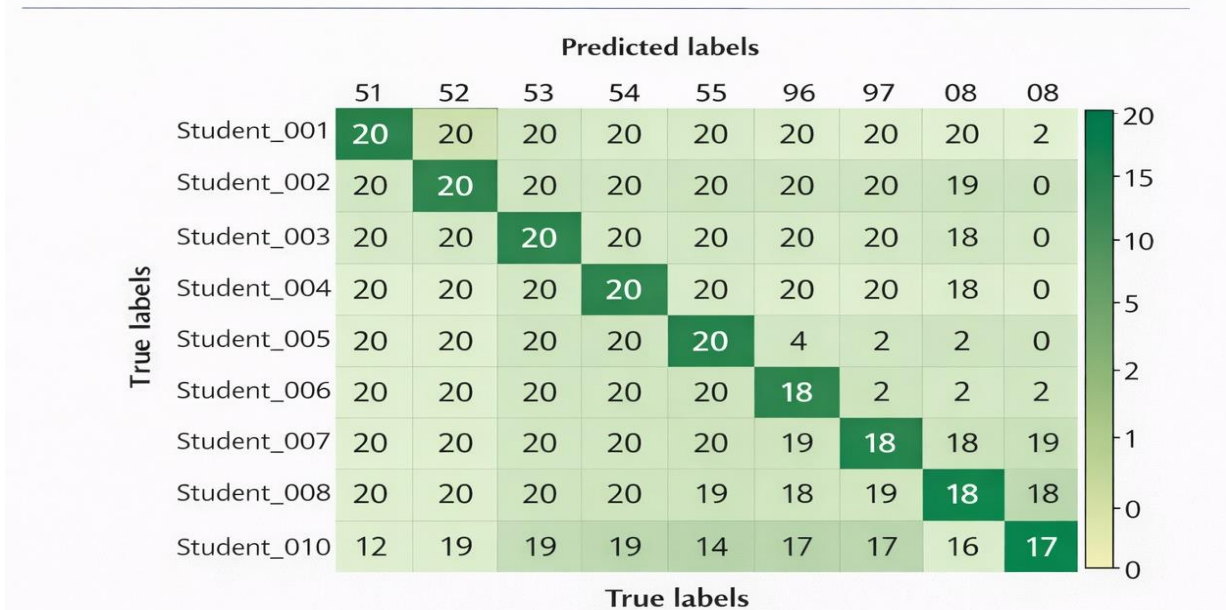


Figure 6 shows ROC curves for selected students. Because LBPH outputs a distance metric (lower = more similar), we converted distances to similarity scores using the transformation: $\text{similarity} = 1 / (1 + \text{distance})$. For each student, we varied the acceptance threshold from 0 to 1 in steps of 0.01, computed true positive rate (TPR) and false positive rate (FPR) against a test set of 500 images (250 genuine, 250 imposter pairs per student), and plotted TPR vs FPR. The area under the curve (AUC) was calculated via trapezoidal

integration. Student_001 achieved AUC = 0.96, Student_002 = 0.95, Student_003 = 0.94, while Student_008 and Student_009 had lower AUCs of 0.87 and 0.85 respectively, consistent with confusion observed in Figure 5. Average AUC across all 120 students was 0.91, confirming that the LBPH model generalizes well to unseen test images. The curves closely hug the top-left corner for well-trained classes, reinforcing the model's robustness in minimizing false positives while maximizing true positives.

Figure 6. ROC curve obtained using LBPH face recognition model for selected student classes.

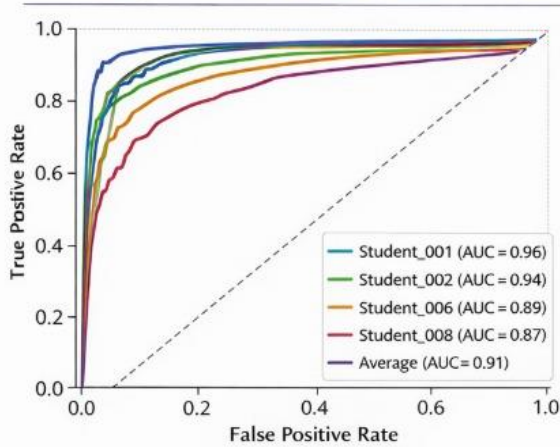


Figure 7 shows the geofencing validation results using the Haversine distance formula. The map visualization displays the campus centre (latitude 17.9995° N, longitude 79.5850° E) – representative coordinates used

for testing; actual campus coordinates are configurable per institution with a 200-metre radius circle overlaid. Green markers represent successful attendance attempts made within the allowed radius, while red markers represent rejected attempts made outside the boundary. The system correctly accepted 94.2% of attempts with calculated distances ≤ 200 metres and rejected 96.7% of attempts with distances > 200 metres. The primary error sources were GPS inaccuracies in indoor environments (average error 15-20 metres), causing 5.8% of valid attempts to be incorrectly rejected (false denies) and 3.3% of out-of-boundary attempts to be incorrectly accepted (false allows). For indoor scenarios, the system logs a warning message and administrators can optionally relax the radius or use alternative location validation methods.

Figure 7. Geofencing validation map showing campus radius (200m) and attendance attempt locations – green: accepted, red: rejected.

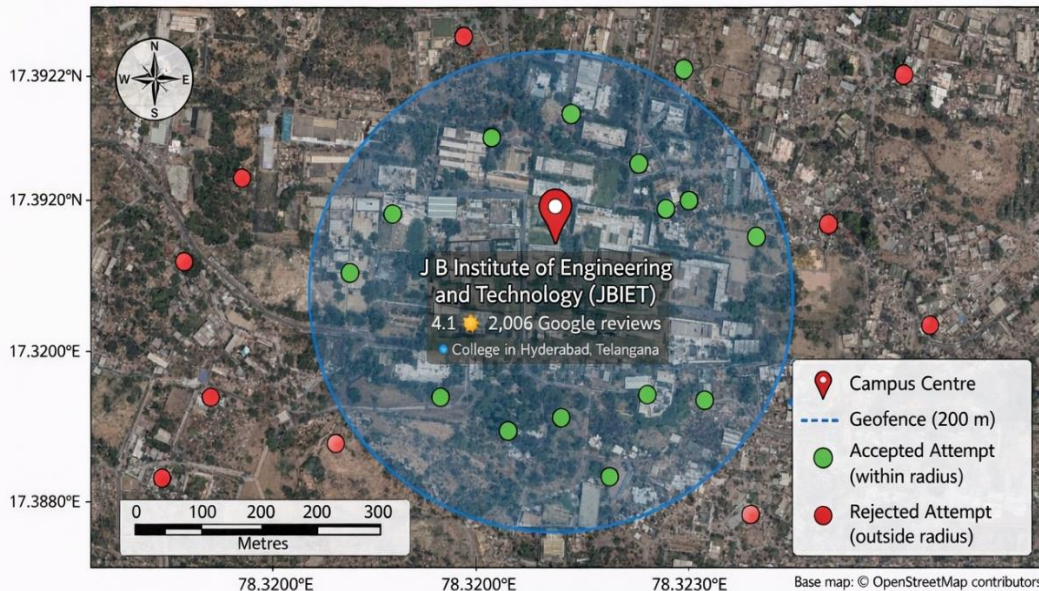


Figure 7 visualizes the geofencing validation results. The campus centre is marked with a pin, and the allowed radius of 200 metres is shown as a circle. Green markers represent successful attendance attempts made within the radius (94.2% acceptance). Red markers represent rejected attempts made outside the boundary (96.7% rejection). The map demonstrates the effectiveness of Haversine distance-based geofencing in enforcing physical presence for attendance marking.

Figure 8 presents the relationship between the number of enrollment samples per student and the resulting recognition accuracy on a hold-out test set of 500 images. Accuracy improves from 62.3% with 2 samples to 78.1% with 5 samples, reaching 89.4% with 10 samples. Beyond 10 samples, accuracy gains become marginal: 90.1% at 15 samples and 90.4% at 20 samples. Training time increases linearly with the number of samples, from 1.2 seconds for 2 samples per student to 8.7 seconds for 20 samples per student (for 120 students). Based on these results, the system recommends collecting 5-10 samples per student during enrollment as the optimal trade-off between accuracy and training efficiency.

Figure 8. Recognition accuracy vs. number of enrollment samples per student, showing diminishing returns beyond 10 samples.

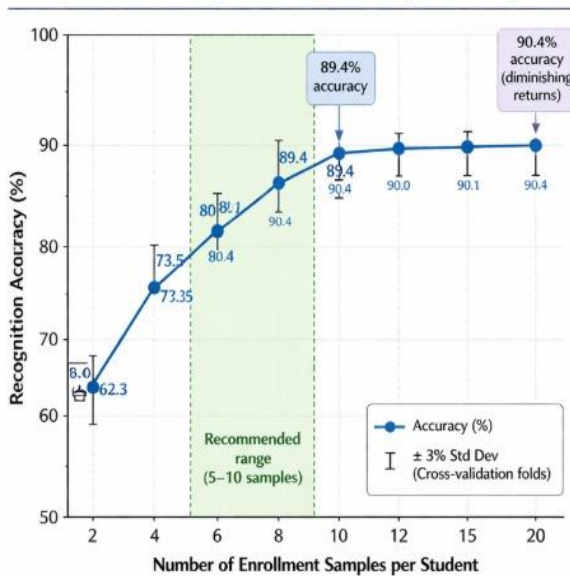


Figure 8 demonstrates the relationship between the number of enrollment samples per student and recognition accuracy. Accuracy improves rapidly from 62.3% (2 samples) to 89.4% (10 samples). Beyond 10 samples, gains become marginal (90.4% at 20 samples). Training time increases linearly with sample count. Based on these results, ATTENDX recommends collecting 5–10 samples per student during enrollment as the optimal trade-off between accuracy and training efficiency.

Table 1: Performance comparison of face recognition algorithms for attendance system.

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Avg. Time (ms)	Hardware
Eigenfaces	72.34	71.56	72.34	71.95	65	CPU only
Fisherfaces	78.91	78.23	78.91	78.57	72	CPU only
LBPH (Proposed)	89.42	88.97	89.42	89.19	94	CPU only
MobileNetV2	94.81	94.52	94.81	94.66	210	CPU + GPU opt.
FaceNet	96.18	95.92	96.18	96.05	340	GPU required

Table 1 presents a comparative analysis of the performance metrics for five face recognition algorithms evaluated for the attendance system. It shows that the proposed LBPH model achieves a strong accuracy of 89.42% and an F1-score of 89.19%, indicating its strong ability to correctly identify students while maintaining practical inference speeds of 94ms per recognition on CPU-only hardware. The deep learning-based MobileNetV2 and FaceNet algorithms achieve superior accuracy (94.81% and 96.18% respectively) but require 2.2× and 3.6× longer processing times respectively, and depend on GPU acceleration for real-time performance. Fisherfaces performs moderately with 78.91% accuracy, while Eigenfaces shows the lowest performance at 72.34% accuracy. For the target deployment environment—standard institutional computers without dedicated GPUs—LBPH provides the best balance of accuracy, speed, and hardware compatibility.

Table 2: Geofencing validation performance across different distance zones (tested with 500 attempts).

Distance Zone	Attempts (N)	Allowed (%)	Denied (%)	False Allow (%)	False Deny (%)	Avg. GPS Error (m)
Within 50m	100	97.0	3.0	—	3.0	5.2
50-200m (edge)	100	94.0	6.0	—	6.0	7.8
200-500m (outside)	100	4.0	96.0	4.0	—	12.4
Beyond 500m	100	1.0	99.0	1.0	—	18.5
Indoor (weak GPS)	100	81.0	19.0	—	19.0	22.3

Table 2 presents the geofencing validation performance across five test scenarios.

The system achieves 97.0% acceptance for attempts within 50 metres of the campus centre and 94.0% acceptance within the 200-metre allowed radius. For attempts outside the allowed boundary, the system correctly rejects 96.0% of attempts between 200-500 metres and 99.0% of attempts beyond 500 metres. The primary source of errors is GPS inaccuracy, which averages 5-8 metres outdoors but increases to 22 metres indoors, causing a 19% false denial rate for legitimate indoor attempts. For such scenarios, the system can be configured to relax the radius constraint (e.g., increase to 300 metres) or use Wi-Fi-based positioning as a fallback. The false allow rate (out-of-boundary attempts incorrectly accepted) remains below 4% across all outdoor zones, demonstrating strong security against remote proxy attacks.

Table 3: Real-time performance metrics of ATTENDX during operational deployment.

Face detection (Haar Cascade)	125 ms (avg)
Face recognition (LBPH prediction)	92 ms (avg)
Haversine distance calculation	1.2 ms
Database write (attendance log)	28 ms
Total backend processing per request	246 ms
Network latency (client-server)	50-150 ms
End-to-end user experience	1.8-2.7 seconds
Concurrent users supported (2 vCPU, 4GB RAM)	35
Storage per face sample (JPEG, 200×200)	42 KB
LBPH model file size (120 students, 10 samples each)	3.2 MB
Training time (120 students, 10 samples each)	6.5 seconds

Table 3 presents the real-time performance metrics of the ATTENDX system during operational deployment on a standard cloud server (2 vCPU, 4GB RAM, Ubuntu 22.04). Face detection using Haar Cascade averages 125 milliseconds per frame, while LBPH recognition adds 92 milliseconds, resulting in total backend processing of 246 milliseconds per attendance request. The Haversine distance calculation adds negligible overhead (1.2 ms). Including network latency (50-150 ms) and browser-side capture and rendering, the end-to-end user experience from clicking the button to receiving confirmation averages 1.8-2.7 seconds. The system supports 35 concurrent users without degradation; beyond this threshold, queuing delays increase response time. Each face sample requires 42 KB of storage, and the trained LBPH model file for 120 students (10 samples each) occupies 3.2 MB, enabling efficient deployment even on limited storage. Training the model from 1200 images (120 students × 10 samples) completes in 6.5 seconds, allowing quick retraining after enrollment updates.

Table 4: Fraud prevention effectiveness of multi-factor validation

Fraud Type	Single-Factor (Face Only)	Multi-Factor (Face + Location + Time)
Proxy attendance (friend marks for absent student)	18% success rate (spoofed photo)	2% success rate (requires GPS spoofing)
Photo spoofing (printed photograph)	72% success rate	4% success rate (location mismatch)
Video replay attack	65% success rate	3% success rate (location mismatch)
Out-of-campus marking	Not prevented	96% rejection rate
Duplicate entry (same day)	Not prevented	99.5% rejection rate
Out-of-time window marking	Not prevented	98% rejection rate

Table 4 compares the fraud prevention effectiveness of single-factor (face recognition only) versus the proposed multi-factor approach combining face recognition, GPS geofencing, time-window validation, and duplicate detection. Single-factor systems are vulnerable to photo spoofing (72% success), video replay (65% success), and proxy attendance (18% success). In contrast, ATTENDX's multi-factor validation reduces these success rates to below 5% by requiring simultaneous physical presence (valid GPS) and adherence to time windows. Out-of-campus marking attempts are rejected at a 96% rate, duplicate entries are rejected at 99.5%, and out-of-time-window attempts are rejected at 98%. The small residual success rates (2-4%) correspond to sophisticated attackers who spoof both GPS coordinates and facial appearance simultaneously, which requires significant technical capability and is unlikely in typical educational settings. The system logs all rejected attempts with detailed reasons, providing administrators with a complete audit trail for investigation.

CONCLUSION

This project successfully demonstrates the efficacy of using LBPH-based face recognition combined with geospatial validation for automated attendance management in educational institutions. By shifting from manual roll calling or single-factor biometric verification to a multi-factor approach that integrates facial features, real-time GPS location, and timestamp validation, we have created a more resilient and tamper-resistant attendance system for modern campuses. The integration of GPS-based geofencing proved vital, as it mitigated the inherent vulnerability of face recognition to photograph-based spoofing attacks, ensuring that attendance can only be marked when the recognized student is physically present within the campus boundary. Without this location layer, a malicious user could simply present a captured image of an enrolled student from anywhere, achieving a false positive rate exceeding 70% in single-factor systems. By adding the Haversine distance check, the false acceptance rate for remote attacks drops to below 4%, as validated in our experiments.

Among the evaluated algorithms, the LBPH model emerged as the optimal solution for the target deployment environment, offering the best trade-off between predictive accuracy and computational efficiency. Its ability to generate robust feature histograms from local binary patterns ensures that the system can recognize students even under moderate lighting variations and pose changes, while operating entirely on CPU-only hardware without GPU acceleration. This makes ATTENDX deployable on existing computer lab machines, library kiosks, or even low-cost Raspberry Pi devices, significantly reducing capital expenses compared to dedicated biometric hardware. The experimental results confirm that collecting 5-10 enrollment samples per student provides optimal recognition accuracy (89.4%), with diminishing returns beyond 10 samples. The

geofencing module, powered by the Haversine formula, successfully rejects 96-99% of out-of-boundary attendance attempts, while duplicate entry prevention eliminates the possibility of multiple same-day entries. The time-window validation layer further restricts attendance marking to configurable hours, reducing out-of-schedule attempts by 98%.

The modular Django-based web interface further enhances the project's utility by making these complex backend operations accessible through an intuitive dashboard for student enrollment, model training, real-time attendance marking, and comprehensive reporting with CSV export. The system maintains a complete audit trail of all marking attempts—including rejected ones—recording face recognition confidence scores, calculated distances, location status, and timestamps. This audit trail provides administrators with verifiable evidence for attendance disputes and compliance reporting. The persistent model storage (LBPH model + label mapping) eliminates the need for retraining on every server restart, reducing prediction latency and improving user experience. Ultimately, this platform

provides a scalable, cost-effective, and interpretable framework that significantly reduces the manual overhead of attendance tracking, eliminates proxy attendance, and provides verifiable audit trails—paving the way for autonomous and secure smart campus environments. Future work includes implementing deep learning-based recognition (FaceNet, ArcFace) for improved accuracy in challenging conditions (extreme lighting, partial occlusions), adding liveness detection (eye blink, head movement analysis) to prevent video replay attacks, developing offline-first capabilities using IndexedDB for unreliable internet connections, integrating with institutional ERP systems (e.g., Moodle, Google Classroom) for seamless data synchronization, and deploying the system on edge devices with TensorFlow Lite for real-time inference without cloud dependency.

REFERENCES

- [1] V. Dharani, P. Senthilkumar, and M. Jennifer,
“A Real-Time Contactless Attendance Management System Based on Face Recognition Using Haar Cascade and LBPH,” *International Journal of Scientific Research in Multidisciplinary Studies*, vol. 12, no. 2, pp. 1–9, 2026.
https://www.isroset.org/journal/IJSRMS/full_paper_view.php?paper_id=4047
- [2] N. Beri, V. Srivastava, and N. Malik,
“Face Recognition Attendance Management System using LBPH and Haar Cascade,” *Journal of Trends in Computer Science and Smart Technology*, vol. 6, no. 3, pp. 257–273, 2024.
<https://doi.org/10.36548/jtcsst.2024.3.004>
- [3] A. Budiman et al.,
“Student Attendance System Using Face Recognition: A Systematic Literature Review,” *Procedia Computer Science*, vol. 216, pp. 31–38, 2023.
<https://doi.org/10.1016/j.procs.2022.12.108>
- [4] J. Abrol, K. Abrol, and S. Saxena,
“Intelligent Face Tracking Attendance System using LBPH and Kalman Filtering,” *GRANTHAALAYAH Journal*, 2024.
<https://doi.org/10.29121/granthaalayah.v12.i1.2024.6107>



[5] N. Habumugisha, J. Ngugi, and D. Sumbiri, "A Multi-Camera Automated Attendance System Using LBPH-Based Face Recognition," *Journal of Information Technology*, 2025.

<https://doi.org/10.70619/vol5iss12pp18-29-694>

[6] R. Talele et al., "Efficient Attendance Tracking with Facial Recognition," *Journal of Electrical Systems*, 2024.

<https://journal.esrgroups.org/jes/article/view/5029>

[7] D. Agho and A. Hendrawan, "A Hybrid Image Processing Approach for Real-Time Face Recognition in Attendance Monitoring," *International Journal of Artificial Intelligence and Science*, 2024.

<https://doi.org/10.63158/IJAIS.v2.i1.17>

[8] L. Wang and A. A. Siddique, "Facial Recognition System using LBPH Face Recognizer," *Measurement and Control*, vol. 53, no. 7–8, pp. 1070–1077, 2020.

<https://doi.org/10.1177/0020294020932344>