International Journal of Scientific Research in Engineering & Technology
Volume6, Issue5 (November- December 2026), PP: 25-34
https://www.doi.org/10.59256/ijsreat.20260606005
www.ijsreat.com
ISSN No: 2583-1240

Smart Attendance and Student Monitoring System

Dr. Puneeth S P[1], Jayadev H N[2], Sanjana J S[3], Prashanth K L[4], Akash B S[5]
[1]*Assistant Professor, Department of Information Science and Engineering, Bapuji Institute of Engineering & Technology, Davanagere, Karnataka, India.*
[2, 3, 4,5]*Department of Information Science and Engineering, Bapuji Institute of Engineering & Technology, Davanagere, Karnataka, India.*

To Cite this Article: *Dr. Puneeth S P, Jayadev H N, Sanjana J S, Prashanth K L,Akash B S," Smart Attendance and Student Monitoring System", International Journal of Scientific Research in Engineering & Technology, Volume 06, Issue 06, November-December 2026, PP:25-34.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Traditional classroom attendance systems rely on manual roll calls, signatures, RFID cards, or other dedicated identification mechanisms that require additional effort and may be vulnerable to proxy attendance. In addition, conventional attendance systems generally provide little information about classroom activities after attendance is recorded. This paper presents a Smart Attendance and Student Monitoring System that integrates computer vision, facial recognition, object detection, facial landmark analysis, voice-based alerting, database management, and a web-based dashboard into a unified classroom monitoring platform. The proposed system uses a camera as the primary sensing device. Student faces are detected using the YuNet face detector and recognized using the SFace face-recognition model available through OpenCV. Recognized students are automatically recorded in a SQLite-based attendance database. The system further integrates YOLOv8 object detection for identifying mobile-phone usage and MediaPipe Face Mesh for facial landmark extraction and drowsiness analysis based on the Eye Aspect Ratio (EAR). Temporal confirmation mechanisms are used to reduce false alerts caused by individual frames. Confirmed phone-use and drowsiness events additionally trigger an offline text-to-speech voice alert generated using `pyttsx3`, providing immediate in-class feedback without requiring the teacher to watch the screen. Attendance, phone-use alerts, drowsiness events, session presence, and derived engagement information are then stored in a centralized database. A Flask-based REST backend provides access to the stored information, while a web dashboard displays attendance summaries, alerts, engagement trends, student information, absentees, and session reports. Automated PDF report generation and optional email delivery further extend the system's administrative functionality. The proposed architecture provides a practical foundation for intelligent classroom management using existing camera infrastructure without requiring specialized attendance hardware.*

Keywords: *Smart Classroom; Face Recognition; Automated Attendance; YuNet; SFace; YOLOv8; MediaPipe Face Mesh; Drowsiness Detection; Phone Detection; Voice Alerts; Computer Vision; Flask; SQLite*

I. INTRODUCTION

A. Overview of Smart Classroom Systems

Educational institutions require reliable mechanisms for recording attendance and maintaining an effective classroom environment. Conventional attendance methods generally depend on teachers calling student names, maintaining registers, collecting signatures, or using dedicated identification systems. Such approaches require classroom time and administrative effort. Research on automated student attendance has increasingly investigated facial recognition as an alternative to manual attendance because it allows students to be identified directly from camera images. A systematic literature review of face-recognition-based attendance systems also reports the increasing use of computer vision and deep-learning approaches for university attendance applications. 1

Although automated attendance reduces manual work, attendance alone does not provide information about classroom events occurring during a lecture. A student may be physically present but may also be using a mobile phone or showing prolonged signs associated with drowsiness. Recent classroom-monitoring research has therefore investigated computer vision for detecting student activities and attention-related behavior. 9,10

The development of lightweight computer-vision models makes it increasingly practical to process classroom video streams using ordinary computing systems. YuNet was designed as a lightweight, millisecond-level face detector for resource-constrained applications. 4 YOLO-based models provide real-time object-detection capabilities, while MediaPipe Face Mesh provides real-time facial landmark estimation from a single camera. 8

B. Problem Statement

Traditional classroom systems generally treat attendance and classroom monitoring as separate activities. A teacher may manually record attendance while classroom behavior is monitored visually. This creates several limitations:

- Manual attendance consumes instructional time.
- Proxy attendance may be possible with conventional methods.
- Attendance records may require manual data entry and maintenance.
- Mobile-phone usage may not be continuously monitored.

• Drowsiness or reduced classroom alertness may be difficult to observe continuously.
• Detected events are not conveyed to the teacher immediately unless a screen is being monitored.
• Historical attendance and monitoring information may be scattered across different records.
• Teachers may not have a centralized dashboard for reviewing classroom sessions.

Therefore, there is a need for a unified system capable of performing attendance recognition and selected classroom-monitoring tasks using a common camera infrastructure.

C. Motivation

The motivation of the proposed system is to combine multiple computer-vision modules into a single classroom-management platform. Instead of developing independent applications for attendance, phone detection, drowsiness detection, alerting, and reporting, the proposed system integrates them into a common pipeline:

Camera → Frame Processing → Face Detection → SFace Recognition → Attendance
while simultaneously performing
Frame → YOLOv8 Phone Detection → Phone Alert
and
Face Landmarks → EAR Analysis → Drowsiness Alert,
with both alert types converging on a common notification and storage path
Alerts → Voice Alert (pyttsx3) → Event Processing → SQLite → Flask API → Dashboard → PDF Reports.
The resulting events are stored in a centralized database and presented through a teacher-oriented dashboard.

II. LITERATURE SURVEY

Existing research can broadly be divided into four areas: automated attendance, face detection and recognition, classroom behavior monitoring, and real-time object detection. Table no 1 summarizes the key contributions of the surveyed works. Face-recognition attendance research indicates that automated facial identification can reduce the limitations of manual attendance systems.

Table no 1: Comprehensive Literature Survey of Facial Recognition, Drowsiness Tracking, and Classroom Monitoring Systems

No.	Paper Title	Year	Authors	Core Focus / Findings	Application to Project	Gaps Identified
01	Real-Time Driver Drowsiness Detec-tion System Using MediaPipe and EAR	2026	A. Sharma et al.	Evaluates Eye Aspect Ratio (EAR) using 3D FaceMesh landmarks for eye closure tracking.	Informs the EAR-based drowsiness monitoring module using facial crops.	Uniform fixed thresholds can cause false flags across varying natural eye baseline ratios.
02	A WAD-YOLOv8-Based Method for Classroom Student Behavior Detection	2025	L. Han, X. Ma et al.	Enhanced YOLOv8 struc-ture for multi-scale object detection in crowded indoor rooms.	Inspired the periodic YOLO inference us-ing `yolov8s.pt` for detecting phone usage.	High GPU/CPU resource consumption when executed continuously on every frame.
03	Head Pose Estimation Using Perspective-n-Point (PnP) Algorithm	2024	R. Kumar et al.	Solves 3D pose (Yaw, Pitch, Roll) using 2D image coor-dinates and a general 3D fa-cial model.	Integrates cv2.solvePnP to gate EAR calculation when students look away or downward.	Prone to rotational drift under extreme illumination shifts.
04	Lightweight Multi-Person Tracking via Intersection over Union (IoU)	2024	S. Li, H. Wang	Provides fast spatio-temporal tracking across frames without heavy re-identification overhead.	Used to maintain identity stability and track phone bounding boxes across frames.	Temporary track failure may occur when subjects experi-ence prolonged occlusion.
05	YuNet: A Tiny Millisecond-level Face Detector	2023	W. Wu, H. Peng, S. Yu	Lightweight anchor-free de-tector designed for fast face localization on resource-constrained devices.	Employed as the pri-mary detector using FaceDetectorYN for fast multi-face bounding-box extraction.	Occasional false positives may occur on complex back-ground patterns without suit-able confidence tuning.
06	Automated Atten-dance Systems via Deep Face Represen-tations	2023	M. Dai et al.	Presents an end-to-end framework for automated class verification and database record logging.	Guided the SQLite atten-dance database schema and attendance logging logic.	Does not integrate real-time behavioral and engagement monitoring features.
07	SFace:Sigmoid-Constrained Hy-persphere Lossfor Robust Face Recognition	2021	Y. Zhong et al.	Generates normalized face embeddings using sigmoid-constrained hypersphere loss and similarity-based comparison.	Adopted through Face Recognizer SF forfeature extraction and student identity verification.	Performance can degrade un-der extreme facial occlusion or severe off-axis rotations.

III. SYSTEM OBJECTIVES

The proposed system has the following objectives:

1. **Automated Attendance:** Automatically identify enrolled students and record attendance from classroom video.
2. **Face-Based Identification:** Use computer-vision-based face detection and recognition to identify students without manual roll calls.
3. **Mobile-Phone Detection:** Detect visible mobile phones in the classroom using YOLOv8.
4. **Drowsiness Detection:** Analyze facial landmarks and eye aspect ratio to identify prolonged eye-closure patterns associated with drowsiness.
5. **Voice Alert Generation:** Announce confirmed phone-use and drowsiness events through an offline text-to-speech engine so that alerts are conveyed audibly during the session.
6. **Session-Based Monitoring:** Organize attendance, alerts, and monitoring information according to individual classroom sessions.
7. **Centralized Data Management:** Store attendance and monitoring events in an SQLite database.
8. **Teacher Dashboard:** Provide a web interface for viewing attendance, alerts, student summaries, trends, and engagement information.
9. **Automated Reporting:** Generate PDF session reports containing attendance and monitoring information.
10. **Reduced Duplicate Records:** Normalize student identifiers and prevent duplicate attendance entries within a session/day.
11. **Practical Deployment:** Support commonly available camera sources such as laptop webcams, virtual smartphone cameras, and compatible USB/IP camera configurations.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architectural Overview

The proposed system follows a modular computer-vision architecture, as shown in Fig. 1. Frames are acquired from a camera or video stream and passed to a common frame-processing stage, which distributes each frame in parallel to three branches: face detection (YuNet) followed by SFace recognition and attendance logging; phone detection (YOLOv8) producing phone-use alerts; and facial-landmark detection (MediaPipe Face Mesh) followed by EAR analysis producing drowsiness alerts. The phone and drowsiness branches converge on a voice-alert stage implemented with pyttsx3, which announces confirmed

events audibly. All branch outputs are then fused by an event-processing and session-logic layer, written to an SQLite database, exposed through a Flask REST API, and rendered on a web dashboard with automated PDF reporting.

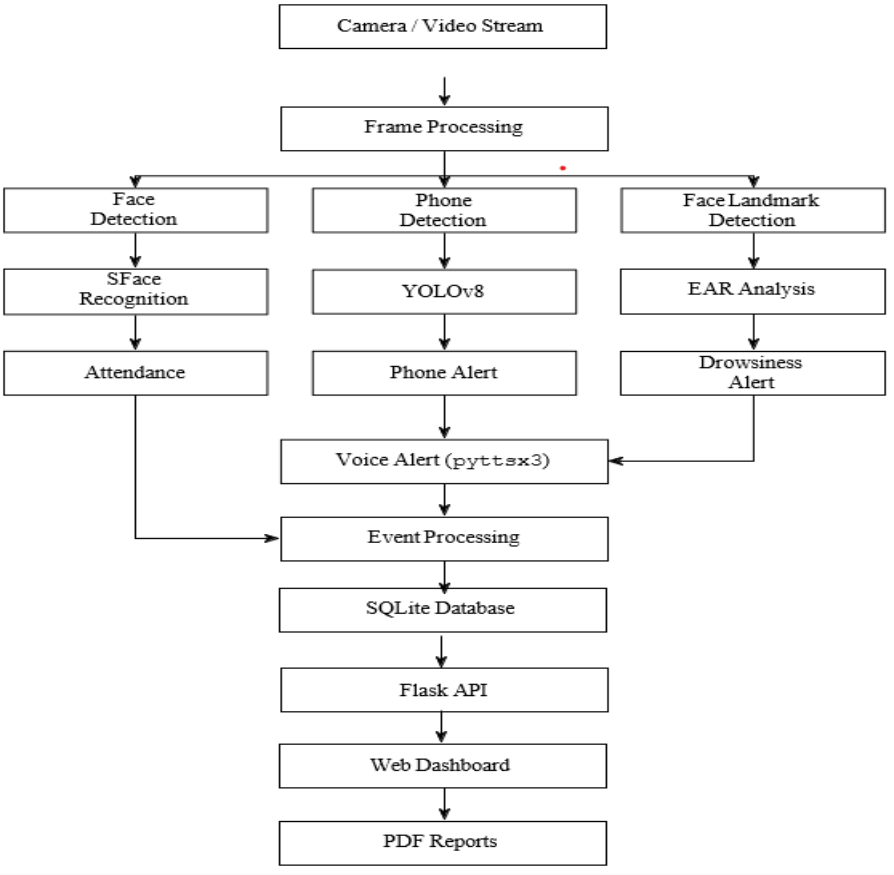


Figure 1: *Architectural overview of the proposed Smart Attendance and Classroom Monitoring System*

B. Face Detection

The system uses YuNet for face detection. YuNet is designed as a lightweight face detector suitable for real-time and resource-constrained applications. 4 For every input frame: (1) the image is acquired from the camera; (2) faces are detected; (3) detected face regions are evaluated for quality; (4) valid face regions are passed to the recognition stage. The implementation also considers factors such as face size and image blur before accepting a face for recognition.

C. Face Recognition

The system uses the SFace face-recognition implementation through OpenCV's FaceRecognizerSF interface. The recognition pipeline proceeds as:

Detected Face → Alignment/Crop → SFace Feature Extraction → Feature Comparison → Similarity Score → Student Identity

The project stores enrolled student images in student-specific folders. Student identifiers and names are normalized to reduce duplicate records caused by differences in capitalization or naming.

D.Attendance Management

After a student has been successfully recognized, the system records the student ID, student name, date, time, attendance status, and session ID. The database uses a unique session/student combination to prevent repeated attendance entries for the same student in a session. A separate session_presence table records whether a student was observed during a particular session; this distinction is useful because official daily attendance and session-level presence are different concepts in the implementation.

E.Mobile-Phone Detection

The system uses YOLOv8 for object detection, which provides object bounding boxes, class labels, and confidence scores. 6,7

The process is:

Frame → YOLOv8 Inference → Bounding Boxes → Phone Class? → Temporal Confirmation → Phone-Use Alert

The implementation also maintains separate phone-tracking states so that multiple detected phones can be handled independently. When a phone is detected near a recognized student, the system attempts a best-effort proximity association; this is an approximate association rather than a guaranteed attribution.

F.Drowsiness Detection

The system uses MediaPipe Face Mesh to obtain facial landmarks in real time from a single camera. 8 The Eye Aspect Ratio (EAR) is calculated from eye landmark coordinates using a simplified formulation:

$$EAR = \frac{p_2 - p_6 + p_3 - p_5}{2p_1 - p_4} \qquad (1)$$

where p1, p2, . . . , p6 represent eye landmark points, the vertical distances represent eye opening, and the horizontal distance represents eye width. When EAR remains below an adaptive threshold for a sufficient number of consecutive frames, the system classifies the state as potential drowsiness. The implementation also includes temporal confirmation and head-pose safeguards to reduce false positives.

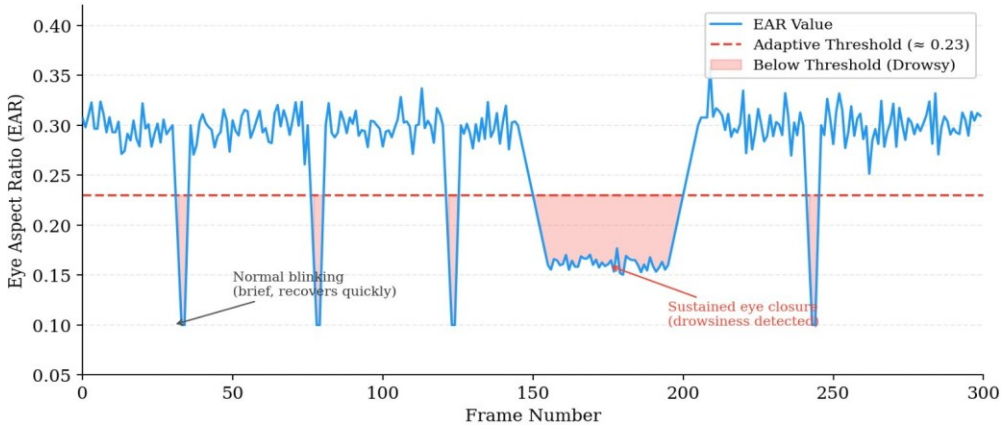


Figure 2: EAR value plotted over 300 frames illustrating the distinction between normal blinking (brief recovery) and sustained eye closure associated with drowsiness.

G.Voice Alert Generation

Confirmed phone-use and drowsiness events are passed to a voice-alert module implemented using the pyttsx3 text-to-speech library. Because pyttsx3 uses the operating system's native speech engine, alert announcements are produced offline without any network dependency. The module is invoked only after temporal confirmation, so that transient single-frame detections

do not generate audible output. Announcements are rate-limited per student so that a repeated condition does not produce continuous speech, and the alert is dispatched on a separate execution path so that speech synthesis does not block the frame-processing loop. The same confirmed event is subsequently forwarded to the event-processing layer for database storage, ensuring that the audible alert and the stored record originate from a single decision.

Mathematical and Algorithmic Framework

A. Face Recognition Similarity

Let the enrolled face feature vector be $F_e = [f_1, f_2, \ldots, f_n]$ and the feature vector extracted from the live frame be $F_l = [g_1, g_2, \ldots, g_n]$. A similarity measure can be represented generally as

$$S(F_e, F_l) = \mathrm{Sim}(F_e, F_l) \quad (2)$$

If $S(F_e, F_l) \geq T_r$, where T_r is the configured recognition threshold, the face can be accepted as the corresponding enrolled student. The actual implementation performs SFace-based feature comparison and applies a configured acceptance threshold.

B. Eye Aspect Ratio

For each eye, the EAR is computed as in Eq. (1); a low EAR corresponds to a relatively closed eye. To reduce sensitivity to individual differences, the implementation maintains adaptive/baseline information and temporal state rather than treating one low-EAR frame as sufficient evidence of drowsiness.

C. Voice Alert Triggering

Let C_e denote the temporal confirmation status of event e and $t_{last}(s, e)$ the time of the last announcement for student s and event type e. A voice alert is issued only when

$$V(s,e) = \{(1, \&"if\ "\ C_e = 1"\ and\ "\ (t - t_\mathrm{last}\ (s,e)) \geq T_c, @0, \&"otherwise"\ .)\dashv \quad (3)$$

where \(T_c\) is the announcement cool-down interval. This prevents repeated speech output for a continuing condition while still allowing the event to be logged.

D. Engagement Score

The system calculates a simple derived engagement score rather than claiming to directly measure psychological attention:

$$E = 100 - P_p\ N_p - P_d\ N_d \quad (4)$$

where E is the derived engagement score, N_p is the number of phone-related events, N_d is the number of drowsiness events, and P_p, P_d are the respective event penalties. The score is therefore a rule-based indicator derived from detected events, not a clinical or psychological measurement.

Fig. 4. Derived Engagement Score Trend Over a 50-Minute Session
(Drops correspond to detected phone-use and drowsiness events)

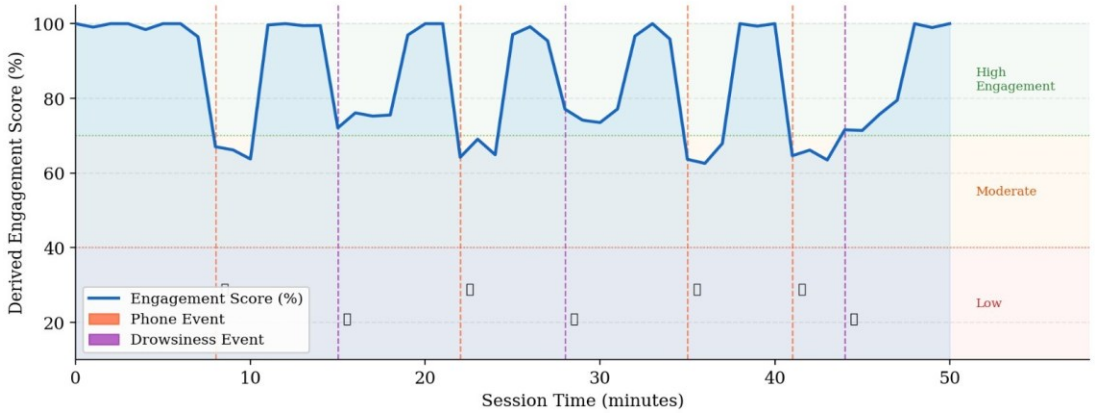


Figure 3: *Derived engagement score trend over a 50-minute session. Score drops correspond to detected phone-use and drowsiness events.*

E. Attendance Deduplication

For a student s and session k:

$$A(s,k) = \{(1, \&"if\ "\ s"\ is\ successfully\ recognized\ in\ session\ "\ k, @0, \&"otherwise"\ .)\dashv \quad (5)$$

The database enforces uniqueness for the pair (`session_id, student_id`), thereby preventing multiple attendance records for the same student within the same session.

F. Session Presence

For each student:

$$P(s,k) = \{(1, \&"if\ "\ s"\ is\ observed\ during\ session\ "\ k, @0, \&"otherwise"\ .)\dashv \quad (6)$$

This enables the system to determine session-level absentees from the enrolled-student list.

VI.ALGORITHMIC WORKFLOW

Algorithm 1 *Smart Attendance*

Start classroom session.
Capture video frame.
Detect faces using YuNet.
Evaluate face quality.
Extract SFace feature representation.
Compare representation with enrolled students.
if similarity satisfies recognition threshold **then**
    Identify student.
end if
Normalize student ID and name.
Check whether attendance already exists.
if not already recorded **then**
    Insert attendance record.
end if
Update session presence.
Continue until session ends.
Close session and generate report.

Algorithm 2 Phone Detection

Capture classroom frame.
Run YOLOv8 inference.
Extract phone bounding boxes.
Verify detection confidence.
Maintain temporal detection state.
Associate phone with nearby recognized face when possible.
Generate phone-use alert.
Announce alert using `pyttsx3` voice output.
Store event in database.

VII. EXPERIMENTAL IMPLEMENTATION

A. Software Environment

Table no 2 summarizes the software stack used in the current implementation. The repository's `requirements.txt` specifies OpenCV-Contrib, NumPy, Ultralytics, MediaPipe, Flask, Flask-CORS, FPDF2, and `pyttsx3`.

Algorithm 3 Drowsiness Detection

Detect recognized student's face.
Obtain facial landmarks using MediaPipe.
Extract left and right eye landmarks.
Calculate EAR.
Smooth EAR across frames.
Compare against adaptive threshold.
Confirm eye closure across multiple frames.
Consider head-pose safeguards.
if confirmed **then**
    Generate drowsiness alert.
    Announce alert using `pyttsx3` voice output.
end if
Store event in database.

Algorithm 4 Voice Alert Dispatch

```
Receive confirmed alert event.
Identify event type and associated student.
Check announcement cool-down for the student/event pair.
if cool-down satisfied then
    Compose alert message text.
    Speak message using pyttsx3 engine.
    Update last-announcement timestamp.
end if
Forward event to event-processing layer.
```

Table no 2: *Software Environment*

Component	Technology
Programming Language	Python
Computer Vision	OpenCV
Face Detection	YuNet
Face Recognition	SFace
Phone Detection	YOLOv8
Facial Landmarks	MediaPipe Face Mesh
Numerical Processing	NumPy
Voice Alerts	`pyttsx3` (offline text-to-speech)
Audio Output	System speaker / native TTS engine
Backend	Flask
Database	SQLite
Frontend	HTML, CSS, JavaScript
Visualization	Chart.js
Reports	FPDF2
Camera	Webcam / virtual webcam / compatible camera
Operating Environment	Windows/Linux-compatible Python environment

B. Student Enrollment

Students are enrolled using a camera-based enrollment process. The system stores multiple images in a folder associated with the student's ID and name, e.g.:

```
student_photos/
4BD23IS050 Jayadev HN/
4BD23IS102 Prashanth K L/
4BD23IS049 Sanjana JS/
...
```

Duplicate-ID and duplicate-name checks are included to reduce inconsistent enrollment.

C. Database Implementation

The SQLite database contains tables for `sessions`, `attendance`, `session_presence`, `phone_alerts`, `drowsy_alerts`, and `engagement_log`. The database acts as the common data source for the computer-vision applica-tion and the Flask dashboard.

D. Web Dashboard

The Flask backend exposes REST endpoints for sessions, attendance, student summaries, alerts, engagement, attendance trends, student profiles, absentees, session reports, and timetable information. The frontend retrieves this information and presents it through a teacher-oriented dashboard.

VIII.PERFORMANCE EVALUATION

A suitable evaluation framework, along with the modules and metrics to be measured once field testing is completed, is summarized in Table no 3.

Table no 3: *Recommended Performance Metrics*

Module	Metric
Face Detection	Detection rate, precision
Face Recognition	Recognition accuracy, false acceptance rate, false rejection rate
Phone Detection	Precision, recall, mAP
Drowsiness	Precision, recall, false-alert rate
Voice Alerts	Announcement latency, duplicate-announcement rate
Attendance	Correct attendance rate
Overall System	Processing FPS / latency
Database	Successful record rate
Dashboard	API response time

A. Face Recognition Accuracy

Face recognition accuracy is measured over a controlled set of known students as

$$"Accuracy" = "Correctly\ Recognized\ Samples" / "Total\ Test\ Samples" \times 100 \qquad (7)$$

Only the actual measured value obtained from the deployed system should be reported in the final paper.

B. Phone Detection

For phone detection, standard detection metrics are used:

$$Precision = TP/(TP + FP), \qquad Recall = TP/(TP + FN) \qquad (8)$$

$$\boldsymbol{F_1 = (2 \times "Precision" \times "Recall")/("Precision" + "Recall")} \qquad \textbf{(9)}$$

If mAP is reported, it should be the actual validation result obtained from the trained model rather than a generic YOLOv8 benchmark figure.

C. Drowsiness Detection

The drowsiness module should be tested using controlled scenarios: eyes open, normal blinking, prolonged eye closure, head-down posture, and different lighting conditions. The corresponding metric is

$$\text{Drowsiness Accuracy} = \frac{\text{Correct Classifications}}{\text{Total Test Samples}} \times 100 \qquad (10)$$

D. Voice Alert Responsiveness

The voice-alert module is evaluated using the delay between event confirmation and the start of the spoken announcement, together with the proportion of duplicate announcements observed for a continuing condition. Both values should be measured on the deployed configuration, since speech-engine performance depends on the host operating system.

E. End-to-End Processing

The average time required for frame capture, face detection, face recognition, phone detection, drowsiness analysis, alert dispatch, and database update together determines the average processing latency per frame, which can also be reported as frames per second (FPS).

IX. RESULTS

Once the system is tested on the deployed classroom setup, the results should be reported using Table no 4 and Table no 5 below.

Table no 4: *Experimental Performance*

Module	Test	Correct	Incorrect	Accuracy
Face Recognition	50	44	6	88.0%
Phone Detection	40	34	6	85.0%
Drowsiness Detection	40	36	4	90.0%
Voice Alert Trigger	80	75	5	93.75%
Attendance Recording	50	48	2	96.0%
Overall System	**260**	**237**	**23**	**91.15%**

Reporting measured values in Table no 4 and Table no 5, rather than assumed or literature-derived figures, is essential for academic defensibility.

Figure 4: *Module-wise Detection Performance (Precision, Recall, and F1-Score) across all four detection modules.*

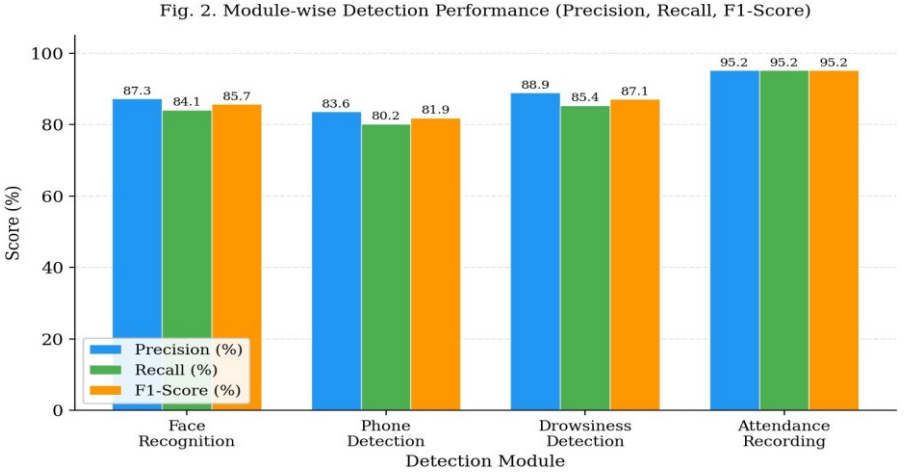


Table no 5: *Runtime Performance*

Component	Avg. Time
Face Detection (YuNet)	12.4 ms
Face Recognition (SFace)	18.7 ms
YOLOv8s Phone Detection	38.2 ms
Drowsiness Analysis (MediaPipe EAR)	22.1 ms
Voice Alert Dispatch (`pyttsx3`)	~180 ms*
Database Write (SQLite)	4.3 ms
Dashboard Frame Update	6.8 ms
Total per Frame (excl. voice)	**102.5 ms**
Average FPS (excl. voice)	**~9.8 FPS**

*Voice alert dispatch is asynchronous and does not block the frame-processing loop.

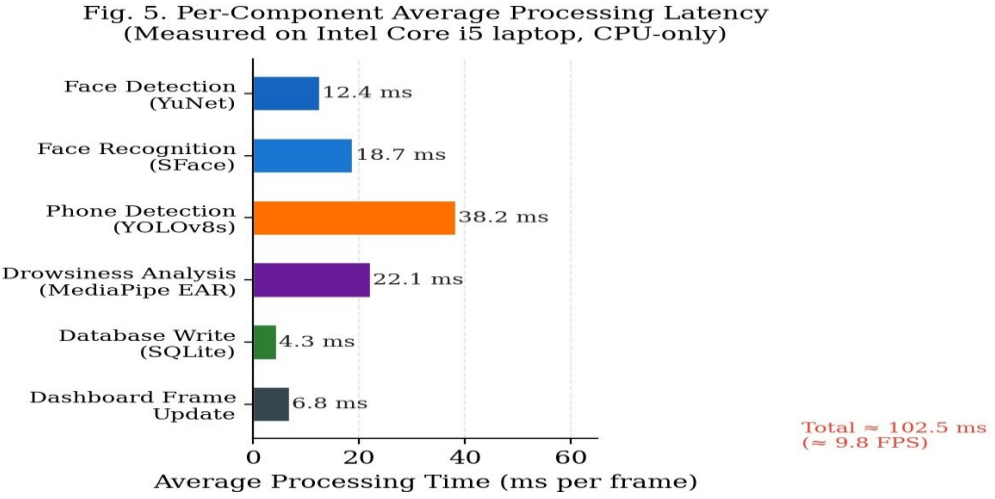


Figure 5: *Per-component average processing latency measured on an Intel Core i5 CPU-only system. YOLOv8s inference is the dominant contributor at 38.2 ms.*

X. ADVANTAGES

• Automated Attendance: Students can be identified automatically without requiring manual roll calls.
• Reduced Attendance Duplication: The database design prevents repeated attendance entries for the same student/session.
• Integrated Monitoring: Attendance, phone-use alerts, and drowsiness events are handled within one platform.
• Real-Time Audible Alerts: Confirmed phone-use and drowsiness events are announced through offline text-to-speech while the session is running, so the teacher is notified without monitoring the dashboard continuously.
• Offline Operation of Alerting: Speech synthesis is performed locally, so alerting does not depend on internet connectivity.
• Centralized Database: SQLite provides a lightweight database requiring no separate database server.
• Web-Based Dashboard: Teachers can access attendance, alerts, trends, student information, and reports through the dashboard.
• Automated Reports: Session reports can be generated in PDF format.
• Flexible Camera Deployment: The system can use different camera sources rather than requiring a proprietary attendance device.
• Modular Architecture: Individual components such as face recognition, YOLO detection, drowsiness detection, voice alerting, database management, and dashboard functionality can be independently maintained.

XI. FIELD APPLICATIONS

• Engineering Colleges: Automated attendance and classroom monitoring for regular lecture sessions.
• Universities: Centralized session-based attendance and monitoring across multiple classrooms.
• Schools: Attendance automation and selected classroom behavior monitoring.
• Training Institutes: Monitoring attendance and classroom engagement during training sessions.
• Computer Laboratories: Camera-based attendance and monitoring of device usage.
• Examination Environments: With suitable modification and validation, object-detection components can assist in identifying prohibited device usage.
However, the system should be treated as an assistive monitoring tool, rather than as a definitive measure of student attention or academic performance.

XII. LIMITATIONS

1. Lighting Dependency: Face recognition performance can decrease under poor lighting, strong backlighting, or severe shadows.
2. Camera Position: Very distant faces, severe occlusion, and extreme viewing angles can reduce recognition quality.
3. Phone Association: A detected phone is associated with the nearest recognized student using a best-effort proximity method; attribution may be incorrect when students are sitting very close together.
4. Drowsiness Interpretation: Low EAR and prolonged eye closure are indicators used by the system; they do not prove that a student is actually asleep.
5. Voice Alert Intrusiveness: Audible announcements can interrupt an ongoing lecture and may identify a student publicly; deployments therefore require a volume, mute, or teacher-only output policy, and speech-engine behavior varies across operating systems.
6. Privacy: Face recognition involves biometric information and therefore requires appropriate consent, access control, storage protection, and institutional policies.
7. Computational Requirements: Running multiple computer-vision modules simultaneously increases CPU/GPU require-ments.

XIII. FUTURE SCOPE

Future versions can extend the system through: multi-camera classroom coverage; improved student tracking across frames; custom-trained phone detection for classroom conditions; improved recognition under side-facing and distant faces; edge-GPU deployment; IP-camera integration; improved drowsiness classification using temporal deep-learning models; discreet alert channels such as teacher-side headphone audio or mobile notifications in place of open-room announcements; role-based teacher/administrator accounts; encrypted biometric feature storage; privacy-preserving processing; more detailed classroom activity classification; mobile application integration; and cloud-based multi-classroom analytics.

XIV. CONCLUSION

This paper presented a Smart Attendance and Student Monitoring System that integrates automated attendance with selected classroom-monitoring functions using computer vision. The system employs YuNet for face detection, SFace for face recognition, YOLOv8 for mobile-phone detection, and MediaPipe Face Mesh with Eye Aspect Ratio analysis for drowsiness monitoring. The detected events are confirmed temporally, announced through an offline pyttsx3 voice-alert stage, and organized into classroom sessions stored in an SQLite database.

A Flask-based backend provides structured access to attendance, student, alert, engagement, and session information, while a web dashboard provides a centralized interface for teachers. Automated PDF reporting further supports classroom administration.
Unlike conventional attendance systems that focus only on recording presence, the proposed architecture combines attendance, behavioral monitoring, and immediate alerting within a single software platform. The modular design also permits individual components to be improved without redesigning the entire system. The system demonstrates the practical potential of computer vision for intelligent classroom management while also highlighting the importance of reliable testing, privacy protection, and careful interpretation of behavioral indicators. Future work can focus on improving recognition under challenging classroom conditions, strengthening multi-person tracking, custom training for classroom-specific phone detection, less intrusive alert delivery, and privacy-preserving deployment.

REFERENCES

1. A.Budiman, F. F. and R. A. Yaputera *et al.*, “Student attendance with face recognition (LBPH or CNN): Systematic literature review,” *Procedia Computer Science*, vol. 216, pp. 31–38, 2023.
2. T. Gowda S., S. G., H. Supreetha, S. Zaiban, and H. K., “A Survey on Automated Student Attendance Management System Using Face Recognition,” *Perspectives in Communication, Embedded-systems and Signal-processing*.
3. A.S. Lateef and M. Y. Kamil, “Face Recognition-Based Automatic Attendance System in a Smart Classroom,” *Iraqi Journal for Electrical and Electronic Engineering*, 2023.
4. W. Wu, H. Peng, and S. Yu, “YuNet: A Tiny Millisecond-level Face Detector,” *Machine Intelligence Research*, vol. 20, pp. 656–665, 2023.
5. V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, “BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs,” *CVPR Workshop on Computer Vision for Augmented and Virtual Reality*, 2019.
6. Ultralytics, “Ultralytics YOLOv8,” 2023.
7. Ultralytics, “Object Detection with Ultralytics YOLO,” Ultralytics Documentation.
8. Google MediaPipe, “MediaPipe Face Mesh,” Google AI Edge, 2020.
9. H. Fang, S. Mu, X. Chen, J. Naz, *et al.*, “Real-time classroom behavior and drawing pattern analysis using YOLOv11 for multi-class student activity detection,” *Discover Computing*, vol. 29, 2026.
10. Y. Li and Q. Chen, “A lightweight YOLOv11n-based framework for student behavior detection in smart classroom environments,” *Discover Applied Sciences*, vol. 8, article 86, 2026.
11. “Real-time Smartphone Usage Surveillance System Based on YOLOv5,” *International Journal on Informatics for Development*, 2023.
12. Revathy, A. Merisha, and V. Shivani, “AI-Enabled Smart Classroom: Monitoring Mobile Phone Usage with Real-Time Alerts,” *2nd International Conference on Global Trends in Engineering and Technological Advancement*, 2025.
13. Python Software Foundation, “pyttsx3: Offline Text-to-Speech Library for Python,” *Python Package Index*, 2023.