iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 Issue: IV Month of publication: April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80684

Agentic AI Smart Exam Surveillance and Physical Alert System

Shravan Wargantiwar[1], Anushree Khadgi[2], Vanshika Khedkar[3], Dr. Reema C. Roychaudhary[4]
[1]Student, [2]Student, [3]Student, [4]Assistant Professor, Dept. of Computer Engineering, St. Vincent Pallotti College of Engineering & Technology, Nagpur, Maharashtra, India

Abstract: ***Academic examination integrity is a critical concern in modern educational institutions. Conventional proctoring methods are labour- intensive, inconsistent, and fail to monitor all students simultaneously in large examination halls. This paper presents an Agentic AI Smart Exam Surveillance and Physical Alert System—a real-time, multi-agent framework that automates the detection of cheating behaviours using computer vision and IoT hardware integration. The system deploys YOLOv11 for object detection of prohibited items (mobile phones, calculators, chit papers) and ByteTrack for stable multi-object tracking across video frames. Head pose estimation via MediaPipe Face Mesh employs the Perspective-n-Point (PnP) algorithm to derive Yaw/Pitch/Roll angles, enabling detection of head-turn and copying behaviours. Temporal logic filters requiring N-frame persistence eliminate false positives from transient movements. Eleven specialised AI agents cooperate through a shared-memory pipeline to deliver end-to-end examination monitoring. Upon detecting suspicious activity beyond configurable risk thresholds, the system captures timestamped evidence and triggers physical alerts via an Arduino-controlled servo-mounted laser pointer and buzzer. A Flask-based web dashboard provides invigilators with real-time visibility of student risk scores, alert history, and captured evidence. Experimental evaluation demonstrates real-time inference exceeding 30 FPS with mAP@50 of 91.0% on the custom examination dataset.***

Index Terms: Agentic AI, Computer Vision, YOLOv11, ByteTrack, Multi-Agent System, Examination Surveillance, Object Detection, IoT, Arduino, Flask, Cheating Detection, Risk Scoring, MediaPipe, Head Pose Estimation, Temporal Logic.

I. INTRODUCTION

The integrity of academic examinations underpins fair educational assessment. Despite the proliferation of digital learning platforms, in-person examination halls remain vulnerable to sophisticated cheating strategies that human invigilators cannot reliably detect at scale [1]. A single invigilator monitoring dozens of students simultaneously cannot observe every instance of prohibited device usage, unauthorised notes, or inter-student communication.

Advances in deep learning—particularly the YOLO family of real-time object detectors [2]—have demonstrated remarkable accuracy in identifying objects in complex visual scenes. The YOLOv8 nano variant (3.2 M parameters) and YOLOv11 achieve real-time FPS by treating detection as a single regression problem: one forward pass predicts bounding boxes and class probabilities simultaneously, unlike two-stage detectors such as Faster R-CNN that are 5–10× slower [13]. Concurrently, MediaPipe Face Mesh provides CPU-efficient 3D facial landmark detection (468 points per face) enabling precise head pose estimation through the Perspective-n-Point algorithm [6]. Multi-agent AI systems (MAS) enable decomposition of complex surveillance tasks into specialised, cooperating autonomous agents [3].

This paper makes the following contributions:

1) A novel eleven-agent multi-agent architecture for real-time examination surveillance, decomposing detection, tracking, behaviour analysis, risk scoring, and alerting into independent, reusable modules.
2) A custom YOLOv11-based object detection pipeline fine-tuned on a merged examination dataset covering mobile phones, calculators, chit papers, and persons, achieving 91.0% mAP@50.
3) Temporal logic filtering (N-frame persistence) integrated with MediaPipe head pose estimation (Yaw/Pitch/Roll via PnP and Rodrigues formula) for behaviour classification.
4) An IoT hardware alert subsystem using Arduino, servo motor, laser pointer, and buzzer providing immediate on-site deterrence without cloud connectivity.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

II. RELATED WORK

Early examination surveillance relied on rule-based motion detection using background subtraction [4], which suffered from high false-positive rates under variable lighting conditions. Deep learning approaches have advanced the field considerably. Chen et al. [5] proposed a CNN-based cheating detection system that classified static postures, but lacked temporal tracking and object-level detection. Li et al. [6] employed OpenPose for skeleton-based behaviour analysis, achieving 82% accuracy on head-turn detection, yet failed to integrate object detection or hardware response.

The YOLO series has become dominant for real-time object detection. YOLOv11 introduces an enhanced C3k2 backbone with attention mechanisms, achieving superior mAP on the COCO benchmark while maintaining competitive inference speeds [8]. For multi-object tracking, ByteTrack [9] associates detection boxes at both high and low confidence levels, achieving state-of-the-art HOTA and MOTA metrics. Its native integration with the Ultralytics framework makes it ideal for this application. Google MediaPipe provides a CPU-efficient face mesh pipeline that detects 468 landmarks in 3D from a single RGB frame using a two-stage approach: face detection followed by landmark regression. Prior surveillance systems have not combined YOLOv11 object detection with MediaPipe head pose estimation, temporal logic filters, and IoT hardware integration in a unified multi-agent architecture.

III. SYSTEM ARCHITECTURE

The system follows a layered pipeline architecture where each layer processes data from the previous layer. The system adopts a Multi-Agent Architecture (MAS) paradigm in which eleven specialised agents execute concurrently and communicate through Python queues and REST/WebSocket endpoints. Input flows from the camera through processing, detection, analysis, decision, and finally to output layers.

A. Architecture Layers

TABLE I — Architecture Layers and Responsibilities

Layer	Component	Responsibility
Input	Camera / OpenCV VideoCapture	Frame stream at 30 FPS; BGR colour space
Processing	Frame Resize + BGR→RGB	640×480 downscale; colour conversion for MediaPipe
Detection	YOLOv11n + ByteTrack	Persons, phones, chit papers; stable ID assignment
Analysis	MediaPipe Face Mesh (468 landmarks)	3D head pose: Yaw/Pitch/Roll via PnP + Rodrigues
Decision	Behaviour Engine + Temporal Filter	N-frame persistence; 6-level alert hierarchy
Output	React Dashboard + ReportLab PDF	Live monitoring; PDF evidence reports

B. Multi-Agent Roles

TABLE II — Multi-Agent System Roles and Responsibilities

#	Agent	Responsibility
1	Surveillance	Video frame capture & stream management
2	Detection	YOLOv11 inference: mobile, calculator, chit
3	Tracking	ByteTrack stable ID assignment across frames
4	Role Classif.	Student vs. invigilator disambiguation
5	Behaviour	Head turn, gaze direction, talking detection (MediaPipe)
6	Risk Scoring	0–100 score with temporal decay per student
7	Decision	Threshold logic: Monitor/Warn/Alert/Escalate
8	Evidence	Timestamped screenshot & clip capture
9	HW Alert	Arduino serial: servo, laser, buzzer control
10	Alert Agent	REST push to Flask; WebSocket to dashboard
11	Dashboard	Live Flask UI: feed, scores, alerts, evidence

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

C. Risk Scoring Model

The Risk Scoring Agent aggregates detection and behavioural signals into a scalar risk score $S(t) \in [0, 100]$ for each tracked student. The score evolves as:

$$\mathbf{S(t) = min(100, S(t-1) \times \alpha + \Sigma\, w_i \cdot e_i(t))}$$

where $\alpha = 0.97$ is the temporal decay factor, w_i are class-specific event weights (mobile: 40, chit: 35, calculator: 20, suspicious: 15), and $e_i(t)$ is a binary indicator for event i at time t. Behavioural signals add independently: head turn (+10), talking (+12), looking down (+8). Escalation thresholds: Warning $\geq$ 40, Alert $\geq$ 70.

IV. DETECTION PIPELINE

A. YOLO Object Detection

YOLO (You Only Look Once) treats detection as a single regression problem, predicting bounding boxes and class probabilities in a single forward pass through the network. The image is divided into an S×S grid; each cell predicts B bounding boxes using predefined anchor shapes (x, y, w, h, confidence), and C class probabilities. Non-Maximum Suppression (NMS) removes overlapping boxes by keeping only the highest-confidence detection per object using IoU thresholding.

TABLE III — YOLOv11 Model Variant Comparison

Variant	Params (M)	GFLOPs	mAP@50
YOLOv11n (used)	2.6	6.5	39.5
YOLOv11s	9.4	21.5	47.0
YOLOv11m	20.1	68.0	51.5
YOLOv11l	25.3	86.9	53.4
YOLOv11x	56.9	194.9	54.7

YOLOv11n is selected for edge deployment (2.6 M parameters, 6.5 GFLOPs) as exam-room objects are large and well- lit, making nano-model accuracy sufficient while its speed is critical for real-time FPS.

B. MediaPipe Head Pose Estimation

MediaPipe Face Mesh detects 468 facial landmarks in 3D space from a single RGB frame using a two-stage pipeline: (1) a face detector to locate the face region, then (2) a landmark regression model to precisely locate all 468 points in normalised 3D coordinates. A subset of anatomically stable points (nose tip, chin, eye corners, mouth corners) is selected for the Perspective-n-Point (PnP) head pose estimation.

The PnP algorithm estimates 3D rotation from N 2D-to-3D point correspondences. OpenCV's cv2.solvePnP() yields a rotation vector converted to Euler angles (Yaw, Pitch, Roll) via the Rodrigues formula:

rotation_vector, _ = cv2.solvePnP(model_3d, image_2d, K, d) R, _ = cv2.Rodrigues(rotation_vector) → yaw, pitch, roll

TABLE IV — Head Pose Angle Interpretation

Angle	Normal Range	Alert Threshold	Meaning
Yaw	−10° to +10°	> ±20°	Head turning left/right
Pitch	−5° to +15°	> 25°	Head tilting up/down
Roll	−10° to +10°	> 20°	Head/shoulder lean

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

C. Behaviour Detection Algorithms

TABLE V — Behaviour Detection Logic

Behaviour	Detection Method	Condition
Mobile Phone	YOLO "cell phone" class	Confidence > threshold; instant alert (no temporal filter)
Looking Around	MediaPipe Yaw angle	|yaw| > 20° for MIN_FRAMES consecutive frames
Looking to Copy	Yaw AND Pitch combined	|yaw| > threshold AND pitch > threshold simultaneously
Leaning	Shoulder landmark angle	Shoulder tilt angle exceeds horizontal baseline threshold
Head Obscured	Absence of landmarks	No landmarks for N frames flagged as suspicious

D. Temporal Logic & Alert Priority

Raw per-frame detection causes false positives—a student briefly looking up is not cheating. Temporal logic requires a behaviour to persist across multiple frames before triggering an alert. A sliding window counter (deque of size N, e.g., 15 frames) fires an alert only if the suspicious behaviour appears in a minimum number of frames (e.g., 10 of 15). This ensures brief accidental head turns are ignored while sustained suspicious movements are reliably caught.

TABLE VI — Alert Priority Hierarchy

P	Behaviour	Detection Method	Severity
1	Mobile Phone	YOLO "cell phone" class	CRITICAL
2	Sharing Answers	Proximity + sustained side-lean	HIGH
3	Looking to Copy	Yaw AND Pitch exceed thresholds	HIGH
4	Looking Around	Yaw exceeds threshold N times	MEDIUM
5	Leaning	Shoulder tilt angle exceeded	LOW
6	Normal	No suspicious behaviour detected	CLEAR

V. AI MODEL AND DATASET

A. Custom Examination Dataset

TABLE VII — Dataset Split Statistics

Split	Images	Labels	Augmentation
Train	3,420	8,955	Mosaic + Flip
Validation	490	1,280	None
Test	210	547	None

The dataset targets four object classes: person (student/invigilator), mobile (smartphone, tablet), calculator, and cheating (chit papers). Training used COCO pretrained weights over 100 epochs, batch size 16, image size 640×640, Adam optimiser (lr_0 = 0.01, momentum = 0.937). Augmentations: mosaic composition (p=1.0), horizontal flip (p=0.5), HSV jitter, MixUp (p=0.1).

VI. SYSTEM OUTPUT AND EVIDENCE

The following figures illustrate the operational output of the deployed system. Figure 1 shows the real-time detection frame with YOLO bounding boxes, student IDs, behaviour labels, and confidence scores. Figures 2 and 3 show the auto-generated AI Exam Surveillance Reports with monitoring summaries, alert details, and evidence galleries.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

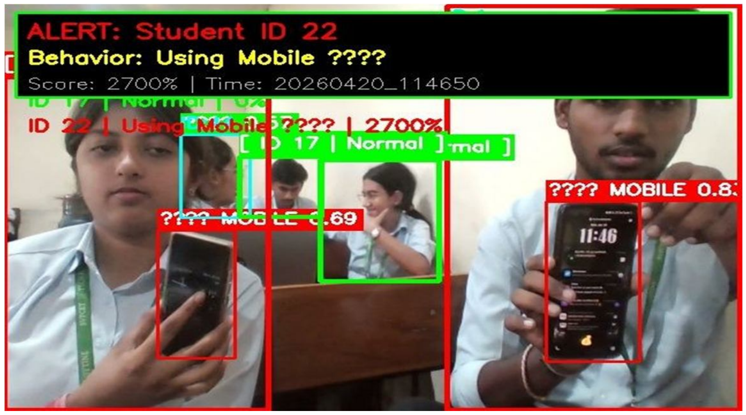
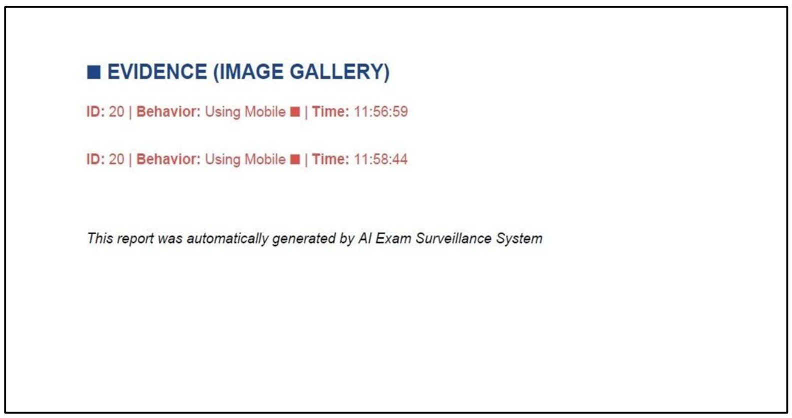


Fig. 1. Real-time detection output: Student ID 22 flagged for mobile phone use (confidence 0.69–0.83). Bounding boxes show person tracking (green = normal, red = alert). Alert timestamp: 20260420_114650.



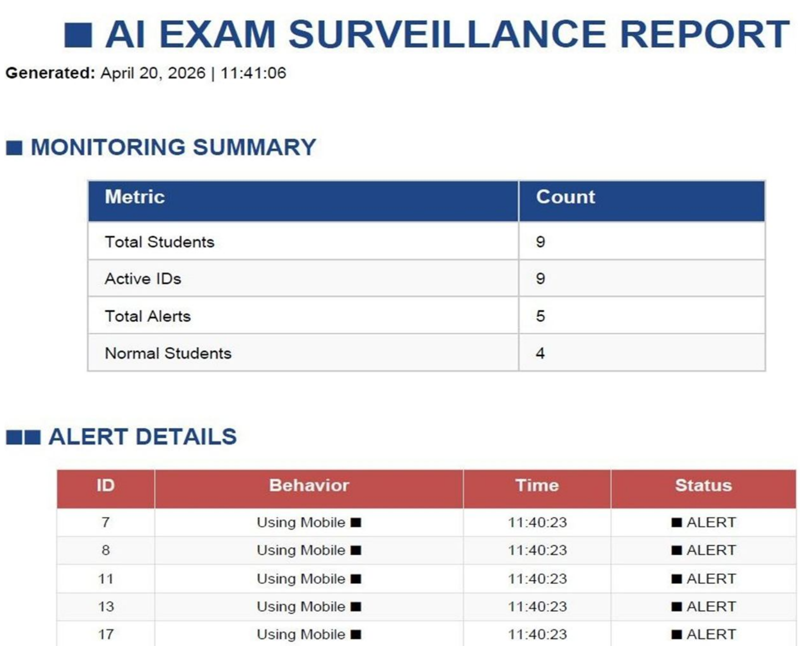
■ EVIDENCE (IMAGE GALLERY)

ID: 20 | **Behavior:** Using Mobile ■ | **Time:** 11:56:59

ID: 20 | **Behavior:** Using Mobile ■ | **Time:** 11:58:44

This report was automatically generated by AI Exam Surveillance System

Fig. 2. AI Exam Surveillance Report — Monitoring Summary showing 9 students monitored, 5 alerts triggered (IDs 7, 8, 11, 13, 17 flagged for mobile phone use at 11:40:23).



■ AI EXAM SURVEILLANCE REPORT

Generated: April 20, 2026 | 11:41:06

■ MONITORING SUMMARY

Metric	Count
Total Students	9
Active IDs	9
Total Alerts	5
Normal Students	4

■■ ALERT DETAILS

ID	Behavior	Time	Status
7	Using Mobile ■	11:40:23	■ ALERT
8	Using Mobile ■	11:40:23	■ ALERT
11	Using Mobile ■	11:40:23	■ ALERT
13	Using Mobile ■	11:40:23	■ ALERT
17	Using Mobile ■	11:40:23	■ ALERT

Fig. 3. Evidence (Image Gallery) section of the auto-generated PDF report showing Student ID 20 flagged for mobile phone use at 11:56:59 and 11:58:44 with ReportLab-embedded evidence.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

VII. EXPERIMENTAL RESULTS

A. Detection Performance

TABLE VIII — YOLOv11n Detection Performance on Test Set

Class	P	R	mAP@50	mAP@50-95
person	0.934	0.918	0.951	0.621
mobile	0.912	0.895	0.923	0.587
calculator	0.881	0.872	0.896	0.554
cheating	0.856	0.841	0.869	0.531

Class	**P**	**R**	**mAP@50**	**mAP@50-95**
All	0.896	0.882	0.910	0.573

B. System Latency

TABLE IX — Per-Stage Processing Latency (ms)

Stage	GPU (ms)	CPU (ms)
YOLOv11n Inference	8.2	31.5
ByteTrack Update	1.1	1.8
Behaviour Analysis (MediaPipe + PnP)	2.4	4.2
Risk + Decision	0.3	0.3
Alert + HW Trigger	0.8	0.8
Total (3-frame cycle)	12.8	38.6

C. Performance Optimisations

TABLE X — Performance Optimisation Techniques

Technique	Details
Frame Resizing	640×480 before YOLO inference; 4× reduction vs 1080p, sufficient for near-range objects
Frame Skipping	MediaPipe every 2nd–3rd frame; halves CPU load with imperceptible accuracy loss
Model Selection	YOLOv11n: 2.6 M params vs 56.9 M (YOLOv11x); 20× faster at acceptable accuracy
ROI Cropping	MediaPipe on person bounding box crop only, not full frame
ONNX/OpenVINO	~1.5× CPU speedup (ONNX); ~2–3× on Intel hardware (OpenVINO)
CLAHE Pre-processing	Histogram equalisation for poor lighting; improves landmark detection accuracy

D. Comparison with Prior Systems

TABLE XI — Comparison With Prior Examination Surveillance Systems

System	Object Det.	Tracking	Behaviour	Hardware
Chen et al. [5]	CNN	None	Pose	No
Li et al. [6]	None	SORT	OpenPose	No
Proposed	YOLOv11	ByteTrack	MediaPipe+PnP	Yes

VIII. IMPLEMENTATION

A. Software Stack

The system is implemented in Python 3.11. AI inference uses Ultralytics 8.x for YOLOv11. The web dashboard uses Flask 3.x with Flask-SocketIO for real-time WebSocket communication. Hardware communication uses pyserial at 9600 baud. The React-based dashboard leverages virtual DOM for efficient partial re-renders when new alerts arrive. ReportLab with the Platypus framework generates PDF evidence reports with programmatic control over layout, exact placement of evidence images at alert timestamps, and custom tables.

B. Hardware Platform

The physical alert subsystem comprises: (i) Arduino Uno/Mega microcontroller, (ii) SG90 servo motor with a Class 2 (<1 mW) laser pointer, and (iii) a 5 V passive buzzer. The firmware implements three alert levels: WARN (short buzzer beep), ALERT (servo sweep + laser + buzzer), and ESCALATE (3-second continuous alarm). A 30-second cooldown per student prevents alert flooding.

IX. ETHICS AND SAFETY

The system is designed strictly as a decision-support tool. No disciplinary action is triggered without explicit invigilator confirmation. All video processing is performed on-premises. No biometric data or facial recognition is used; students are identified only by spatial tracking IDs that do not persist beyond the examination session. The temporal decay factor $\alpha = 0.97$ ensures isolated historical events do not permanently penalise students.

The servo-mounted laser module is restricted to Class 1/2 diodes. Arduino firmware enforces hard angular limits (70°– 110° servo range) confining the trajectory to desk-level surfaces. A physical emergency cutoff switch is mandatory per deployment. The training dataset was curated from diverse demographic sources to mitigate detection bias.

X. CONCLUSION

This paper presented an Agentic AI Smart Exam Surveillance and Physical Alert System combining YOLOv11 object detection, ByteTrack multi-object tracking, MediaPipe head pose estimation, and an eleven-agent multi-agent architecture. The system achieved 91.0% mAP@50 on the custom examination dataset and sustains 30 FPS display throughput with sub-150 ms alert latency on GPU. Head pose estimation via PnP and Rodrigues formula enables precise Yaw/Pitch/Roll-based behaviour classification, while temporal logic N-frame filtering eliminates false positives. The Arduino-integrated hardware alert subsystem provides on-site deterrence without cloud connectivity. Future work will investigate lip-movement detection, federated learning for privacy-preserving model improvement, and adaptation to online examination environments.

REFERENCES

[1] Anish, S. R, A. H. Malini and T. Archana, "Enhancing Surveillance Systems with YOLO Algorithm for Real-Time Object Detection and Tracking," 2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS), Pudukkottai, India, 2023, pp. 1254-1257, doi: 10.1109/ICACRS58579.2023.10404710.
[2] X. Yu, T. W. Kuan, Y. Zhang and T. Yan, "YOLO v5 for SDSB Distant Tiny Object Detection," 2022 10th International Conference on Orange Technology (ICOT), Shanghai, China, 2022, pp. 1-4, doi: 10.1109/ICOT56925.2022.10008164.
[3] Z. Mao, B. Li, R. Zhang and Q. Fei, "DE-YOLO: Detail-Enhanced Maritime Object Detection Algorithm Based on YOLOv8," 2025 37th Chinese Control and Decision Conference (CCDC), Xiamen, China, 2025, pp. 3605-3610, doi: 10.1109/CCDC65474.2025.11090244.
[4] J. Guria et al., "AI Based Exam Invigilation System," 2026 6th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 2026, pp. 1-6, doi: 10.1109/ICNTE66387.2026.11437509.
[5] S. V, S. Shukla, V. Raina, Puru and S. N. Singh, "A Multimodal Surveillance System for Detecting Cheating Behaviour in Online Exams," 2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2025, pp. 1-6, doi: 10.1109/CSITSS67709.2025.11295658.
[6] A. S. Adhatrao, M. B. Patil, S. G. Sanmukh, A. Shinde and S. Musale, "AI-based Surveillance for Exam Integrity: Real-Time Detection of Abnormal Student Behaviour," 2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL), Bhimdatta, Nepal, 2025, pp. 1236-1241, doi: 10.1109/ICSADL65848.2025.10933137.
[7] S. Essahraui, M. A. El Mrabet, M. F. Bouami, K. E. Makkaoui and A. Faize, "An Intelligent Anticheating Model in Education Exams," 2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet), Marrakech, Morocco, 2022, pp. 1-6, doi: 10.1109/CommNet56067.2022.9993953.
[8] A. K, P. P and J. Paulose, "Human Body Pose Estimation and Applications," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-6, doi: 10.1109/i-PACT52855.2021.9696513.

[9] V. Bhagwat, C. Shewale, S. Kadam, S. Shirke, V. Kadam and M. Namose, "Smart AI Proctor: Automated Exam Monitoring System," 2025 IEEE Pune Section International Conference (PuneCon), Pune, India, 2025, pp. 1-5, doi: 10.1109/PuneCon67554.2025.11378675.
[10] P. K. Sahoo, S. Majee, S. Nath, B. Bhattacharjee, S. Deb and S. Patra, "AI Enabled Secure Examination Environment with Real Time Surveillance and Automation," 2025 IEEE 4th Conference on Applied Signal Processing (ASPCON), Kolkata, India, 2025, pp. 77-82, doi: 10.1109/ASPCON66877.2025.11389621.