DRUSHTI-AI: A Smart Surveillance System

Pushkar Mahajan[1], Arjun Deshmukh[2], Sumit Suryawanshi[3], Yash Ingale[4], Prof. Tanmayee Kute[5]
[1,2,3,4]*Department of Artificial Intelligence & Machine Learning, Progressive Education Society's Modern College of Engineering, Shivajinagar, Pune – 411005, Maharashtra, India*
[5]*Guide, Progressive Education Society's Modern College of Engineering, Shivajinagar, Pune – 411005, Maharashtra, India*

***Abstract*—Academic examination environments continue to depend on human invigilation as the primary mechanism for malpractice prevention, a method constrained by operator fatigue, divided attention across large candidate populations, and inconsistent enforcement. This paper presents Drushti AI, an intelligent surveillance system that autonomously detects cheating behaviours in examination halls through continuous analysis of live video feeds. The system integrates YOLO11 Nano for real-time person detection, DeepSort for persistent multi-candidate identity tracking, and MediaPipe FaceMesh for granular facial landmark extraction. Behavioural violations are inferred via OpenCV-based head pose estimation computing yaw, pitch, and roll angles using the solvePnP algorithm alongside iris-based gaze deviation analysis and lip movement monitoring. Five malpractice categories are identified: lateral head deviation, sustained gaze redirection, lip movement, inter-candidate proximity, and prolonged face absence. A FastAPI backend orchestrates the detection pipeline with threaded multi-camera support and streams incident data to clients via WebSocket, persisting all detected events in a Supabase database. Operators interact through a Next.js web dashboard and a companion mobile application. Experimental evaluation on webcam- and CCTV-based video demonstrates reliable detection across all five categories with low false-positive rates at near-real-time processing speeds.**

***Index Terms*—Intelligent Surveillance, Examination Hall Monitoring, YOLO11, DeepSort, MediaPipe FaceMesh, Head Pose Estimation, Gaze Deviation Detection, Malpractice Detection, FastAPI, WebSocket, Supabase, Next.js**

I. INTRODUCTION

The reliability of academic examinations depends fundamentally on the effectiveness of the invigilation process. In practice, human invigilators managing halls with tens or hundreds of candidates simultaneously cannot maintain consistent vigilance over every individual throughout an examination session. Fatigue accumulates over extended periods, attention is inherently divided, and subtle physical cues through which malpractice most commonly manifests persistent sideways glances, repeated gaze deviation, quiet lip movement, or sustained proximity between neighbouring candidates are easily overlooked. As a result, traditional invigilation is structurally incapable of providing the uniform and continuous monitoring required for examination integrity [1].

Academic malpractice carries consequences that extend beyond a single examination event. Compromised assessment outcomes undermine institutional credibility, distort competitive processes such as admissions and employment selection, and erode the value of qualifications for all candidates. Despite these implications, the technological infrastructure deployed in most examination environments has not evolved significantly beyond passive closed-circuit recording systems solutions that document incidents retrospectively rather than detecting and responding in real time [2]. The absence of automated behavioural monitoring leaves institutions dependent on human operators whose capacity for sustained attention is inherently limited.

To address these limitations, this paper presents Drushti AI, an intelligent surveillance system designed for real-time malpractice detection in examination halls. The proposed system employs a multi-model computer vision pipeline integrating YOLO11 Nano for person detection, DeepSort for persistent candidate tracking across frames, and MediaPipe FaceMesh for facial landmark analysis, enabling advanced behavioural inference including head pose estimation, gaze deviation analysis, and lip movement detection. Unlike conventional surveillance approaches that analyse frames independently, Drushti AI maintains a continuous behavioural state for each tracked candidate. Alerts are triggered only when suspicious behaviour persists beyond configurable temporal thresholds, significantly reducing false positives while ensuring reliable detection of genuine malpractice [3].

II. LITERATURE SURVEY

A. Deep Learning-Based Video Surveillance

Solorzano and Rengifo [1] presented a deep learning-based surveillance system combining convolutional neural networks with a YOLOv8 backbone, trained on 2,782 images capturing normal and simulated shoplifting scenarios. The pipeline achieved 82% precision, 80% recall, and 84% mAP50. While demonstrating the viability of automated surveillance, the system lacks persistent identity tracking and facial behavioural analysis gaps central to the examination monitoring problem.

B. Real-Time Object Detection for Surveillance

Saldanha et al. [2] proposed a YOLO-based surveillance system with daytime weapon identification and nighttime unauthorised person detection modules. Operating without persistent tracking, each frame was analysed independently. This frame-level approach is insufficient for examination monitoring, where violations such as sustained head deviation require temporal observation across multiple frames a distinction that Drushti AI handles through its per-track state machine with configurable duration thresholds.

C. YOLO-Based Theft Detection

Gaikwad et al. [3] investigated multiple YOLO architecture generations for automated theft detection, validating YOLO as a strong candidate for real-time surveillance. However, the system operated on individual frame detections without temporal continuity or identity preservation precisely the limitation that Drushti AI addresses through its combination of DeepSort tracking and time-gated behavioural state management.

III. METHODOLOGY

Drushti AI performs real-time intelligent surveillance in examination environments by integrating computer vision, object tracking, and behavioural analysis. The system processes live video streams to detect suspicious activities based on temporal and spatial patterns. The methodology is structured into four components: system design, architecture, detection pipeline, and behavioural analysis.

A. System Design

The overall system follows a modular and scalable design enabling continuous monitoring and real-time alert generation. The workflow begins with video acquisition from webcams or CCTV feeds. Captured frames are processed through a sequence of detection and analysis stages to identify potential cheating behaviours:

- Video input is captured and frames are preprocessed for efficient inference.
- Person detection is performed using YOLO11 Nano, generating bounding boxes per frame.
- Detected individuals are assigned unique identities via DeepSort for persistent tracking.
- Facial regions are analysed using MediaPipe FaceMesh to extract 468 landmarks per candidate.
- Behavioural patterns gaze, head orientation, lip activity are evaluated against configurable thresholds.
- Temporal consistency checks confirm suspicious activity before alert generation.
- Alerts are transmitted via WebSocket; snapshots are stored in Supabase.

B. System Architecture

The architecture of Drushti AI supports real-time processing, scalability, and seamless inter-component communication. It is divided into three primary layers, as illustrated in Fig. 1.

1) Input Layer: Captures live video streams from webcams or CCTV cameras. Supports multiple concurrent camera inputs for scalability across large examination environments.

2) Processing Layer: The YOLO11 Nano model performs real-time person detection; DeepSort maintains persistent multi-object identity tracking; MediaPipe FaceMesh extracts facial landmarks; and a dedicated behavioural analysis module infers malpractice patterns from extracted features.

3) Application Layer: A FastAPI backend orchestrates the detection pipeline and streams results via WebSocket. Supabase stores incident logs and

snapshot evidence. A Next.js web dashboard enables live feed monitoring and incident review. A companion mobile application allows invigilators to register session metadata and review incidents remotely.

C. Detection Pipeline

Each video frame undergoes a structured sequence of processing steps that transform raw pixel data into actionable behavioural insights.

1) Frame Acquisition and Preprocessing: Frames are captured from the video stream. Resizing and normalisation are applied for optimal model performance.

2) Person Detection: The YOLO11 Nano model detects individuals in each frame, generating bounding boxes and confidence scores. The nano variant balances detection accuracy with computational efficiency.

3) Object Tracking: DeepSort assigns unique persistent IDs to each detected individual and maintains identity continuity across frames, essential for temporal behavioural analysis.

4) Face and Landmark Detection: Face regions are extracted from detected bounding boxes. MediaPipe FaceMesh identifies 468 facial landmarks per candidate, providing the basis for head pose and gaze analysis.

5) Feature Extraction: Head pose comprising yaw, pitch, and roll angles is estimated using the solvePnP algorithm applied to a generic 3D facial model. Eye gaze deviation is computed from iris landmark positions. Lip movement is analysed through inter-landmark distance variation across consecutive frames.

6) Behaviour Evaluation and Alert Generation: Each detected feature is compared against predefined thresholds. Temporal consistency is checked using duration-based rules per tracked individual. If suspicious behaviour persists beyond the threshold duration, an alert is triggered, a snapshot is logged to Supabase, and the alert is pushed via WebSocket to connected interfaces.

D. Behavioural Analysis and Decision Logic

Unlike traditional frame-based systems, Drushti AI incorporates temporal behaviour modelling to improve accuracy and reduce false positives. Each tracked individual is associated with an evolving state. Five malpractice categories are detected:

- Looking Sideways: Triggered when head yaw exceeds a defined angular threshold for a sustained duration.
- Gaze Deviation: Detected when eye movement deviates significantly from the forward direction using iris landmark positions.
- Talking Detection: Identified through continuous lip movement change exceeding a threshold over a time interval.
- Left Seat Detection: Triggered when the face is absent for a specified duration, indicating seat abandonment.
- Proximity Detection: Activated when two candidates remain within a defined pixel distance for a prolonged period, indicating potential collusion.

Each behaviour is validated using time-based thresholds, minimising false alarms caused by brief or natural movements.

E. Real-Time Communication and Alert System

WebSocket connections maintained by the FastAPI backend stream detection results simultaneously to the Next.js web dashboard and the mobile application. Snapshots of detected incidents are stored in Supabase for subsequent audit. The mobile application allows professors to register session metadata including classroom number and total student count enabling structured post-examination review.

IV. RESULTS AND DISCUSSION

The performance of Drushti AI was evaluated using real-time video streams from webcam- and CCTV-based classroom environments. Evaluation focused on accurate detection and classification of malpractice behaviours, consistent tracking across multiple candidates, and near-real-time processing speeds.

A. Detection Performance

The system successfully detected and classified all five malpractice categories. YOLO11 Nano ensured efficient person detection in moderately crowded environments. DeepSort enabled consistent identity tracking, associating behavioural patterns with specific individuals across frames. MediaPipe FaceMesh provided reliable head orientation and gaze direction estimation under normal lighting, with minor sensitivity to extreme angles or partial occlusions.

B. Temporal Behaviour Analysis
Short-duration gaze shifts were ignored unless sustained beyond the configured threshold; proximity alerts were triggered only when candidates remained within a defined distance for a continuous interval; and lip movement detection required consistent activity over multiple frames. This time-based filtering substantially improved alert reliability and reduced false positives caused by natural or incidental actions.

C. Real-Time Processing Performance
The system operates at near-real-time speeds with performance influenced by hardware configuration, frame resolution, and number of active camera streams. Frame skipping and the selection of YOLO11 Nano maintained a balance between speed and accuracy. The FastAPI backend with WebSocket communication enabled low-latency transmission of detection results to the frontend dashboard.

D. User Interface and Monitoring Effectiveness
The Next.js dashboard provided an intuitive interface featuring live multi-camera feed display, real-time alert notifications, a paginated filterable incident log, and snapshot visualisation. The companion mobile application allowed invigilators to register examination sessions, view incidents remotely, and access snapshot evidence for verification. This dual-interface design enabled both centralised and remote monitoring.

E. Limitations and Observations
Detection accuracy may decrease under poor lighting or heavy occlusion. Extreme head poses or partially visible faces can affect landmark detection. High-density environments may introduce tracking inconsistencies. Performance depends on hardware capabilities, particularly for multi-camera setups.

V. CHALLENGES

A. False Positives and Behavioural Ambiguity
Movements such as briefly looking sideways or adjusting posture may resemble cheating behaviour. Ensuring a balance between sensitivity and accuracy through appropriate threshold calibration remains critical to deployment effectiveness.
B. Environmental Variability
Performance depends heavily on environmental factors including lighting, camera angles, and resolution. Poor lighting, shadows, or occlusions can affect face detection and landmark extraction accuracy in crowded examination halls.
C. Computational Resource Dependency
Real-time video processing requires significant computational resources, particularly for multiple concurrent camera streams. Systems without GPU acceleration may experience latency or reduced frame rates. Camera quality directly impacts detection accuracy.
D. Tracking Consistency
Occlusion, rapid movement, or overlapping individuals may cause identity switching or temporary tracking loss, affecting the continuity of behavioural analysis. The system relies on consistent tracking IDs to evaluate temporal patterns.
E. Privacy and Ethical Considerations
Continuous monitoring raises concerns related to privacy and data security. Capturing and storing video feeds, facial data, and incident snapshots requires appropriate safeguards, controlled access mechanisms, and compliance with applicable data protection regulations.
F. Scalability and Network Dependency
Deploying across multiple examination halls requires scalable architecture and stable network connectivity. Real-time alert transmission via WebSocket depends on reliable infrastructure; network interruptions may delay alerts or disrupt monitoring.
G. Parameter Tuning and Generalisation
Configurable thresholds for gaze deviation, head movement, and proximity may vary across environments, requiring calibration for optimal performance. Designing a system that generalises across diverse settings without frequent recalibration remains an open challenge.

VI. CONCLUSION

This paper presented Drushti AI, an intelligent surveillance system addressing the structural limitations of human-dependent examination invigilation. By integrating YOLO11 Nano for person detection, DeepSort for persistent identity tracking, and MediaPipe FaceMesh for facial landmark-based behavioural inference, the system autonomously monitors examination hall video feeds and detects five malpractice categories lateral head deviation, gaze redirection, lip movement, inter-candidate proximity, and seat abandonment without requiring continuous human observation.

A key distinguishing contribution is its temporal behaviour modelling approach: Drushti AI evaluates each tracked candidate's behavioural state across time and issues alerts only when a suspicious pattern is sustained beyond a configurable threshold. This substantially reduces false positives while preserving sensitivity to genuine violations, addressing a fundamental weakness in existing frame-level surveillance approaches.

The system architecture comprising a FastAPI detection backend, Supabase incident database, Next.js web dashboard, and companion mobile application provides a complete end-to-end pipeline from video ingestion to operator-facing alert delivery. Experimental evaluation confirms reliable detection across all five behaviour categories under standard classroom conditions.

Planned extensions include model fine-tuning on domain-specific examination datasets, integration with edge computing hardware, expansion to additional behavioural indicators such as material concealment, and automated session reporting. Drushti AI demonstrates that scalable, accurate, and deployable intelligent surveillance for examination environments is achievable with current computer vision technology, establishing a foundation for further research in automated academic integrity monitoring.

REFERENCES

[1] Aguero, Y. A. S., and Rengifo, J. K. C. (2025). Intelligent surveillance system using deep learning to reduce shoplifting in minimarkets. In Proceedings of the 22nd International Conference on Informatics in Control, Automation and Robotics (ICINCO 2025), Vol. 2, pp. 342–349. https://doi.org/10.5220/0013750800003982

[2] Saldanha, V., Naik, K. D., Parashtekar, S. S., Sirsat, V. V., and J, V. V. (2025). Intelligent surveillance and theft detection using real-time object detection. *Research Proposal*, March 2025. https://doi.org/10.13140/RG.2.2.22664.53765

[3] Gaikwad, N., Raut, A., Tandekar, E., Nighot, E., and Narnavrek, P. Machine learning based theft detection using YOLO object detection. *Journal of Technology*, 14(5), 16–20.

[4] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. https://doi.org/10.1109/CVPR.2016.91

[5] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint*, arXiv:2004.10934.

[6] Wojke, N., Bewley, A., and Paulus, D. (2017). Simple online and realtime tracking with a deep association metric. In Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 3645–3649.

[7] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. In Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 3464–3468.

[8] Kartynnik, V., Ablavatski, A., Grishchenko, I., and Grundmann, M. (2019). Real-time facial surface geometry from monocular video on mobile GPUs. arXiv preprint, arXiv:1907.06724.

[9] Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). EPnP: An accurate O(n) solution to the PnP problem. International Journal of Computer Vision, 81(2), 155–166.

[10] Ramírez, S. (2018). FastAPI: Modern, fast web framework for building APIs with Python. Available: https://fastapi.tiangolo.com

[11] Ultralytics. (2024). *YOLO11*: Real-time object detection. Available: https://docs.ultralytics.com

APPENDIX

System Architecture Diagram

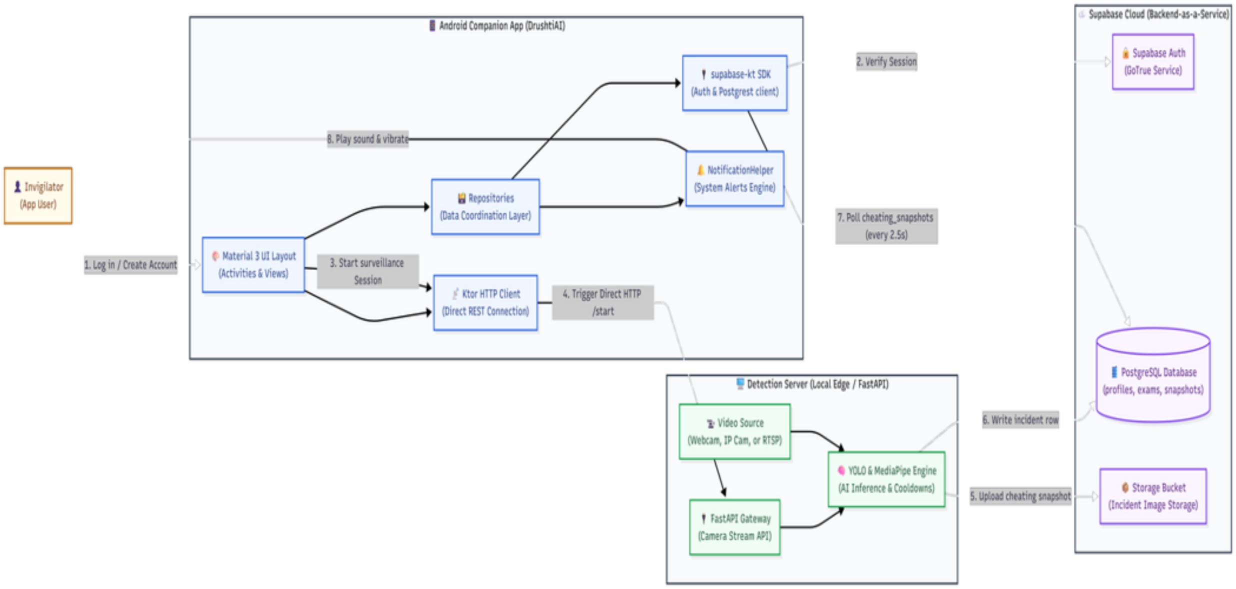

Fig. 1. System Architecture of Drushti AI showing the Android Companion App (DrushtiAI), Detection Server (Local Edge / FastAPI), and Supabase Cloud Backend-as-a-Service with their numbered interaction steps.