

Attention Educosmos: A Smart Classroom Focus and Smartphone Monitoring System

Veeresha S M^{*1}, Sharath Ramesh Shiragaon², Siddesh G S³, Mallikarjuna T M⁴,
Parveen Taj S⁵

^{*1,2,3,4}UG Scholars, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India.

⁵Professor, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davanagere, Karnataka, India.

^{*}Corresponding Author.

To Cite this Article: Veeresha S M^{*}, Sharath Ramesh Shiragaon², Siddesh G S³, Mallikarjuna T M⁴, Parveen Taj S⁵, "Attention Educosmos: A Smart Classroom Focus and Smartphone Monitoring System", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 455-459.



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: Manual roll-calls consume teaching time and are easy to falsify, while a single teacher cannot watch a whole room for smartphone use. Existing face-recognition attendance systems stop at attendance and take no account of classroom behaviour. This paper presents Attention Educosmos, a CPU-only system that identifies students from a classroom video stream, marks attendance automatically, detects mobile phones, attributes each phone to a specific student, and responds with a graduated policy: two private warning e-mails to the student, followed by escalation to the teacher and parent only if use continues. Monitoring is bound to a configurable timetable and is switched off during breaks. The pipeline combines the YuNet face detector, dlib 128-dimensional face embeddings and YOLOv8n. We describe four design decisions that were needed to make these off-the-shelf models behave reliably together: track-level identity persistence, a zoomed per-student region-of-interest pass for small phones, a face-size-scaled ownership rule, and a timer-driven escalation state machine. We report the parameters and the measurements behind them from a single-classroom recording, and we state plainly that a formal precision/recall evaluation on a larger dataset remains future work.

Key Word: Smart classroom; Face recognition; YOLOv8; Smartphone detection; Automated attendance; Student monitoring; Computer vision

I. INTRODUCTION

Two routine classroom problems motivate this work. Roll-call can consume several minutes of every period and is easy to defeat by proxy answers. Phone use during lectures is also hard to police: a teacher facing forty or more students cannot lecture and observe every row at once, and the usual remedies are poor. Signal jammers are unlawful in India and block emergency calls, confiscation is confrontational and interrupts teaching, and doing nothing lets the distraction continue.

In the literature we reviewed, most face-recognition attendance systems solve only the first problem, and most classroom-behaviour detectors report an aggregate behaviour class rather than acting on an identified individual [6]–[8]. What is missing is one system that links a detected behaviour to a named student and responds proportionately.

Contributions.

1. An integrated pipeline joining face-based attendance and phone detection on a shared recognition backbone;
2. a graduated, timetable-aware escalation policy that never contacts a third party before the student has been warned privately;
3. four documented engineering measures that make lightweight CPU models usable in a real classroom scene; and (4) an open account of the limitations of the current evaluation.

II. RELATED WORK

Face detection and recognition. Cascade detectors [1] are fast but weak on non-frontal and small faces, which dominate a wide classroom shot. YuNet [2] is a compact CNN detector distributed with OpenCV that handles such faces at millisecond cost. Embedding-based recognition [3] reduces identification to nearest-neighbour matching, so a new student is enrolled without retraining; the dlib toolkit [4] provides a widely used 128-dimensional implementation.

Object detection. The YOLO family [5] performs single-pass real-time detection, and the Ultralytics YOLOv8 release [10] ships COCO-pretrained weights that include the *cell phone* class [11]. The nano variant is the smallest and weakest on small objects, a limitation addressed in Section IV-C.

Classroom behaviour analysis. Chen et al. [6] improve YOLOv8 for student-behaviour detection, Liu et al. [7] propose YOLOv8n-BT for learning-behaviour recognition, Yang and Wang [8] release the SCB-Dataset3 benchmark, and Indira et al. [9] describe a mobile-usage detection and notification system for restricted areas. These works detect behaviour; our work additionally identifies who is responsible, applies a staged response, and gates monitoring by the class timetable.

III. SYSTEM ARCHITECTURE

The system uses four layers (Fig. 1). A single-page web dashboard (HTML, CSS, JavaScript) shows the live annotated stream and six functional views. A Flask server exposes about 23 REST endpoints and an MJPEG stream. The vision layer runs concurrently on separate threads for capture, recognition and phone detection, so the two expensive operations never block the live video. State is stored in SQLite across five tables (students, timetable, attendance, warnings, alert history), and warnings are delivered by SMTP.

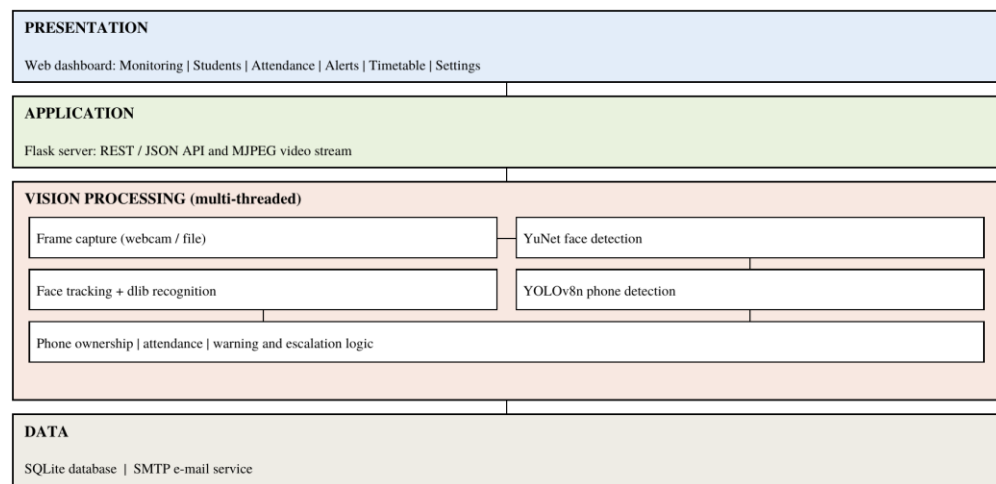


Fig. 1. Layered, multi-threaded architecture of Attention Educosmos.

The video source is either a webcam or a recorded file placed in a watched folder; the watcher accepts a file only after its size is stable across two scans, normalises it to a cached proxy and loops it. Every frame is downscaled to 960 px width before any processing; at this size JPEG encoding costs about 3 ms against about 38 ms for a 4K frame, which is the difference between smooth and stuttering playback on a CPU. The live monitoring view of the dashboard is shown in Fig. 2.

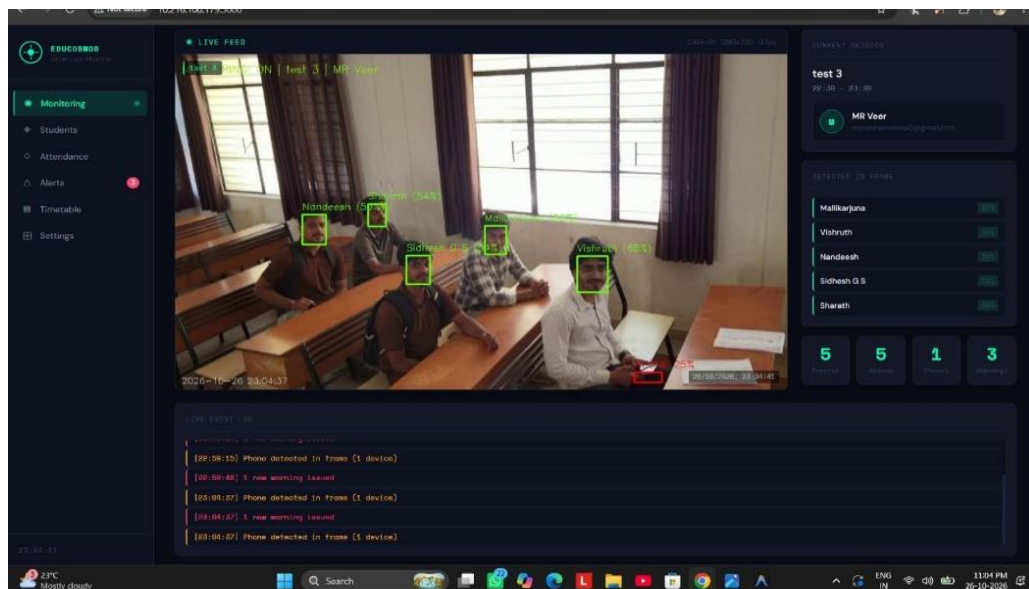


Fig. 2. Live monitoring view of the dashboard: annotated video feed with recognised students, and Present / Phones / Warnings counters. Faces are pixelated for privacy.

IV. METHODOLOGY

A. Timetable gating

Each timetable row stores subject, start and end time, teacher, teacher e-mail, class, a break flag and a day-of-week field that accepts ALL, a single day or a list. At each cycle the current time and weekday are matched to a row; monitoring runs only if a non-break session is active. This keeps the system silent outside teaching time and also supplies the correct teacher for any later escalation.

B. Face detection, tracking and recognition

YuNet runs every 0.12 s (score threshold 0.6, NMS 0.3). Recognition is slower and runs on its own thread every 0.40 s: dlib encodings are compared with the enrolled matrix, and a match is accepted at a confidence of at least 0.50. The key decision is that *identity belongs to the track, not to the frame*. A frame where recognition fails (the student looks down) carries no evidence about who the person is and must not erase what is known. Each track therefore coasts up to 24 detection cycles, needs three consistent contrary matches before it is relabelled, and is associated first by IoU (0.25) and then by a distance-and-size fallback.

Two further rules suppress spurious boxes. An unnamed track is drawn only after four sightings and coasts for only two cycles, which removes one-frame false faces on clothing or chair backs and stale “Unknown” boxes beside correctly named students.

C. Two-pass phone detection

A phone in a wide classroom frame can be only a few tens of pixels across. The detector therefore runs two passes per cycle: a whole-frame pass at 640 px (confidence 0.20) and a pass over crops around each tracked student (confidence 0.15, at most eight crops). Each crop is enlarged to a target zoom of 2.75 times, with the inference size clamped between 320 and 640 px to bound cost. The lower threshold is safe in the crop pass because clutter elsewhere in the room cannot enter a single student’s crop.

D. Phone-to-student ownership

A detected phone is assigned to the nearest eligible identified student, using a reach region measured in multiples of that student’s own face size: 3.0 face widths each side, 0.5 face heights above and 3.4 below the chin. Scaling by face size makes the rule scale-invariant, so a student at the back has a proportionally smaller region. To stop ownership flickering between neighbours, a rival must be 1.35 times nearer to take a phone, and a phone box is held for 3.5 s after it was last seen.

E. Graduated escalation

When a phone is first attributed to a student, Warning 1 is e-mailed immediately and the student is registered as a pending violation. If the violation is still active 30 s later, Warning 2 is sent; 30 s after that the teacher of the current session and the student’s parent are notified, and the pending record is cleared. The sequence runs on a timer from the stored violation context, so the phone does not need to stay visible for the whole minute, and the context is refreshed if the phone reappears. Every message is logged with recipient, role, time and delivery outcome.

F. Attendance

On a confident recognition during an active session the student is marked present once per student, session and date, enforced by a database uniqueness constraint.

V. IMPLEMENTATION

The system runs on a single CPU-only machine (minimum 4-core processor and 4 GB RAM) and needs no GPU or cloud service. Table 1 lists the stack. If no camera or ML library is present the dashboard falls back to a demo mode so that management and reporting remain usable.

Component	Technology
Language / server	Python 3.10+, Flask, flask-cors
Face detection	OpenCV YuNet (ONNX)
Face recognition	dlib, face_recognition (128-d embeddings)
Phone detection	Ultralytics YOLOv8n (COCO class 67)
Storage / mail	SQLite; SMTP (TLS)
Front end	HTML5, CSS3, JavaScript

Table 1: Software stack

The dashboard provides student management with face-image enrolment, timetable management, live monitoring with present and absent counts, attendance records grouped by subject, warning and alert history, and SMTP configuration with a test-mail function.

VI. RESULTS AND DISCUSSION

The prototype was run on a classroom recording of a small group of enrolled students. Table 2 summarises the deployment, and Table 3 the diagnostic measurements that fixed the parameters in Section IV. These are design measurements from one recording, not accuracy results.

Findings. *First*, small-object detection on CPU is a framing problem rather than a model limit: matching a crop’s size to a 320 px input gave only a 1.37x enlargement and a phone about 26 px wide, whereas targeting a multiple of the crop gave a phone roughly 43 px wide. *Second*, most visible failures (identity flicker, ghost boxes, forked tracks, false faces) were

removed by temporal reasoning rather than a better detector: the false faces scored 0.61–0.62 against 0.90 for real faces yet were wider than the median real face, so neither a score nor a size filter separated them, while persistence did. *Third*, the alert policy had to be checked against real detection behaviour. The original three warnings at 60 s placed the teacher three minutes of unbroken attribution away, which was never reached, and the alert history held only isolated first warnings; an escalation branch sharing the warning cooldown was also starved. Separating the two timers and using two warnings at 30 s made the full sequence reachable.

Item	Value
Class / students registered	CSE-A / 10
Students with face enrolment	7
Timetable sessions	5
Attendance rows logged	53
Warnings / alert-history rows	10 / 20

Table 2: Deployed configuration

Observation (single recording)	Value
JPEG encode, 4K vs 960 px frame	~38 ms vs ~3 ms
Real-face detection persistence	97–100% of passes
False faces (collar, chair back)	3 in 1041 detections
Their persistence	2 sightings in 28 s
Track fork: IoU vs own stale box	0.156 (< 0.25 gate)
Required crop reach, sideways (max / median)	2.71 / 1.73 face widths
Required crop reach, below chin (max / median)	3.03 / 2.29 face heights
Phone outside crop at old reach	5 of 65 frames
Effective zoom, size-matched vs targeted	1.37x vs 2.75x

Table 3: Diagnostic measurements behind the parameters

Behaviour on unenrolled faces. When the recording contains people who are not enrolled, the system correctly labels them “Unknown” and neither marks attendance nor issues warnings for them, since the ownership rule of Section IV-D considers identified students. Recognition of many faces per cycle is also the most expensive step on a CPU (about 0.5 s per face in our profiling), so recognition effort should be limited to new or unidentified tracks when many faces are in view.

VII. LIMITATIONS, ETHICAL CONSIDERATIONS AND FUTURE WORK

Evaluation. The study uses 10 registered students (7 enrolled) and one recording. No precision, recall or confusion matrix is reported, so no accuracy claim is made. A ground-truth annotated, multi-session dataset is the first priority.

Scope and security. The system serves one class and one video source and has no authentication, so anyone with dashboard access has full administrative control.

Sensing. A phone held wholly below a desk is invisible to a front camera, and enrolment from one frontal photograph limits robustness to pose and lighting.

Privacy and ethics. The system processes facial images of students and sends notifications to parents and teachers, so it carries privacy and fairness risks. In this work, faces in all published figures are pixelated, recognition data are stored locally in the deployment database, and no data are sent to a cloud service. Any real deployment would require the informed consent of students and, for minors, their guardians; a stated retention period for stored images and encodings; role-based access control; and a human review of every warning before escalation, because false detections could wrongly accuse a student.

Future work. Role-based access, a branch/year/section/classroom hierarchy with several cameras, multi-angle enrolment, GPU inference with a larger detector, behaviour cues beyond phones such as head pose, analytics for repeat patterns, and privacy hardening such as face blurring in stored frames.

VIII. CONCLUSION

Attention Educosmos shows that attendance and smartphone monitoring can share one recognition pipeline on ordinary CPU hardware and respond proportionately, warning a student privately before involving a teacher or parent. Its practical value lies less in the choice of models than in the measured tuning around them: track-level identity, a zoomed per-student detection pass, a face-size-scaled ownership rule and a reachable escalation timer. A larger formal evaluation and access control are the necessary next steps before wider use.

Acknowledgement

The authors thank Prof. Parveen Taj S, Department of Information Science and Engineering, BIET Davanagere, for guidance throughout this work.

REFERENCES

1. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE CVPR*, 2001, pp. 511–518.
2. W. Wu, H. Peng, and S. Yu, "YuNet: A tiny millisecond-level face detector," *Mach. Intell. Res.*, vol. 20, no. 5, pp. 656–665, 2023.
3. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proc. IEEE CVPR*, 2015, pp. 815–823.
4. D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. 10, pp. 1755–1758, 2009.
5. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR*, 2016, pp. 779–788.
6. Chen, G. Zhou, and H. Jiang, "Student behavior detection in the classroom based on improved YOLOv8," *Sensors*, vol. 23, no. 20, p. 8385, 2023.
7. Q. Liu, R. Jiang, Q. Xu, *et al.*, "YOLOv8n_BT: Research on classroom learning behavior recognition algorithm based on improved YOLOv8n," *IEEE Access*, vol. 12, pp. 36391–36403, 2024.
8. Yang and T. Wang, "SCB-Dataset3: A benchmark for detecting student classroom behavior," arXiv:2310.02522, 2023.
9. D. Indira, A. Yaseen, C. S. Venkat, D. Supriya, and C. Udayini, "Student mobile usage detection and fine notification system in restricted areas using deep learning techniques," in *Proc. ICoICI*, IEEE, 2024, pp. 1413–1420.
10. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. [Online]. Available: <https://github.com/ultralytics/ultralytics>
11. T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. ECCV*, 2014, pp. 740–755.