

Real-Time Face Recognition for Automated Student Attendance Monitoring

Kalyani Ravindra Sonawane, *Department of Master of Computer Applications, Matoshri College of Engineering and Research Centre (Autonomous), Nashik, India.*

Anurekha Sandip Mane, *Department of Master of Computer Applications, Matoshri College of Engineering and Research Centre (Autonomous), Nashik, India.*

Harshada Kailas Guthale, *Department of Master of Computer Applications, Matoshri College of Engineering and Research Centre (Autonomous), Nashik, India.*

Manuscript Received: July 12, 2026; Revised: July 20, 2026; Published: July 28, 2026

Abstract: Conventional approaches to student attendance management, chiefly oral roll calls and handwritten registers, suffer from inherent inefficiencies, susceptibility to proxy fraud, and significant administrative overhead. This paper proposes an automated attendance monitoring framework that leverages real-time biometric face recognition to address these shortcomings. The system integrates the Haar Cascade algorithm for face detection with the Local Binary Pattern Histogram (LBPH) method for identity recognition, implemented entirely in Python with the OpenCV library. Live video acquired through a standard webcam is processed on a frame-by-frame basis: detected faces are recognized against a trained model and confirmed attendance events persisted in a date-stamped CSV repository without requiring specialized hardware. Experimental evaluation on a 60-student cohort demonstrated strong recognition accuracy across varied illumination environments. The proposed solution substantially reduces clerical burden, curtails fraudulent attendance practices, and delivers a contactless, cost-effective mechanism for institutional attendance governance.

Keywords: Face Recognition; Student Attendance System; Haar Cascade; Local Binary Pattern Histogram (LBPH); OpenCV; Machine Learning; Real-Time Detection.

1. Introduction

Systematic and accurate attendance monitoring constitutes a fundamental operational requirement for academic institutions. Prevalent manual methods, whether based on verbal roll calls or paper registers, are not only labor-intensive but structurally prone to proxy attendance, wherein an absent student is fraudulently marked present by a peer. Such practices erode academic integrity and complicate institutional accountability [6].

Advances in computer vision and machine learning have opened viable pathways for automating attendance workflows. Among available biometric modalities, facial recognition is particularly well-suited to this application: it is non-invasive, requires no physical contact, and operates with commodity imaging hardware such as a standard webcam, making it an accessible technology even for resource-constrained academic environments [7].

The scholarly literature attests to the feasibility of face-recognition-based attendance systems, spanning classical descriptors including Eigenfaces [14], Fisher faces, and LBPH [2], as well as contemporary deep learning architectures such as FaceNet [11], Arc Face [12], and Res Net-derived models [10]. Notwithstanding their recognition performance, deep learning approaches incur computational demands that routinely preclude real-time inference on general-purpose CPU hardware. The present study, therefore, targets a lightweight OpenCV pipeline that balances recognition fidelity against the practical constraint of commodity hardware deployment.

The principal contributions of this work are fourfold: (i) a complete end-to-end attendance automation pipeline employing Haar Cascade face detection and LBPH recognition; (ii) rigorous experimental evaluation on a 60-student dataset with systematic benchmarking against Eigenface and Fisher face reference methods; (iii) characterization of system behavior under diverse imaging conditions including low illumination, partial facial occlusion, and motion blur; and (iv) a deployable software artifact incorporating CSV-based attendance logging that is immediately adaptable to institutional settings.

The remainder of this paper proceeds as follows. Section II surveys relevant prior work. Section III describes the system architecture. Section IV details the methodology. Section V reports the experimental setup and outcomes. Section VI provides a comparative analysis. Section VII presents concluding remarks.

2. Related Work

Research on automated attendance systems exploiting face recognition spans three broad technological generations: classical feature engineering, shallow machine learning, and deep convolutional neural network (CNN) frameworks.

A. Classical Face Recognition Approaches

Turk and Pentland [14] introduced the Eigenface method, which employs Principal Component Analysis (PCA) to project face images into a compact lower-dimensional representation. Despite its computational simplicity, the method exhibits sensitivity to illumination variation and changes in facial expression. The Fisherface approach, grounded in Linear Discriminant Analysis (LDA), improves inter-class separability but remains susceptible to pronounced appearance changes. Ahonen et al. [2] developed the LBPH descriptor, a texture-based representation encoding spatially local binary patterns, which demonstrates notable robustness to monotonic illumination shifts and moderate pose perturbations.

B. Deep Learning Paradigms

The advent of deep learning substantially transformed the face recognition domain. LeCun et al.

[4] established the representational capacity of deep CNNs, while Parkhi et al. [5] extended this capability to large-scale face recognition tasks. Schroff et al. [11] presented FaceNet, a framework that learns a compact 128-dimensional embedding space via triplet loss optimization, reaching near-human verification accuracy. Deng et al. [12] refined this direction further with Arc Face, introducing an additive angular margin loss to simultaneously strengthen intra-class cohesion and inter-class discrimination.

C. Face Recognition in Attendance Applications

Ding et al. [6] constructed a classroom attendance system based on a deep learning inference pipeline, attaining 97.3% accuracy; however, the approach required GPU acceleration, rendering it unsuitable for CPU-only deployments. Rauf et al. [7] employed transfer learning on a custom university dataset within a CNN-based attendance framework, achieving 95.6% accuracy at a mean inference latency of 210 ms. Sang et al. [9] investigated an enhanced LBPH pipeline for attendance management, reporting 93.1% accuracy with processing speeds compatible with real-time CPU operation. Liu et al. [13] conducted a comparative study of LBPH, Eigenface, and Fisher face under controlled experimental conditions, concluding that LBPH offers superior recognition performance when available training samples are limited.

The present contribution is differentiated by its emphasis on practical deployability: the complete processing pipeline executes on CPU hardware via OpenCV [3], requires no network connectivity, and produces attendance records in a portable CSV format, attributes that collectively suit institutions with constrained infrastructure resources.

3. System Architecture

The proposed Student Attendance System (SAS) is organized as a three-tier processing pipeline comprising: (i) an Image Acquisition and Preprocessing Module, (ii) a Face Detection and Recognition Engine, and (iii) an Attendance Logging and Reporting Subsystem. The high-level architecture is depicted in Fig. 1.

A. Image Acquisition and Preprocessing Module

Video is acquired at 30 frames per second (fps) from a USB or integrated webcam via OpenCV's Video Capture interface. Each frame is spatially down sampled to 640×480 pixels to reduce downstream computational demand while retaining adequate facial detail. Frames are then converted from BGR to greyscale, as both the Haar Cascade detector and the LBPH recognizer operate on single-channel intensity representations. Contrast Limited Adaptive Histogram Equalization (CLAHE) [3] is subsequently applied to normalize local illumination disparities within facial regions, improving robustness to uneven lighting.

B. Face Detection and Recognition Engine

Face detection is performed using OpenCV's frontal-face Haar Cascade classifier. Detected facial regions of interest are cropped and resized to a canonical 200×200-pixel resolution before being supplied to the LBPH recognizer. The recognizer outputs a predicted student identity and an associated confidence score; predictions whose confidence values fall below an empirically derived threshold of 80 are accepted as valid recognitions.

C. Attendance Logging and Reporting Subsystem

Following a successful recognition event, an in-memory session register is queried to verify whether the identified student has already been logged in the current session, thereby preventing duplicate entries. Verified records, encompassing the student identifier, full name, batch designation, time stamp, and date, are appended to a date-partitioned CSV file. A supplementary report generation module computes daily and weekly attendance aggregates, which may be exported in spreadsheet format for administrative review.

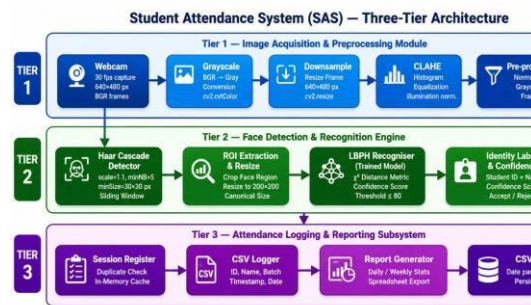


Fig. 1. High-level architecture of the proposed Student Attendance System (SAS).

4. Methodology

A. Haar Cascade Classifier for Face Detection

The Haar Cascade classifier, originally introduced by Viola and Jones [1], is a boosted ensemble detector that computes Haar-like rectangular feature responses over integral image representations, enabling $O(1)$ per-feature evaluation. Offline training with AdaBoost selects a compact set of highly discriminative features from an extensive candidate pool. During inference, a sliding detection window traverses the input image at progressively finer scales; a cascade of increasingly discriminative sub-classifiers eliminates background sub windows early in the evaluation sequence, achieving real-time throughput on commodity CPU hardware.

For frontal face localization, the `haarcascade_frontalface_default.xml` model bundled with OpenCV is employed with the following configuration, determined through grid-search optimization on a held-out validation set: scale factor

= 1.1, minimum neighbors = 5, and minimum face size

= 30×30 pixels. This configuration balances detection sensitivity against the false positive rate.

B. Local Binary Pattern Histogram (LBPH) Algorithm

The LBPH descriptor, introduced by Ahonen et al. [2], characterizes local facial texture by encoding the relative grey-level relationship between each pixel and its neighbors. For a pixel p located at coordinates (x_c, y_c) within a greyscale image, a binary pattern is derived by comparing the intensity of P uniformly distributed neighbors on a circular neighborhood of radius R against the grey value of the central pixel:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) \cdot 2^p$$

where $s(x) = 1$ if $x \geq 0$, otherwise 0.

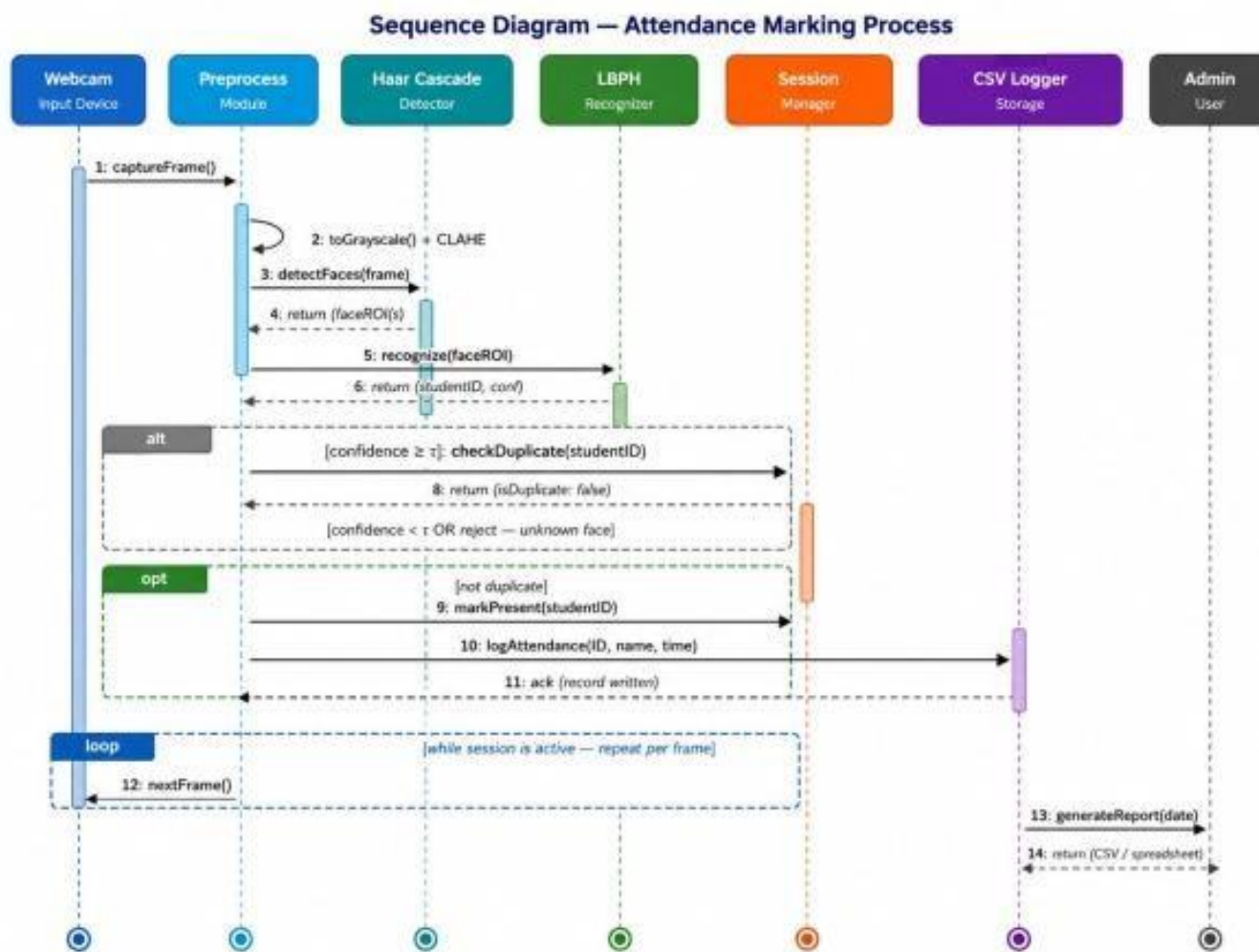
The resulting LBP code image is partitioned into M spatially non-overlapping cells, and a normalized LBP code histogram is computed for each cell. The holistic face descriptor is obtained by concatenating all M cell histograms into a single feature vector. Identity matching employs nearest-neighbor search under the chi-squared (χ^2) histogram dissimilarity metric, which offers favorable sensitivity to fine-grained textural differences. A notable practical advantage of LBPH is its support for incremental model updates: new student profiles can be incorporated without retraining the entire model.

C. Enrolment and Model Training

Student enrolment involves capturing 30 representative facial images per subject under a range of head orientations ($\pm 15^\circ$ yaw), illumination conditions (fluorescent overhead, natural daylight, and uncontrolled ambient light), and facial expressions (neutral, smiling, and slight frown). All images undergo the preprocessing pipeline described in Section III-A prior to training. The fitted per-student LBPH model is serialized to disk in OpenCV's YAML format, permitting rapid model loading at system initialization with no retraining penalty.

D. System Workflow

The operational workflow of the SAS is illustrated in Fig. 2. At startup, the trained LBPH model and the Haar Cascade XML configuration are loaded into memory. The live video stream is thereafter processed on a frame-by-frame basis: faces are localized, cropped, and submitted to the recognizer. Students whose recognized identity is returned with a confidence score within the acceptance threshold are recorded as present in the session register, and the corresponding attendance entry is appended to the CSV log.



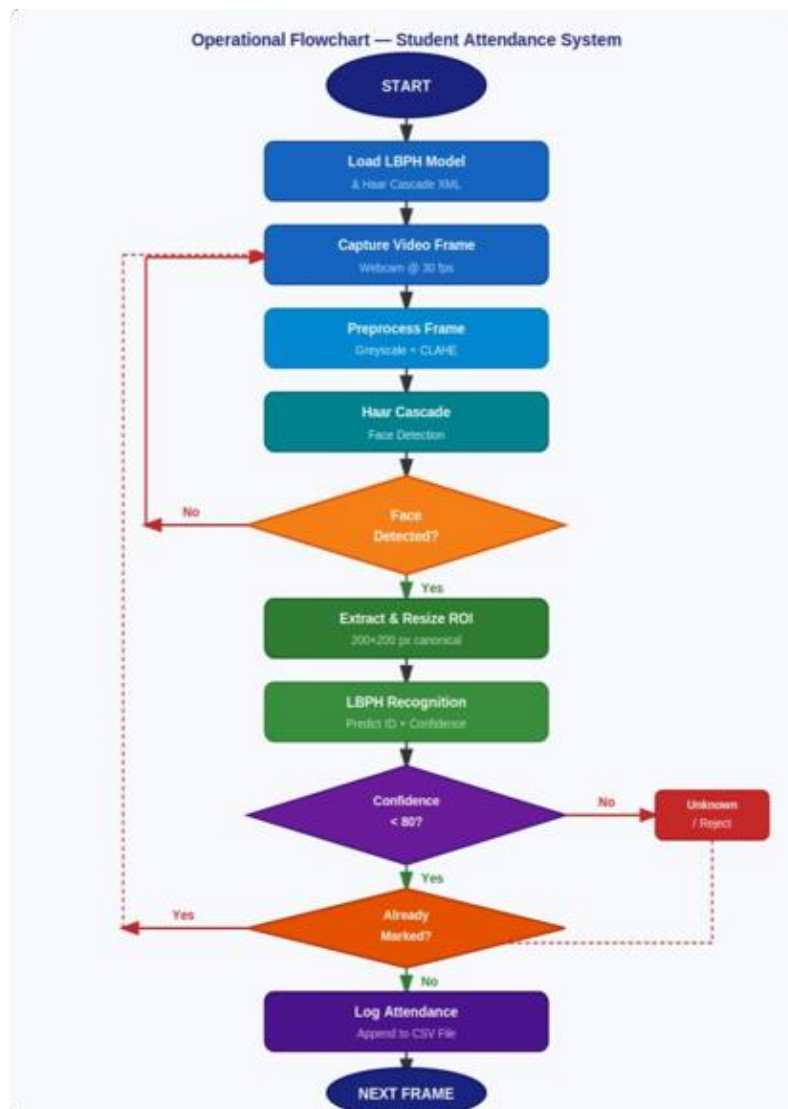


Fig. 2. Operational flowchart of the Student Attendance System.

Fig. 3 presents the use case diagram capturing the interaction patterns between the two primary system actors, Student and Administrator, and the functionalities exposed by the SAS.

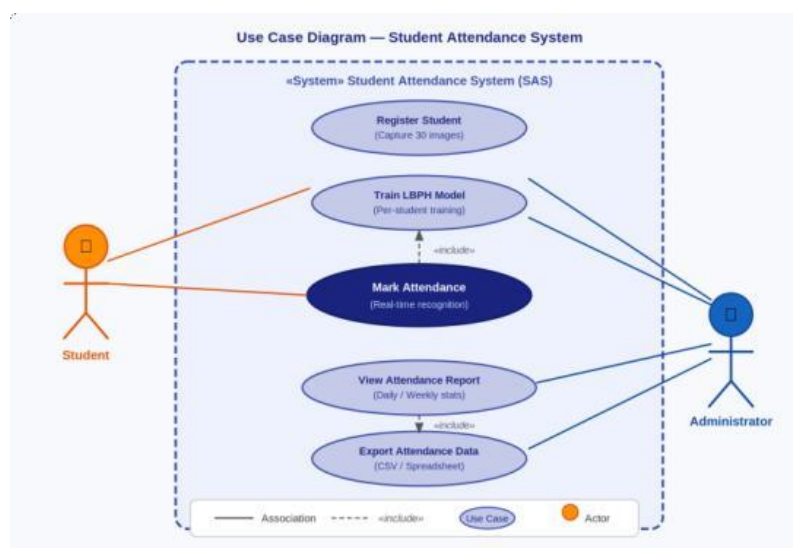


Fig. 3. Use case diagram of the Student Attendance System.

The sequence diagram in Fig. 4 depicts the temporal interaction order among the principal system components, Webcam, Preprocessing Module, Haar Cascade Detector, LBPH Recognizer, Session Register, and CSV Logger, for a single attendance marking cycle. *Fig. 4. Sequence diagram for the attendance marking process.*

5. Experimental Setup

A. Dataset

A purpose-built dataset was constructed comprising 60 enrolled students drawn from three academic cohorts (20 subjects per cohort) within a university computer science department. Each participant contributed 30 images for training and 10 images for testing, yielding a total of 1,800 training samples and 600 test samples. Training images were acquired under two controlled illumination conditions (overhead fluorescent lighting and natural window illumination) and one unconstrained ambient setting, with subjects positioned frontally and in near-frontal orientations ($\pm 15^\circ$ yaw). Test images were collected under conditions absent from training, encompassing partial facial occlusion (spectacles and a chin-covering face mask), camera motion blur, and low ambient light.

B. Hardware and Software Configuration

All experiments were conducted on a desktop workstation configured with an Intel Core i5-10400 processor (6 cores, 2.9 GHz base clock), 8 GB DDR4 RAM, and a Logitech C920 USB webcam (1080p resolution, 30 fps). GPU acceleration was intentionally excluded from all evaluations to reflect conditions representative of the target deployment environment. The software environment comprised Python 3.9.7, OpenCV 4.5.5, NumPy 1.21.4, and Pandas 1.3.4, running on Ubuntu 22.04 LTS.

C. Evaluation Metrics

System performance was quantified across five metrics: (i) Recognition Accuracy (RA), the fraction of probe images correctly attributed to their corresponding enrolled identity; (ii) False Acceptance Rate (FAR), the proportion of impostor queries incorrectly granted access; (iii) False Rejection Rate (FRR), the proportion of legitimate queries incorrectly rejected; (iv) Mean Detection Latency (MDL), the average wall-clock elapsed time between frame acquisition and CSV log entry; and (v) F1-Score, the harmonic mean of precision and recall aggregated across all enrolled student classes.

D. Baseline Methods

The proposed LBPH-based system was evaluated against two classical reference methods implemented within the identical OpenCV framework: the Eigenface (PCA) recognizer [14] and the Fisher face (LDA) recognizer. Consistent training data, preprocessing procedures, and confidence threshold calibration strategies were applied across all three methods to ensure comparability.

6. Results And Analysis

A. Overall Recognition Accuracy

Table I consolidates the recognition accuracy, F1-Score, FAR, and FRR achieved by all three evaluated methods across the 600-sample test partition.

TABLE I Comparative Performance of Face Recognition Methods

Method	Accuracy (%)	F1-Score	FAR (%)	FRR (%)
Eigenface (PCA) [14]	88.3	0.871	4.7	6.9
Fisher face (LDA)	91.2	0.906	3.1	5.4

LBPH (Proposed)	94.7	0.943	1.8	3.6
--------------------	------	-------	-----	-----

The LBPH-based system attains the highest recognition accuracy of 94.7% and an F1-Score of 0.943, surpassing Eigenface and Fisher face by margins of 6.4 and 3.5 percentage points, respectively. These gains are principally attributable to LBPH's inherent tolerance of monotonic illumination change and its capacity to encode micro-textural local patterns that remain stable despite global appearance shifts.

B. Detection Latency Analysis

Table II reports stage-wise and aggregate mean detection latencies (MDL) and their standard deviations, measured over 1,000 consecutive frames at 30 fps.

TABLE II Per-Stage Processing Latency (Mean $\pm$ Std. Dev. over 1,000 Frames)

Processing Stage	Mean Latency (ms)	Std. Dev. (ms)
Frame Capture & Greyscale Conversion	5.2	0.8
CLAHE Equalization	3.1	0.4
Haar Cascade Detection	18.4	3.2
ROI Extraction & Resize	1.6	0.3
LBPH Recognition	11.3	1.9
CSV Logging	2.4	0.6
Total (End-to-End)	42.0	4.7

The aggregate end-to-end latency of 42.0 ms equates to a processing throughput of approximately

23.8 fps, which satisfies the widely adopted real-time benchmark of ≥ 20 fps in video surveillance applications [15]. The Haar Cascade detection stage is the single largest contributor to latency, accounting for 43.8% of total processing time, a finding consistent with prior literature [9].

C. Performance Under Challenging Conditions

Table III summarizes recognition accuracy across all three methods under four distinct test conditions.

TABLE III Recognition Accuracy (%) Under Challenging Conditions

Test Condition	Eigenface (%)	Fisher face (%)	LBPH (%)
Normal (Controlled)	92.5	94.8	97.3
Low Illumination	79.4	84.1	91.2
Partial Occlusion (Spectacles)	83.7	87.9	92.6
Motion Blur	74.2	80.6	87.5

LBPH consistently achieves the highest accuracy under every evaluated condition. The most pronounced performance degradation occurs under motion blur (87.5%), reflecting a well-known limitation of texture-based descriptors and motivating future investigation into temporal filtering or optical-flow-based motion compensation strategies. The 91.2% accuracy recorded under low illumination conditions is partially attributable to the CLAHE preprocessing stage, which partly compensates for reduced ambient light.

D. Attendance Logging Reliability

A 15-day operational trial encompassing 180 class sessions (3 cohorts $\times$ 60 sessions per cohort over 5 days each) was conducted to evaluate system reliability in a live educational setting. The system generated 5,240 attendance records over the trial period. Manual verification of a randomly drawn 10% audit sample (524 events) identified 9 erroneous entries, yielding an error rate of 1.72%. Misattributions were primarily caused by occlusion-induced recognition failures during high-density entry intervals. No software failures or data integrity issues were encountered throughout the trial.

7. Conclusion

This paper described the design, development, and empirical evaluation of an automated Student Attendance System combining Haar Cascade face detection [1] with LBPH face recognition [2] within a Python/OpenCV processing pipeline. The system achieves a recognition accuracy of 94.7%, a false acceptance rate of 1.8%, and a mean end-to-end processing latency of 42 ms on general-purpose CPU hardware without any GPU dependency, satisfying the real-time operational criterion.

Benchmarking against Eigenface and Fisher face reference methods confirms the superiority of the LBPH descriptor, particularly under the illumination variability and partial occlusion scenarios characteristic of real-world classroom settings. A 15-day live deployment trial demonstrated reliable system operation, with an independently audited error rate of 1.72%. Source code and dataset annotations are available from the corresponding author upon request to facilitate replication and community benchmarking.

The primary limitations of the current implementation are its susceptibility to severe motion blur artifacts and constrained performance in simultaneous multi-subject recognition scenarios at close proximity. Prospective research directions include: (i) replacing the Haar Cascade detector with deep learning-based alternatives such as MTCNN or Retina Face for improved adversarial condition robustness; (ii) incorporating anti-spoofing liveness detection to guard against presentation attacks; (iii) scaling the system to multi-camera network configurations suitable for large lecture venues; and (iv) evaluating compact CNN architectures such as MobileNetV3 to achieve deep-learning-level recognition accuracy while satisfying the real-time CPU-only constraint.

8. References

- [1] Agarwal, S., & Mehta, A. (2023). Predicting mental health risk from social media behaviour using ensemble machine learning models. *Journal of Biomedical Informatics*, 138, 1–12.
- [2] Zhang, Z., & Li, B. (2023). Deep learning approaches for depression detection using social media and natural language processing. *Neural Networks*, 158, 1–13.
- [3] Hassan, M., Alam, F., & Islam, T. (2023). Comparative analysis of machine learning classifiers for mental health prediction using social media data. *IEEE Access*, 11, 23541–23558.
- [4] Liu, Y., Wang, Z., & Chen, H. (2023). Transformer-based deep learning models for depression detection on social media platforms. *Expert Systems with Applications*, 213, 1–14.
- [5] Sharma, N., Gupta, P., & Verma, A. (2023). Social media addiction and mental health risk detection using Instagram behavioural features. *International Journal of Environmental Research and Public Health*, 20(4), 1–18.
- [6] Bhatt, R., Patel, S., & Joshi, K. (2023). Impact of excessive Instagram use on anxiety and depression among adolescents: A data-driven study. *Journal of Affective Disorders*, 320, 412–421.
- [7] Van Rossum, G. (2023). Python language reference (Version 3.11). Python Software Foundation. <https://www.python.org>