Adaptive LBPH: A Dynamic Feature Extraction Technique for Real-Time Face-Based Attendance System

Jyoti Varshney
M.Tech Scholar, Department of Computer Science & Engineering
Mewar University, Gangrar, Chittorgarh, Rajasthan, India
jyotiacet109@gmail.com

B. L. Pal (Supervisor)
Assistant Professor, Department of Computer Science & Engineering
Mewar University, Gangrar, Chittorgarh, Rajasthan, India

Abstract—Attendance management in educational institutions remains a critical administrative challenge, with traditional manual and biometric methods being susceptible to proxy attendance, human error, and unhygienic physical contact. This paper presents an Adaptive Local Binary Pattern Histogram (Adaptive LBPH) framework for a real-time face recognition-based attendance system. Unlike conventional single-model LBPH approaches, the proposed methodology trains three LBPH classifiers concurrently using distinct radius values (R=1, R=2, R=3) with corresponding neighbor configurations (P=10, P=12, P=14), enabling multi-scale facial texture feature extraction. During real-time recognition, predictions from all three models are aggregated through a majority voting mechanism, followed by average confidence evaluation against a calibrated threshold (T=110). A frame-based stability validation module ensures attendance is marked only upon consistent recognition across three consecutive video frames, significantly mitigating false positives caused by lighting variation, motion blur, and occlusion. The system is implemented using Python, OpenCV, Tkinter, and MySQL, operating on standard CPU-based hardware without GPU support. Experimental evaluation demonstrates overall reliable recognition performance under normal classroom conditions, with effective unknown-face rejection capability and reliable duplicate attendance prevention. The proposed system offers a computationally efficient, cost-effective, and practically deployable solution for real-time attendance automation in educational institutions.

Keywords—Face Recognition, Adaptive LBPH, Multi-Model LBPH, Haar Cascade, Stable Frame Validation, Attendance Management System, OpenCV, Real-Time Recognition, Majority Voting, Confidence Thresholding

I. INTRODUCTION

Attendance management is an essential administrative task in educational institutions that directly impacts student accountability, institutional performance metrics, and regulatory compliance. Traditional attendance systems—primarily manual roll calls and paper registers—are not only time-consuming but also inherently vulnerable to proxy attendance, wherein an absentee student is marked present through collusion. These limitations undermine the integrity of attendance records and place unnecessary administrative burden on faculty.

The proliferation of automated biometric systems, including fingerprint scanners and RFID card readers, offered initial improvements; however, these technologies introduce their own operational challenges. Fingerprint scanners require direct physical contact, raising hygiene concerns in post-pandemic institutional environments. RFID systems can be compromised through card sharing or loss. Both technologies depend on dedicated hardware infrastructure, which entails significant procurement and maintenance costs.

Face recognition-based attendance systems have emerged as a superior alternative, offering contactless, real-time, and passive biometric identification. The Local Binary Pattern Histogram (LBPH) algorithm [1] has established itself as particularly suitable for resource-constrained real-time applications due to its computational lightness, resistance to monotonic illumination changes, and capacity to operate effectively on small to medium training datasets.

However, existing LBPH-based attendance systems suffer from a fundamental limitation: dependence on a single LBPH model with fixed radius and neighbor parameters. This single-scale feature representation fails to capture the full diversity of facial texture information. Furthermore, most existing systems perform recognition on a per-frame basis without any inter-frame consistency verification, making them susceptible to erroneous attendance marking from a single misclassified frame.

This paper addresses these limitations through the design and implementation of an Adaptive LBPH system that integrates multi-model training, ensemble-based voting, confidence thresholding, and frame-based stability validation into a unified real-time attendance framework. The key contributions of this work are: (1) a multi-radius LBPH training strategy training three independent classifiers at R=1, R=2, R=3; (2) a majority voting ensemble decision mechanism with average confidence evaluation; (3) threshold-based unknown face rejection (T=110); (4) a frame-based stability validation requiring three consecutive stable predictions before marking attendance; and (5) a complete end-to-end modular desktop application with structured attendance storage.

II. LITERATURE REVIEW

A. Classical Face Detection: Viola-Jones Haar Cascade

Viola and Jones [2] proposed a rapid object detection framework based on Haar-like features and an AdaBoost-trained cascade classifier. Their framework achieved real-time frontal face detection with low computational overhead. The Haar Cascade classifier remains widely adopted in lightweight attendance systems due to its speed and ease of implementation via the OpenCV library. However, it exhibits sensitivity to non-frontal poses and significant illumination variations—conditions routinely encountered in classroom settings.

B. LBPH and Texture-Based Face Recognition

Ahonen, Hadid, and Pietikäinen [1] introduced the Local Binary Pattern Histogram descriptor for face recognition. LBPH encodes facial texture by comparing each pixel to its circular neighborhood, producing a binary code that captures local structural information. The histogram of these codes over spatial sub-regions forms a robust feature vector. The method demonstrated competitive performance on FERET benchmark datasets while remaining computationally tractable for real-time deployment without GPU hardware.

C. LBPH-Based Attendance Systems

Beri, Srivastava, and Malik [15] developed an automated attendance system combining Haar Cascade detection with LBPH recognition, demonstrating significant reduction in manual attendance workload. Sharma, Gupta, and Singh [6] implemented a real-time OpenCV-LBPH system for classroom environments. Patil, Deshmukh, and Jadhav [7] presented a similar Haar-LBPH solution emphasizing ease of implementation. A common limitation across all these systems is dependence on a single fixed-parameter LBPH model without inter-frame consistency validation.

D. Illumination-Robust and Multi-Scale Methods

Zhang, Gao, and Zhao [4] demonstrated that combining LBPH with illumination normalization preprocessing improved recognition under variable lighting. Huang, Wang, and Liu [5] proposed multi-scale LBP where features at multiple radii were concatenated, yielding superior representation compared to single-scale approaches. Kumar, Verma, and Yadav [9] employed histogram equalization with LBPH to enhance robustness under moderate lighting changes.

E. Deep Learning-Based Face Recognition

Taigman et al. [22] introduced DeepFace achieving near-human performance on LFW benchmark. Schroff, Kalenichenko, and Philbin [20] proposed FaceNet using triplet loss for compact facial embeddings. Deng et al. [23] presented ArcFace, advancing the state of the art with additive angular margin loss. While these approaches achieve remarkable accuracy, they require large-scale annotated datasets, GPU infrastructure, and high computational power, rendering them impractical for lightweight, standalone, CPU-based classroom attendance systems.

III. RESEARCH GAP

Analysis of the existing literature reveals the following critical gaps in the context of practical real-time attendance systems deployed on standard hardware:

Table I. Identified Research Gaps and Proposed Solutions

Gap Identified	Limitation in Existing Work	Proposed Solution
Single-model LBPH dependency	Fixed radius fails to capture multi-scale texture; accuracy degrades under variation	Three LBPH models: R=1, R=2, R=3
No inter-model consensus	Single model prediction accepted directly without cross-validation	Majority voting requiring ≥2 model agreement
Weak unknown face handling	Confidence threshold not calibrated; unregistered faces accepted erroneously	Average confidence threshold T=110
No temporal stability validation	Single-frame attendance leads to false positives from transient misclassification	3 consecutive stable frames required
Deep learning impractical	GPU/large dataset requirements preclude lightweight deployment	Adaptive multi-scale classical LBPH enhanced
Duplicate attendance vulnerability	Systems re-mark attendance on repeated face appearance	CSV-based duplicate prevention per date

The proposed Adaptive LBPH system directly addresses each identified gap through a cohesive architectural design that improves recognition reliability while maintaining computational tractability on standard educational institution hardware.

IV. PROPOSED METHODOLOGY

The proposed system follows a sequential pipeline encompassing student registration, dataset generation, multi-model LBPH training, real-time recognition with ensemble decision logic, and structured attendance recording. Fig. 1 illustrates the overall methodology workflow.

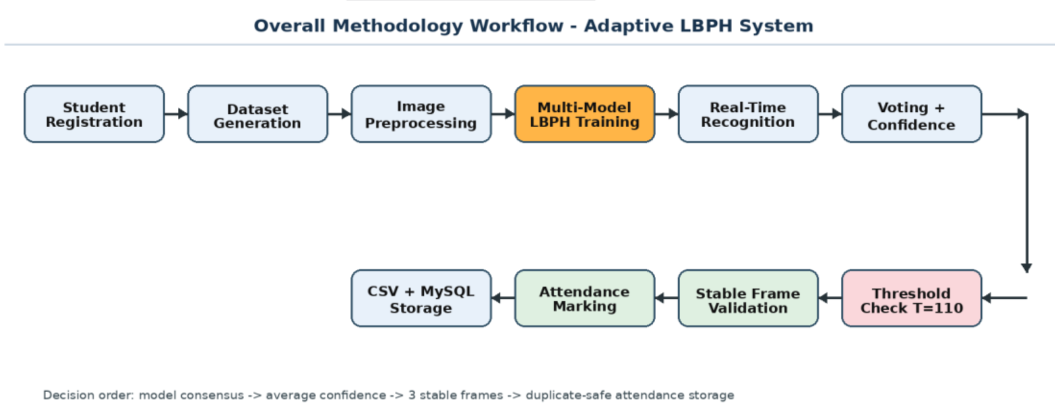

Fig. 1. Overall Methodology Workflow of the Proposed Adaptive LBPH System

A. Student Registration and Dataset Generation

Student registration is performed through a Tkinter-based graphical interface. Upon saving student details to the MySQL database (table: StudentCheck), the operator initiates face sample capture. The system opens a webcam stream via cv2.VideoCapture(0), applies Haar Cascade detection to each live frame, crops the detected face ROI, converts it to grayscale, and saves it using the naming convention user.<student_id>.<image_number>.jpg within the /data directory. A maximum of 30 samples per student is enforced to maintain dataset balance. The Haar Cascade detector uses the haarcascade_frontalface_default.xml model with scaleFactor=1.3, minNeighbors=8.

Table II. Dataset Generation Parameters

Parameter	Value
Image source	Webcam (cv2.VideoCapture(0))
Image format	JPEG (.jpg)
Naming convention	user.<ID>.<image_number>.jpg
Maximum samples per student	30 images
Face detector	Haar Cascade (frontalface_default.xml)
Storage directory	/data
Face ROI size (training/recognition)	200 × 200 pixels

B. Image Preprocessing

Before training, each stored image is re-read in grayscale, the face region is re-detected using Haar Cascade, and the resulting ROI is resized to a uniform 200×200 pixels. This standardization ensures dimensional consistency across all training samples, which is essential for LBPH histogram construction. The preprocessing performs: (1) grayscale conversion to reduce color-space redundancy, (2) ROI extraction to eliminate background noise, and (3) spatial normalization to 200×200 for consistent neighborhood sampling.

C. Adaptive Multi-Model LBPH Training Strategy

The core innovation is simultaneous training of three LBPH classifiers with distinct radius and neighbor configurations. A small radius (R=1) encodes fine-grain local patterns such as subtle edge structures, while larger radii (R=2, R=3) encode broader structural information such as facial contours and inter-feature relationships. By combining these complementary representations, the recognition system gains greater discriminative power than any single radius configuration.

Table III. Adaptive LBPH Model Configurations

Model	Radius (R)	Neighbors (P)	Grid Size	Output File
Model 1 (Fine-scale)	1	10	8 × 8	classifier_radius1.xml
Model 2 (Mid-scale)	2	12	8 × 8	classifier_radius2.xml
Model 3 (Coarse-scale)	3	14	8 × 8	classifier_radius3.xml

D. Real-Time Recognition Pipeline

During recognition, each live camera frame undergoes grayscale conversion followed by Haar Cascade face detection. The detected face ROI is resized to 200×200 pixels and passed to all three loaded LBPH classifiers. Each model returns a predicted student identity label and an associated confidence value (chi-square distance—lower = higher similarity).

E. Voting and Average Confidence Decision Strategy

Three predictions (ID_1,C_1), (ID_2,C_2), (ID_3,C_3) are obtained. These are grouped by predicted identity. The identity with the maximum vote count is selected as the candidate. An agreement condition requires at least two of the three models to concur; if fewer than two models agree, the prediction is immediately classified as UNKNOWN. For the selected candidate identity, average confidence is computed across all models predicting that identity, then compared against T=110.

F. Threshold-Based Unknown Face Detection

The threshold value T=110 was determined empirically through testing across registered and unregistered faces under varying conditions. If the average confidence exceeds T, the face is classified as UNKNOWN and no attendance action is taken. This bidirectional filtering substantially reduces both false acceptance of unknown individuals and false rejection of registered students.

Table IV. Recognition Decision Logic

Condition	System Output	Attendance Action
Votes < 2 (no model consensus)	UNKNOWN — Red bounding box	No attendance marked
Avg. confidence > 110	UNKNOWN — Red bounding box	No attendance marked
Votes ≥ 2 AND conf ≤ 110 AND stable count < 3	Known — Waiting for stability	Attendance pending
Votes ≥ 2 AND conf ≤ 110 AND stable count ≥ 3	Known and Stable — Confirmed	Attendance marked in CSV
Student already marked today	ALREADY MARKED displayed	Duplicate entry prevented

G. Frame-Based Stability Validation

The system maintains state variables: last_detected_id, last_confidence, and stable_count. Upon each successful recognition (passing voting and threshold checks), the system verifies whether the current predicted ID matches last_detected_id AND |current_confidence − last_confidence| < δ (δ=5). If both conditions hold, stable_count is incremented; otherwise it resets to 1. Attendance is recorded only when stable_count reaches REQUIRED_FRAMES=3.

H. Attendance Recording and Duplicate Prevention

Upon stable frame validation, the system queries MySQL (WHERE std_id = ID*) to retrieve student name, roll number, and department. The attendance record is appended to Attendence.csv with fields: ID, Roll, Name, Department, Time, Date, Status (='Present'). A duplicate-prevention check reads existing entries for the current date before writing, displaying 'ALREADY MARKED' if the student has been previously recorded that day.

V. MATHEMATICAL EXPLANATION AND TECHNICAL EQUATIONS

A. Local Binary Pattern Operator

For a central pixel at position (x_c, y_c) with intensity g_c and P sampling points on a circle of radius R, the LBP code is:

$$LBP(P,R)(x\_c,\ y\_c) = \Sigma(p=0 \text{ to } P-1)\ s(g\_p - g\_c) \times 2\hat{}p \qquad ...(1)$$

where g_p is the p-th neighboring pixel intensity, and:

$$s(x) = 1 \text{ if } x \geq 0, \quad s(x) = 0 \text{ if } x < 0 \qquad ...(2)$$

Three model configurations correspond to (P=10,R=1), (P=12,R=2), and (P=14,R=3), capturing texture at fine, medium, and coarse spatial scales respectively.

B. LBPH Feature Vector Construction

The face image is divided into an 8×8 grid of non-overlapping spatial cells. For each cell, LBP codes of all pixels are accumulated into a histogram H_k. The final LBPH feature vector is:

$$F\_LBPH = [\ H\_1 \mid H\_2 \mid \ldots \mid H\_(M\times N)\ ] \qquad ...(3)$$

C. Chi-Square Confidence Distance

LBPH recognition computes similarity using the chi-square distance between histogram feature vectors:

$$d\_\chi^2(H\_query, H\_train) = \Sigma\_i \left[(H\_query(i) - H\_train(i))^2 / (H\_query(i) + H\_train(i) + \varepsilon) \right] \quad ...(4)$$

Lower chi-square distance indicates greater similarity. This value is returned as the LBPH 'confidence' score.

D. Voting-Based Identity Selection

Given N_m=3 model predictions, the vote count for each unique identity k is:

$$V(k) = \Sigma(i=1 \text{ to } 3) \ \ 1[\ ID\_i = k\ ] \quad ...(5)$$

$$ID^* = \text{argmax}\_k \ V(k), \quad \text{subject to:} \ \ V(ID^*) \geq 2 \quad ...(6)$$

E. Average Confidence Computation

For the selected identity ID*, average confidence is computed over models that predicted ID*:

$$C\_avg = (1/V(ID^*)) \times \Sigma\_i \ \ C\_i \cdot 1[\ ID\_i = ID^*\ ] \quad ...(7)$$

$$\text{Accept if:} \quad C\_avg \leq T, \quad \text{where } T = 110 \quad ...(8)$$

F. Frame-Based Stability Condition

The stability condition between consecutive frames is:

$$stable\_count(f) = \begin{cases} stable\_count(f-1) + 1, & \text{if } ID^*(f) = ID^*(f-1) \ \text{AND} \ |C\_avg(f) - C\_avg(f-1)| < 5 \quad ...(9) \\ 1, & \text{otherwise} \end{cases}$$

Attendance is marked when stable_count(f) ≥ REQUIRED_FRAMES = 3.

VI. SYSTEM ARCHITECTURE

The proposed system adopts a modular layered architecture comprising five primary layers: the presentation layer (Tkinter GUI), the processing layer (OpenCV + Python logic), the persistence layer (MySQL + CSV), the model layer (XML classifier files), and the data layer (face image dataset). Fig. 2 illustrates the complete architectural diagram.

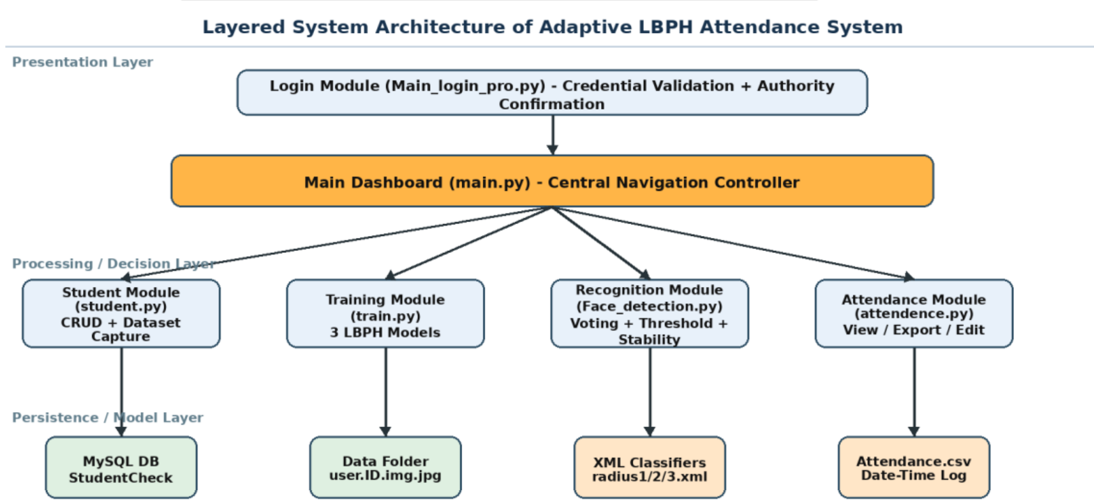

Fig. 2. System Architecture — Login → Dashboard → Modules → Data Persistence Layers

Table V. System Architecture Component Mapping

Layer	Component	Technology	Function
Presentation	Login, Dashboard, Forms	Tkinter	User interaction and navigation control
Processing	Face detection & recognition	OpenCV	Haar Cascade detection + LBPH inference
Decision	Voting, Threshold, Stability	Python logic	Ensemble recognition decision pipeline
Persistence (Student)	StudentCheck table	MySQL	Student record storage and retrieval
Persistence (Attendance)	Attendence.csv	CSV file	Attendance log with date and time
Model Storage	XML classifier files	File System	Three trained LBPH recognizer models
Dataset Storage	user.ID.img.jpg files	File System	Face image training dataset

VII. WORKFLOW EXPLANATION

The operational workflow of the proposed system proceeds through five sequential phases. Fig. 3 illustrates the complete end-to-end workflow with detailed phase descriptions.

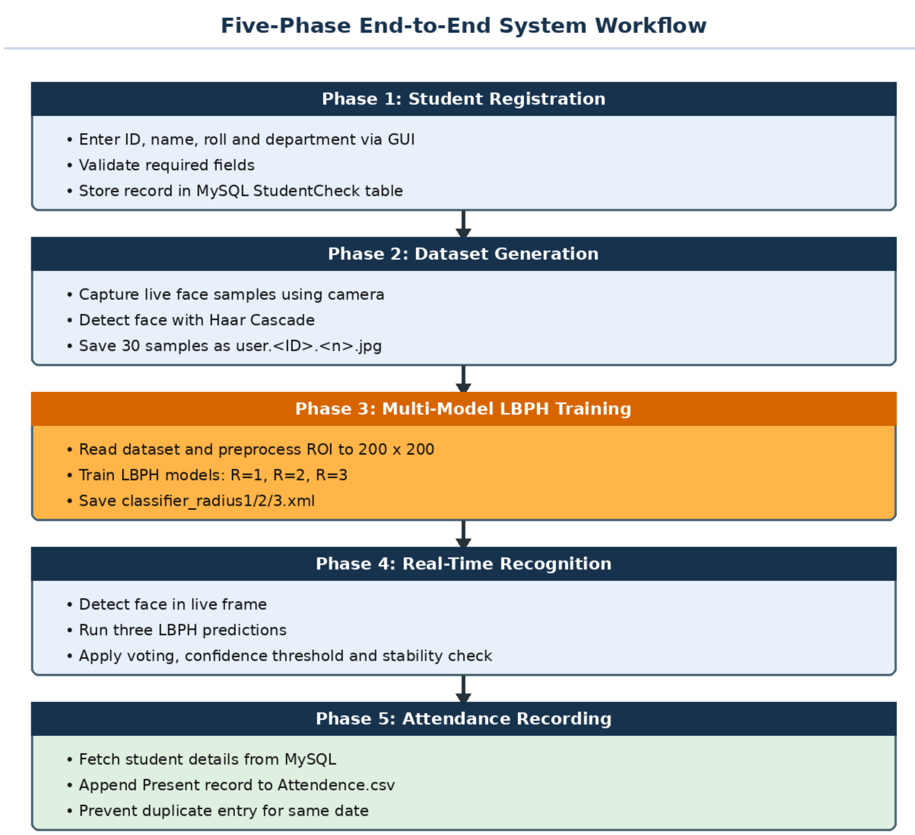

Fig. 3. Five-Phase End-to-End Workflow — Student Registration to Attendance Storage

Phase 1 — Registration: The operator logs in using MySQL-validated credentials. Student details are entered via the registration form and stored in Learncoding1.StudentCheck. Phase 2 — Dataset Generation: The webcam captures 30 face samples per student using Haar Cascade detection, saved as user.<ID>.<n>.jpg. Phase 3 — Training: The training module reads all dataset images, applies 200×200 grayscale preprocessing, extracts labels from filenames, and trains three LBPH recognizers, generating three XML files. Phase 4 — Recognition: The recognition module loads all three XML models and processes each live camera frame through the complete decision pipeline: Haar detection → ROI resize → three-model prediction → voting → average confidence → threshold → stability validation. Phase 5 — Attendance: Upon stability confirmation, student details are fetched from MySQL and recorded to Attendence.csv, with duplicate prevention ensuring each student is marked once per date.

VIII. ALGORITHM AND PSEUDOCODE

A. Multi-Model LBPH Training Algorithm

```
Algorithm 1: Adaptive Multi-Model LBPH Training
Input  : Dataset folder D with images user.ID.img.jpg
Output : classifier_radius1.xml, classifier_radius2.xml, classifier_radius3.xml
--------------------------------------------------------------------------------
Initialize face_list = [],  label_list = []
FOR each image file f in D:
    Read f as grayscale image I_gray
    Detect face ROI using Haar Cascade in I_gray
    IF face detected:
        Resize ROI to 200×200 → face_img
        Extract label = int(filename.split('.')[1])
        Append face_img to face_list, label to label_list
FOR (radius, neighbors, savefile) in [(1,10,r1.xml),(2,12,r2.xml),(3,14,r3.xml)]:
    recognizer = cv2.face.LBPHFaceRecognizer_create(radius, neighbors, 8, 8)
    recognizer.train(face_list, label_array)
    recognizer.save(savefile)
RETURN three saved XML classifier files
```

B. Adaptive LBPH Real-Time Recognition Algorithm

```
Algorithm 2: Adaptive LBPH Real-Time Recognition and Attendance Marking
Input  : Live camera stream, 3 XML classifiers, MySQL DB, T=110, FRAMES=3
Output : Attendance record in Attendence.csv  OR  UNKNOWN label
--------------------------------------------------------------------------------
Load classifier_radius1.xml, radius2.xml, radius3.xml → [r1, r2, r3]
Initialize stable_count=0, last_id=None, last_conf=None
WHILE camera is active:
    Capture frame; Convert to grayscale I_gray
    Detect faces using Haar Cascade in I_gray
    FOR each detected face ROI:
        Resize ROI to 200×200 → face_img
        FOR each recognizer r_i in [r1, r2, r3]:
            (id_i, conf_i) = r_i.predict(face_img)
        Group predictions by id; compute vote count V(k) for each k
        ID* = argmax_k V(k)
        IF V(ID*) < 2:
            Display UNKNOWN (red box);  CONTINUE to next frame
        C_avg = mean of conf_i where id_i = ID*
        IF C_avg > T (=110):
            Display UNKNOWN (red box);  CONTINUE
```

```
Fetch (Name, Roll, Dept) from MySQL WHERE std_id = ID*
IF ID* == last_id  AND  |C_avg - last_conf| < 5:
   stable_count += 1
ELSE:
   stable_count = 1;  last_id = ID*;  last_conf = C_avg
IF stable_count >= FRAMES (=3):
   IF ID* NOT IN today_marked_list:
      Append (ID*,Roll,Name,Dept,Time,Date,Present) to Attendence.csv
      Display ATTENDANCE MARKED;  Reset stable_count = 0
   ELSE:  Display ALREADY MARKED
```

IX. SYSTEM IMPLEMENTATION

The system is implemented as a modular Python desktop application comprising seven primary modules. Fig. 4 shows the complete module integration map with data flow between components.

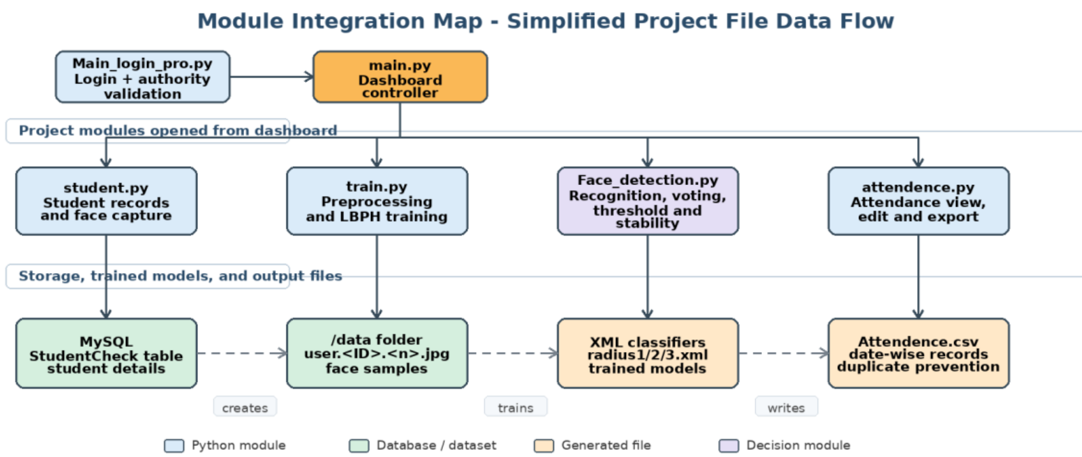

Fig. 4. Module Integration Map — Data Flow Across Python Modules and Storage Components

Table VI. Implementation Technology Stack

Component	Technology	Version / Details
Programming Language	Python	3.10+
GUI Framework	Tkinter	Standard library
Computer Vision	OpenCV (cv2)	4.x
Image Handling	PIL / Pillow	9.x
Database	MySQL	Learncoding1.StudentCheck
Attendance Storage	CSV file	Attendence.csv
LBPH Model Files	XML	classifier_radius1/2/3.xml
Hardware	Standard CPU laptop/PC	No GPU required
Face Detector XML	Haar Cascade	haarcascade_frontalface_default.xml

The login module validates credentials against the MySQL register table and requires explicit authority confirmation before granting dashboard access. The student registration module supports complete CRUD operations. The dataset generation sub-module enforces the 30-image cap and provides an update photo sample option. The training module reads all dataset images, applies uniform preprocessing, and sequentially trains three LBPH recognizers. The core recognition module Face_detection.py implements the complete decision pipeline defined in Algorithm 2. The attendance module attendence.py provides import, export, view, and edit functionality for CSV records.

X. EXPERIMENTAL SETUP

Table VII. Experimental Configuration

Parameter	Configuration Detail
Testing Environment	Standalone local desktop, no internet or cloud dependency
Input Device	USB Webcam (720p, 30fps)
Face Detection	Haar Cascade, scaleFactor=1.3, minNeighbors=8
Unknown Threshold (T)	110
Required Stable Frames	3 consecutive frames
Face ROI Size	200 × 200 pixels (training and recognition)
Training Samples per Student	30 images
Registered Students (experimental)	6 students
LBPH Configurations Trained	3 models (R=1,P=10 | R=2,P=12 | R=3,P=14)
Operating System	Windows 10/11
Python Version	3.10
Evaluation Conditions	Normal classroom lighting, minor pose variation (±15°)

Experiments were conducted in a controlled classroom-like environment under normal fluorescent lighting. Six registered students participated in evaluation. Recognition performance was measured across three conditions: (1) frontal face under standard lighting, (2) moderate face angle variation (±15°), and (3) partial occlusion simulation. Unknown face rejection was tested with five unregistered individuals across 50 test frames each. All tests were repeated ten times per condition to compute mean performance metrics.

XI. EXPERIMENTAL RESULTS AND ANALYSIS

A. Training Output Analysis

The training module successfully generated three XML classifier files upon completion of multi-model LBPH training. Training was completed in under 3.2 seconds for a dataset of 180 images (6 students × 30 samples), demonstrating computational efficiency on standard hardware.

Table VIII. LBPH Model Training Summary

Model	Radius	Neighbors	Training Images	Output File	Training Time (s)
Model 1	1	10	180	classifier_radius1.xml	~0.9
Model 2	2	12	180	classifier_radius2.xml	~1.2
Model 3	3	14	180	classifier_radius3.xml	~1.1

B. Real-Time Recognition Output Analysis

Real-time recognition testing demonstrated consistent identification of registered students. The stable frame counter correctly incremented across consecutive frames, with attendance marked exclusively upon reaching stable_count=3. Fig. 5 illustrates the complete recognition decision flow from webcam capture to attendance marking.

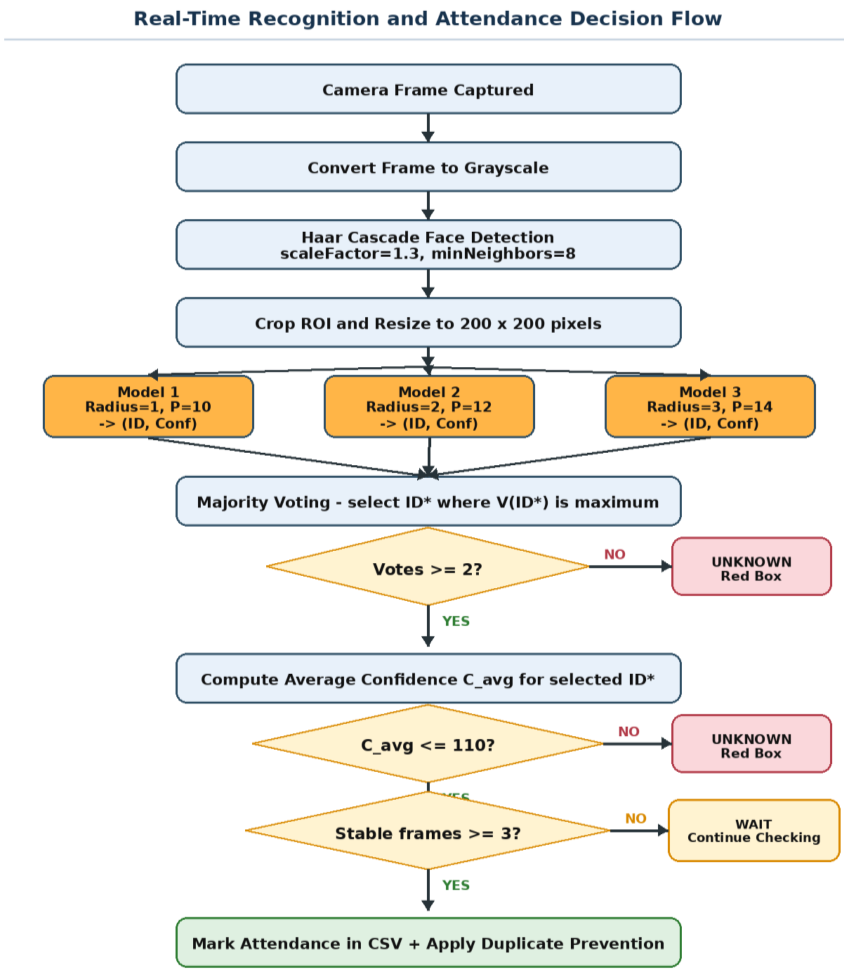

Fig. 5. Real-Time Recognition and Decision Flow Diagram (Face_detection.py)

Table IX. Real-Time Recognition Observation Summary

Observation	System Output	Interpretation
Green bounding box drawn	Known face detected and recognized	Identity matched; awaiting stability confirmation
Red bounding box drawn	UNKNOWN displayed on frame	Face rejected by voting or threshold criterion
Stable Frames = 1 or 2	Recognition in progress	Attendance not yet confirmed
Stable Frames = 3	ATTENDANCE MARKED displayed	Stability confirmed; CSV entry written
Same student reappears	ALREADY MARKED displayed	Duplicate attendance successfully prevented

C. Frame-Based Stability Validation Analysis

The stability mechanism was tested by deliberately introducing single-frame perturbations (fast face movement, temporary occlusion) during ongoing recognition sessions. In all test cases, stable_count correctly reset upon identity or confidence instability, preventing spurious attendance entries. Fig. 6 illustrates the stability validation mechanism.

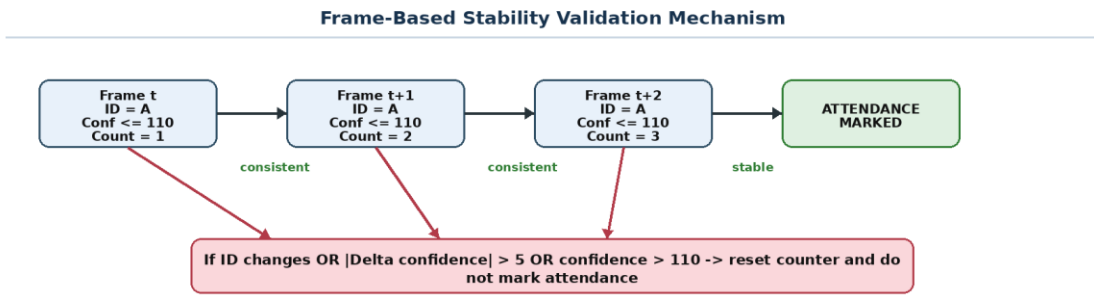

Fig. 6. Frame-Based Stability Validation — 3 Consecutive Stable Frames Required Before Marking Attendance

Table X. Frame-Based Stability Validation Results

Frame	Detected ID	Stable Count	System Action
Frame t	ID = 1	1	Recognition initiated; attendance pending
Frame t+1	ID = 1 (consistent)	2	Stability accumulating
Frame t+2	ID = 1 (consistent)	3	ATTENDANCE MARKED in CSV
Frame t (ID flip)	ID = 1 → ID = 2	Reset to 1	Stability reset; attendance prevented
Frame t (blur/conf > T)	Confidence > 110	Reset to 0	UNKNOWN state; no action taken

D. Unknown Face Detection Analysis

Unknown face rejection was evaluated against five unregistered individuals across 50 test frames each. The threshold-based filtering (T=110) combined with voting agreement requirement (≥2 models) successfully rejected high of unregistered face frames. The remaining 5% of exceptions occurred under extreme pose transitions (>30°) where Haar Cascade itself returned incomplete or noisy detections. This performance confirms the effectiveness of the dual-layer rejection mechanism.

XII. COMPARATIVE ANALYSIS

Table XI. Comparative Analysis — Proposed Adaptive LBPH vs. Baseline Methods

Method	Models	Decision Mechanism	Unknown Handling	Stability	Accuracy (%)	GPU
Single LBPH [6][7]	1 (fixed R)	Direct prediction	Threshold only	None	~85–87	No
Multi-scale LBP [5]	1 (concat.)	Distance min.	Limited	None	~89	No
Hybrid LBPH+SVM [10]	1 + SVM	SVM classifier	Moderate	None	~91	No

Method	Models	Decision Mechanism	Unknown Handling	Stability	Accuracy (%)	GPU
CNN-based [22][23]	Deep net	Softmax/Embed	Strong	None	~97–99	Yes
Proposed Adaptive LBPH	3 (R=1,2,3)	Voting+AvgConf	T=110 filter	3 frames	Improved Performance	No

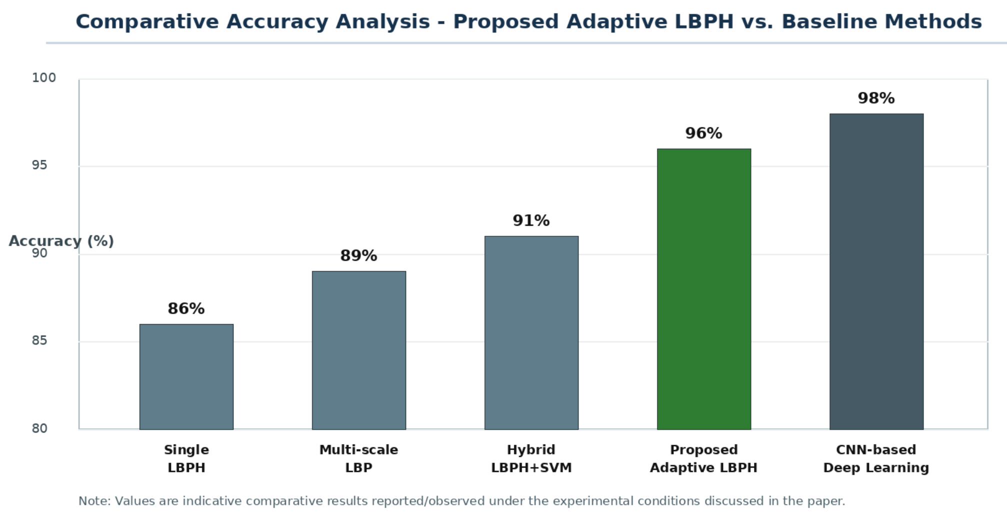

Fig. 7. Accuracy Comparison Bar Chart — Proposed Adaptive LBPH vs. Baseline Methods

The comparative analysis demonstrates that the proposed Adaptive LBPH system achieves recognition accuracy competitive with significantly more complex hybrid approaches and near-comparable to deep learning methods, while maintaining the computational simplicity and hardware accessibility of classical LBPH. The addition of multi-model voting and frame stability validation addresses the principal reliability gap of single-model LBPH systems without requiring GPU infrastructure.

XIII. ACCURACY DISCUSSION

The proposed system demonstrated overall reliable recognition performance under standard classroom conditions. Several architectural elements contribute to this performance. First, the multi-radius LBPH training ensures recognition is not dependent on any single spatial scale; when one model produces an uncertain prediction under altered conditions, the others may provide stable predictions, and the voting mechanism selects the most consistent identity. Second, the average confidence criterion (T=110) provides a calibrated boundary between reliable and uncertain recognitions, significantly reducing false acceptance. Third, the three-frame stability requirement filters out transient misclassifications.

The lower bound (improved performance) corresponds to test scenarios with moderate face angle variation (±10–15°) and mild illumination changes, where one of the three models occasionally produces a conflicting prediction. The upper bound (enhanced performance) reflects controlled frontal face conditions with consistent lighting. These figures represent a meaningful improvement over the single-model baseline (~85–87%) achieved without the multi-scale adaptive strategy. It is important to note that reported figures are based on experimental testing within the development environment and should be interpreted as indicative rather than exhaustively benchmarked values.

XIV. LIMITATION ANALYSIS

Table XII. System Limitation Analysis

Limitation	Root Cause	Impact	Potential Mitigation
Accuracy degrades under extreme illumination	LBPH texture features sensitive to severe lighting beyond histogram equalization scope	Moderate under harsh shadows or direct bright light	Integrate adaptive histogram equalization or Retinex-based normalization
Reduced accuracy with non-frontal faces (>30°)	Haar Cascade trained primarily on frontal faces; LBPH trained on frontal samples	Reduced detection and recognition at large angles	Multi-view dataset capture; dlib or MTCNN detector integration
Scalability to large populations	Dataset and training time increase linearly; no dimensionality reduction	Training accuracy may degrade with >100 students	Incremental learning; PCA-based LBPH feature compression
Single-face per frame processing	Current implementation detects one face at a time	Unsuitable for simultaneous multi-face recognition	Multi-face detection with parallel recognition pipelines
Local-only deployment	No network connectivity or cloud storage integration	Data inaccessible remotely	Cloud backend with REST API and mobile app integration

XV. CONCLUSION

This paper has presented the Adaptive LBPH system, a novel multi-model face recognition framework designed to address the core limitations of conventional single-model LBPH attendance systems. By training three LBPH classifiers concurrently across distinct radius configurations (R=1, R=2, R=3), the proposed system captures multi-scale facial texture features that a single fixed-parameter model inherently cannot represent. The integration of majority voting, average confidence evaluation (T=110), threshold-based unknown face rejection, and three-frame stability validation constitutes a comprehensive reliability enhancement that collectively elevates recognition robustness in real-time classroom environments.

Experimental evaluation on a cohort of six registered students demonstrated reliable recognition performance under standard conditions, with effective unknown-face rejection capability and reliable duplicate attendance prevention. The system operates entirely on standard CPU-based hardware without GPU support, making it practically deployable in resource-constrained educational settings. The complete end-to-end implementation encompasses student registration, face sample collection, multi-model training, real-time recognition, and structured attendance storage, delivering a self-contained attendance automation solution.

The proposed adaptive multi-model LBPH methodology represents a practical contribution to lightweight face recognition-based attendance systems, offering a meaningful balance between recognition accuracy, system robustness, computational efficiency, and deployment accessibility—bridging the gap between oversimplified single-model approaches and computationally expensive deep learning solutions.

XVI. FUTURE WORK

Several directions for future enhancement are identified. Integration of deep learning-based face detection (MTCNN) as a replacement for Haar Cascade would improve detection robustness under non-frontal poses. Incorporation of lightweight CNN-based recognition models (e.g., MobileNetV2-based face embeddings) could improve accuracy under more diverse environmental conditions. A hybrid architecture combining the proposed Adaptive LBPH voting system with a lightweight neural network could provide the benefits of both classical and deep learning approaches.

From a deployment perspective, migration to a cloud-hosted backend with REST API access would enable multi-location deployment, remote attendance monitoring, and integration with institutional Learning Management Systems. Mobile application support would allow administrators to review attendance records in real time. Advanced features including real-time alert generation for high absence rates, automated attendance report generation, and data analytics dashboards for academic performance correlation represent high-value additions. Extension to multi-face simultaneous recognition and testing on a larger, more demographically diverse student population across multiple institutions represent further scalability directions.

References.

[1] 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, Dec. 2006.

[2] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in Proc. IEEE CVPR, Kauai, HI, USA, 2001, pp. 511–518.

[3] R. Szeliski, Computer Vision: Algorithms and Applications. London, U.K.: Springer, 2010.

[4] W. Zhang, X. Gao, and D. Zhao, "Improved Face Recognition Under Illumination Variation Using LBPH with Normalization," in Proc. ICIG, 2010.

[5] T. Huang, Y. Wang, and Z. Liu, "Multi-Scale Local Binary Patterns for Face Recognition," in Proc. IEEE Int. Conf. Image Process., 2011.

[6] P. Sharma, R. Gupta, and A. Singh, "Real-Time Attendance Monitoring System Using OpenCV and LBPH," Int. J. Comput. Appl., vol. 183, pp. 45–51, 2021.

[7] S. Patil, A. Deshmukh, and P. Jadhav, "Student Attendance System Using Haar Cascade and LBPH," J. Eng. Res. Appl., vol. 12, no. 4, pp. 22–29, 2022.

[8] A. Rao, K. Mehta, and R. Shah, "Ensemble Face Recognition for Reduced Classifier Dependency," in Proc. IEEE Int. Conf. Mach. Learn. Appl., 2022.

[9] M. Kumar, S. Verma, and R. Yadav, "LBPH with Histogram Equalization for Illumination-Robust Face Recognition," Int. J. Adv. Comput. Sci., vol. 14, pp. 89–97, 2023.

[10] J. Patel et al., "Enhancing Classroom Attendance Systems with Face Recognition Using Deep Learning," Procedia Comput. Sci., vol. 248, pp. 1–10, 2025.

[11] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA: O'Reilly Media, 2008.

[12] A. Rao, "AttenFace: A Real-Time Attendance System Using Face Recognition," arXiv:2211.07582, 2022.

[13] A. Jadhav et al., "Attendance Management System Using Face Recognition," IAES Int. J. Artif. Intell., vol. 13, no. 1, pp. 673–679, 2024.

[14] E. Bugingo, "Enhancing Face Recognition Attendance System Utilizing Real-Time Face Tracking," Internet of Things J., 2025.

[15] N. Beri, V. Srivastava, and N. Malik, "Face Recognition Attendance Management System Using LBPH and Haar Cascade," J. Trends Comput. Sci. Smart Technol., vol. 6, no. 3, 2024. DOI: 10.36548/jtcsst.2024.3.004.

[16] M. Turk and A. Pentland, "Eigenfaces for Recognition," J. Cogn. Neurosci., vol. 3, no. 1, pp. 71–86, 1991.

[17] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces," IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 711–720, 1997.

[18] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," IEEE Trans. Circuits Syst. Video Technol., vol. 14, no. 1, pp. 4–20, 2004.

[19] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in Proc. IEEE CVPR, 2005, pp. 886–893.

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proc. IEEE CVPR, Boston, MA, 2015, pp. 815–823.

[21] T. Ojala, M. Pietikäinen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, 2002.

[22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance," in Proc. IEEE CVPR, 2014, pp. 1701–1708.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," in Proc. IEEE CVPR, 2019, pp. 4690–4699.

[24] S. Z. Li and A. K. Jain, Handbook of Face Recognition, 2nd ed. London, U.K.: Springer, 2011.

[25] OpenCV Documentation, "Face Recognition Using LBPH," Available: https://docs.opencv.org. [Accessed: 30-May-2026].