An Intelligent Real-Time System for Exam Cheating Detection using Computer Vision and Deep Neural Networks.

Abdulghani A. Abied (1)
a.abied@azu.edu.ly
Department of computer science, Al-Zaytouna University, Tarhuna, Libya

AllaEddin A. Gribi (2)
Alla.Gribi@Alrefak.edu.ly
Department of computer science, AlRefaK University,Tripoli, Libya

Malak T. Khalaf (3)
malak.khalaf@Alrefak.edu.ly
Department of computer science, Al-Refak University,Tripoli, Libya

Abstract:

The rapid digital transformation in education has introduced new challenges to academic integrity during examinations. Traditional monitoring that relies solely on human supervision is no longer sufficient due to the large number of students and increasingly sophisticated cheating methods. This study aims to develop an integrated real-time cheating detection system based on computer vision and deep learning. The system combines object detection (YOLOv11), pose estimation (Media Pipe), head orientation analysis using the Perspective-n-Point algorithm, and (Ghostface Net) for facial recognition.

The system processes live video streams to identify suspicious behaviors such as looking away, hand proximity to the face, and interaction with unauthorized objects like mobile phones and books. A multi-component suspicion scoring mechanism assigns relative weights: head orientation (75%), hand proximity to the face (15%), and hand-object interaction (10%), producing a unified suspicion score. The system sustains real-time operation (16.98 FPS), ensuring smooth display and immediate responsiveness through a Flask-based interface, while reducing false alarms via temporal smoothing with Exponential Moving Average (EMA) and a deduplication mechanism.

Furthermore, the system adheres to privacy standards by storing only suspicious frames and automatically deleting data to safeguard student information. This work represents a practical and academic contribution to strengthening academic integrity and reducing human errors in exam monitoring.

Keywords: Cheating detection, Computer vision, YOLOv11, Media Pipe, Real-time system, Academic integrity.

المستخلص :

لقد أحدث التحول الرقمي السـريع في التعليم تحديات جديدة أمام النزاهة الأكاديمية أثناء الامتحانات. لم تعد المراقبة التقليدية التي تعتمد فقط على الإشـراف البشـري كافية بسـبب الأعداد الكبيرة للطلاب وأسـاليب الغش المتطورة. يهدف هذا البحث إلى تطوير نظام متكامل للكشـف عن الغش في الزمن الحقيقي يعتمد على الرؤية الحاسـوبية والتعلم العميق. يدمج النظام تقنيات كشـف الأجسـام(YOLOv11) ، تقدير الوضـعيات (Media Pipe)، تحليل اتجاه الرأس باسـتخدام خوارزمية Perspective-n-Point، وتقنية (Ghostface Net) للتعرف على الوجه.

يقوم النظام بمعالجة بث الفيديو المباشـر لتحديد السـلوكيات المشـبوهة مثل النظر بعيدًا، قرب اليد من الوجه، والتفاعل مع الأشـياء غير المصـرح بها مثل الهواتف المحمولة والكتب. تعتمد آلية حسـاب درجة الشـك متعددة المكونات على أوزان نسبية: اتجاه الرأس (75%)، قرب اليد من الوجه (15%)، وتفاعل اليد مع الأشياء (10%) لإنتاج درجة شك موحدة.

ويحقق النظام معدل إطارات يدعم التشـغيل في الزمن الحقيقي (16.98 إطار/ثانية) يضـمن سـلاسـة العرض واسـتجابة فورية عبر واجهة برمجية مطورة باسـتخدام Flask، مع تقليل الإنذارات الكاذبة عبر التنعيم الزمني باستخدام متوسط الحركة الأسي المعدل (EMA) وآلية إزالة التكرار.

كما يلتزم النظام بمعايير الخصـوصـية عبر سـياسـة يتم فيها تخزين الإطارات المشـبوهة فقط مع حذف تلقائي للبيانات لضـمان أمن معلومات الطلاب. يمثل هذا العمل مسـاهمة عملية وأكاديمية في تعزيز النزاهة الأكاديمية وتقليل الأخطاء البشرية في مراقبة الاختبارات.

الكلمات المفتاحية: كشـف الغش، الرؤية الحاسـوبية، YOLOv11، Media Pipe، نظام في الزمن الحقيقي، النزاهة الأكاديمية.

1. Introduction

Academic integrity is the cornerstone of higher education quality and the credibility of scientific assessments. With the accelerated shift toward digital and blended learning, significant challenges have emerged in securing examinations against increasingly sophisticated cheating practices. Traditional proctoring that depends solely on human supervision is no longer sufficient due to large student numbers, limited field of view, and human errors caused by fatigue or distraction (Abdulkarim, Kuliya & Adam, 2025). To address these challenges, recent research

has demonstrated the potential of computer vision and deep learning to automate exam monitoring and analyze human behavior in real time (Abdulkarim et al., 2025). Building on this foundation, this study presents the development of an advanced system that integrates object detection, pose estimation, head orientation analysis, and face recognition within a unified pipeline. The main contribution is a weighted multi-component suspicion scoring mechanism combined with temporal smoothing and alert deduplication, which reduces false positives while maintaining high detection sensitivity. Privacy is preserved by storing only flagged frames and deleting them automatically after a defined retention period.

1.1. Research Objectives

- **To design and implement** a real-time cheating detection system using computer vision and deep learning.
- **To integrate** object detection (YOLOv11), pose estimation (Media Pipe), head orientation analysis (PnP), and hybrid face recognition (Ghostface Net) into a single unified pipeline.
- **To develop** a weighted multicomponent suspicion scoring algorithm (head 75%, hand to face 15%, hand to object 10%) with temporal smoothing modified (EMA).
- **To reduce** false positives through a smart alert deduplication mechanism.
- **To enforce privacy** by storing only flagged frames and deleting them automatically after seven days.
- **To evaluate** the system's accuracy, latency, and resource consumption on standard hardware.

1.2. Research Questions

This research seeks to answer the following scientific questions:

1. To what extent can deep neural network models (e.g., YOLOv11 and Media Pipe) detect suspicious behaviors with accuracy comparable to or exceeding human proctoring?
2. How can head pose estimation algorithms be integrated with object detection to create a comprehensive and reliable behavioral evaluation model?
3. What is the technical impact of temporal smoothing techniques in reducing error rates and improving system stability in exam environments with varying lighting and density conditions?
4. How can the strict requirements of exam monitoring be balanced with the need to protect student data privacy and ensure digital security?

2 . Related work

The detection of cheating behaviors during examinations has become a critical research area due to the increasing digitisation of education and the growing sophistication of cheating methods. Traditional proctoring that depends solely on human supervision has proven inadequate, being constrained by limited visual coverage, fatigue, and the inability to monitor many students simultaneously.

In recent years, research has increasingly shifted toward automated systems based on computer vision and deep learning (e.g., Imah et al., 2025). Hu et al. (2024) proposed a multi-perspective adaptive detection system using SwinTransformer and lightweight YOLOv5-CA, designed to eliminate blind zones by employing three cameras per student that switch detection models based on gaze direction. While the system achieved 35 FPS and 95% behavioral accuracy, its requirement for three cameras per student makes it hardware-intensive and impractical for large-scale deployment. Similarly, Zuo et al. (2024) developed an improved YOLOv8 model with Efficient Channel Attention (ECA) enabling recognition of subtle cues such as passing notes or hidden devices, achieving 27 FPS and 85.67% mAP50. However, their approach suffered from higher false positive rates compared to hierarchical models.

Several studies have addressed the challenge of subtle behavior recognition. Xu et al. (2025) introduced a hierarchical pipeline combining frame differencing with MLP-YOLOv8-SENetV2 and ResNet, achieving 45 FPS and 96% mAP50 by filtering redundant frames before deep analysis. Despite its efficiency, the system remained sensitive to extreme lighting and heavy occlusion. Adhikari (2024) focused on identity verification by integrating YOLO with face recognition models, achieving 89% accuracy and 91% precision, but with a recall of only 83%, indicating that many cheating events were missed.

More recent work has moved toward multi-modal architectures that combine object detection with behavioral analysis. Abdulkarim et al. (2025) fused a fine-tuned YOLOv8 detector for prohibited objects with an Open Pose-based pose estimation pipeline, using a rule-based decision module to generate real-time alerts, and reported a precision of 0.92, a recall of 0.88, and an average latency of 35 ms per frame. Nevertheless, their rule-based fusion yields binary alerts rather than a graduated measure of suspicion, and the reported recall indicates that roughly one in eight cheating events was still missed. Shiddiq and Kurniasih (2025) instead prioritised computational efficiency, replacing the CSPDarknet53 backbone of YOLOv5 with Mobile One to reach 98.1% accuracy while reducing computational complexity for resource-constrained deployment; however, their pipeline relies exclusively on object detection and therefore cannot capture behavioral cues, such as sustained head rotation, that occur without a visible prohibited object. Imah et al. (2025) developed a smart CCTV monitoring system based on deep transfer learning that issues alert for suspicious actions including consulting notes, talking to peers, and using mobile phones, yet the system classifies discrete action categories rather than producing a continuous, weighted suspicion measure, and no explicit temporal smoothing is reported to suppress alerts arising from brief, incidental movements.

Despite these advances, most existing systems share several common limitations. First, they rely on binary classification, deciding whether a frame or clip is "cheating" or "normal". This approach often produces high false positive rates because a single momentary movement can trigger an alert. Second, few systems address temporal noise caused by natural jitter in human movement, leading to unstable detections. Third, the majority ignore privacy concerns by storing complete video footage indefinitely, raising ethical and regulatory issues under frameworks such as GDPR. Fourth, most implementations still use older architectures such as YOLOv5 or YOLOv8, with limited exploration of the latest algorithmic improvements (Shiddiq & Kurniasih, 2025; Abdulkarim et al., 2025).

A notable advancement that addresses these gaps is the weighted multi-component scoring approach adopted in the present study. Unlike binary systems, our work introduces a nuanced suspicion score that combines head orientation (75%), hand-face proximity (15%), and hand-object interaction (10%), allowing proctors to see not just an alert but a graduated degree of concern. Temporal smoothing using Exponential Moving Average (EMA) together with a deduplication mechanism reduces false positives caused by transient movements. Privacy is enforced through selective storage of only flagged frames with automatic deletion after seven days. Moreover, our system integrates YOLOv11, the most recent iteration of the YOLO architecture, providing state-of-the-art detection efficiency. Experimental results show that this configuration achieves precision, recall, and F1 score of 1.0 on a curated evaluation set with real-time performance (16.98 FPS) and median latency of 39.54 ms on standard hardware (NVIDIA RTX 3070). Building on these innovations, the present study demonstrates that a carefully tuned weighted scoring system with temporal smoothing and privacy-aware data handling can overcome the key limitations of existing exam proctoring solutions.

3. Methodology & Implementation

The system is engineered as a real-time video processing pipeline. A single camera, operating at 720p and 30 frames per second, continuously feeds frames into a sequence of detection and analysis modules. The system computes a suspicion score and, upon exceeding a predefined threshold, generates an alert accompanied by a rationale and supporting evidence. Figure 1 illustrates the high-level architecture of the overall frame flow throughout a detection session.

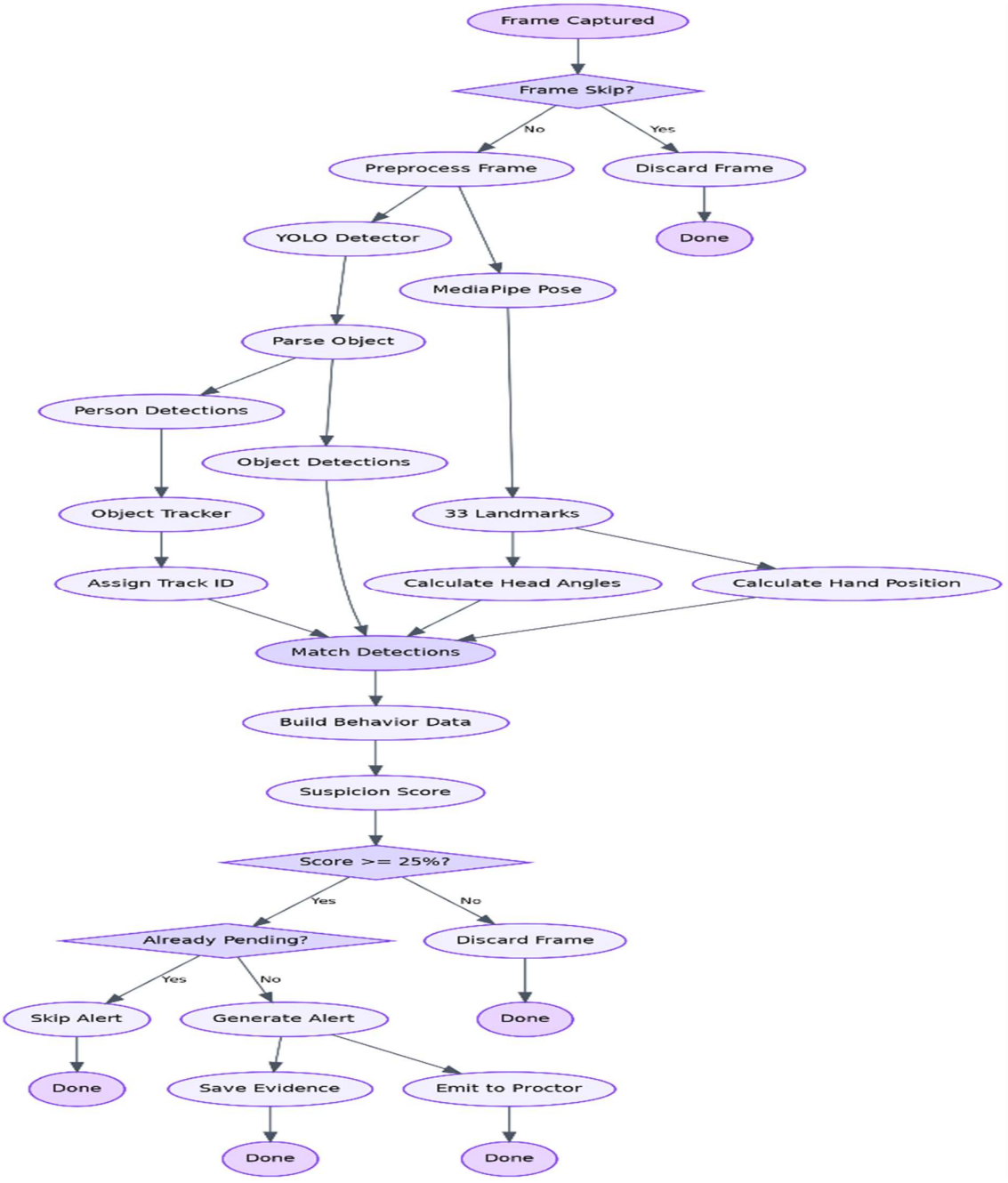


Figure 1: High level System Architecture Flowchart illustrating the data transfer from the camera to the evaluation algorithm.

3.1. System modules

To achieve robust and real-time detection of examination irregularities, the proposed framework is structured into a comprehensive modular pipeline. Each component handles a distinct stage of spatial perception, temporal analysis, tracking, behavioral scoring, and secure data handling. This architecture integrates state-of-the-art computer vision models—ranging from efficient object detection and pose estimation to optimized face recognition—alongside robust tracking, temporal smoothing, and privacy-first evidence preservation mechanisms to ensure both high accuracy and ethical compliance. Below is a detailed overview of the core modules comprising the system:

3.1.1. Object detection (YOLOv11-Large)

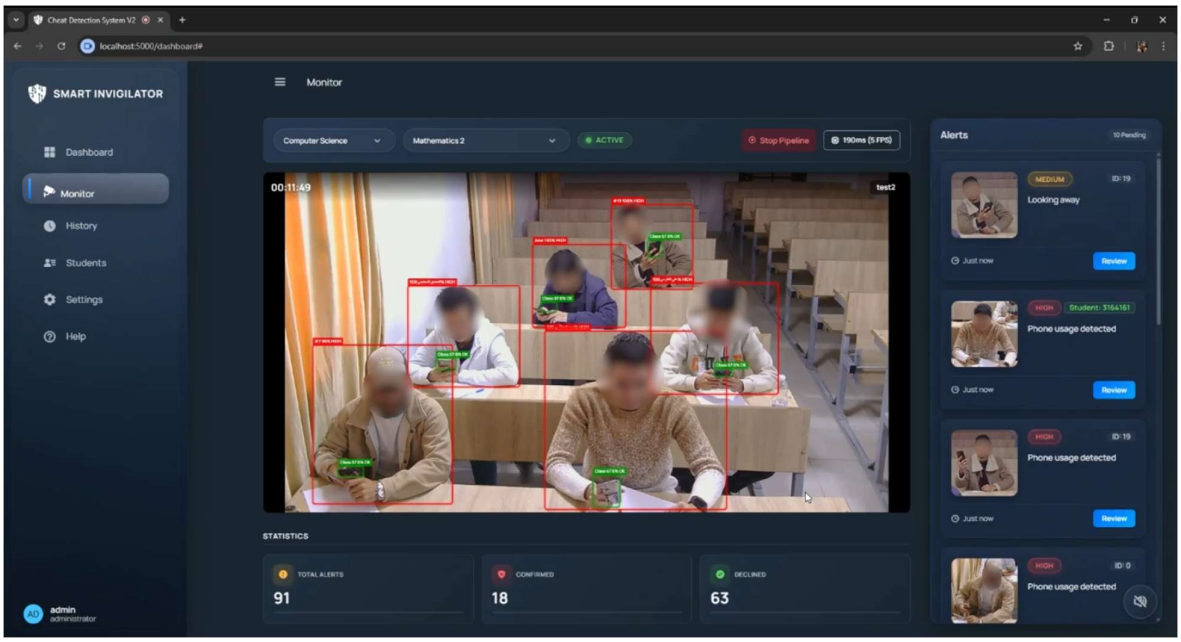


Figure 2: Example of object detection using YOLOv11, showing the identification of a mobile phone and a book with bounding boxes.

The video data were recorded in real time in an examination hall at Alrefak University using the same single camera described above (720p at 30 fps). Six volunteers from the course cohort participated in a staged one-hour exam. Participants enacted both predefined cheating behaviors (e.g., consulting unauthorized devices, exchanging notes, using concealed aids) and unscripted sequences in which they were briefed only on the intended class and then improvised freely, deliberately attempting to make the behavior difficult for the system to classify; the pipeline monitored and recorded their actions throughout to evaluate real-time detection performance. Recordings were streamed directly to the detection modules to reproduce operational timing and latency; all streams were time-stamped and stored on secure university servers. Trained annotators labeled cheating events using a standardized codebook (behavior type, onset time, duration) to produce ground truth for evaluation. All participants provided informed consent, identifying information was removed prior to analysis, and the study was conducted under Alrefaq University institutional ethical oversight.

We used Ultralytics YOLOv11 with the large model variant. Input images are resized to 1024×1024px and 1280×1280px. The detector runs on an NVIDIA RTX 3070 (CUDA). As shown in **Figure 2,** It identifies persons, mobile phones, books, and other unauthorized objects. On our benchmark, YOLO inference averaged 23.7 ms per frame.

The object detector is the pretrained Ultralytics YOLOv11-Large checkpoint (yolo11l.pt, Ultralytics 8.3.24), loaded directly from Ultralytics and used as-is on its original COCO classes without any custom fine-tuning; the relevant classes for this task are person, cell phone, and book, and detections are filtered at a confidence threshold of 0.5. Pose estimation uses Media Pipe Blaze Pose (Media Pipe 0.10.14) out of the box. The supporting stack is PyTorch 2.4.0 (CUDA 12.1), OpenCV 4.10.0.84, and TensorFlow 2.16.1. No library was internally modified; the contribution is the integration and the weighted scoring layer built on top of these components.

3.1.2. Pose estimation (MediaPipe BlazePose)

MediaPipe runs in parallel on the same frame. **Figure 3** represents the extraction of 33 body landmarks, hand keypoints, and head pose (yaw, pitch, roll) using the Perspective-n-Point algorithm. Pose estimation took about 16 ms per frame on average.

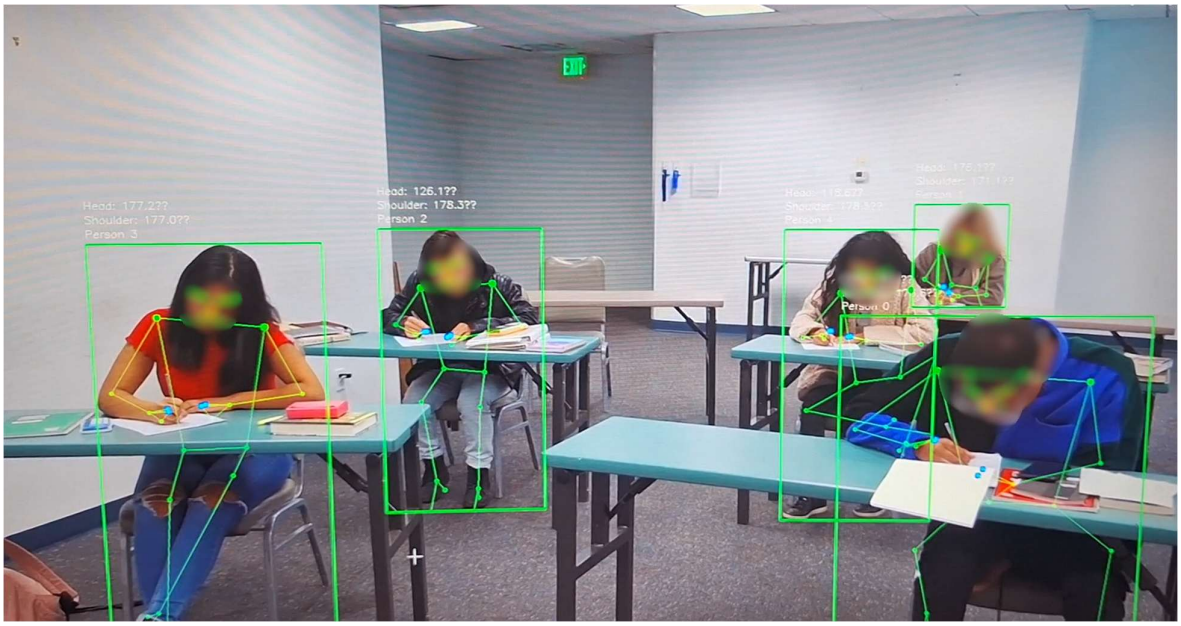


Figure 3: Object detection and pose estimation in an exam hall, showing bounding boxes and labeled landmarks for monitoring students.

3.1.3. Face recognition (Ghostface Net with caching)

We integrated Ghostface Net (Alansari et al., 2023) for student identification. Ghostface Net is a lightweight face recognition architecture that leverages Ghost modules to generate redundant feature maps through inexpensive linear transformations, achieving computational efficiency of 60–275 MFLOPs while maintaining competitive accuracy on benchmark datasets (Alansari et al., 2023). To avoid running the model on every frame, we cache the mapping from a track ID to a student ID. The model runs only when a new person enters the scene or when the cache expires. Median face recognition cost is near zero, but occasional frames take much longer – up to 644 ms in our logs. This happens when the cache misses and the face is at a difficult angle.

3.1.4. Tracking (centroid-based)

We implemented a simple centroid tracker. It assigns a unique track ID to each person and keeps it across frames as long as the person stays in the camera view. Tracking added only 0.07 ms on average.

3.1.5. Weighted suspicion scoring

For every person in the frame, we compute a raw suspicion score as:

raw = 0.75·head+ 0.15·hand_face_ + 0.10·hand_object

These component weights (0.75 for head orientation, 0.15 for hand–face proximity, and 0.10 for hand–object interaction) were selected manually through the iterative tuning process described above rather than by an automated search such as grid search or cross-validation. They are exposed as adjustable parameters in the system's settings (config.json), allowing the operator to re-tune the weights to suit different exam environments for a flexible configuration.

where:

- **head** is normalised head-orientation deviation (yaw/pitch beyond baseline)
- **hand_face** is the inverse of hand-to-face distance (closer = higher)
- **hand_object** is the proximity to detected phones or books, weighted by object type (phone = 1.0, book = 1.0).

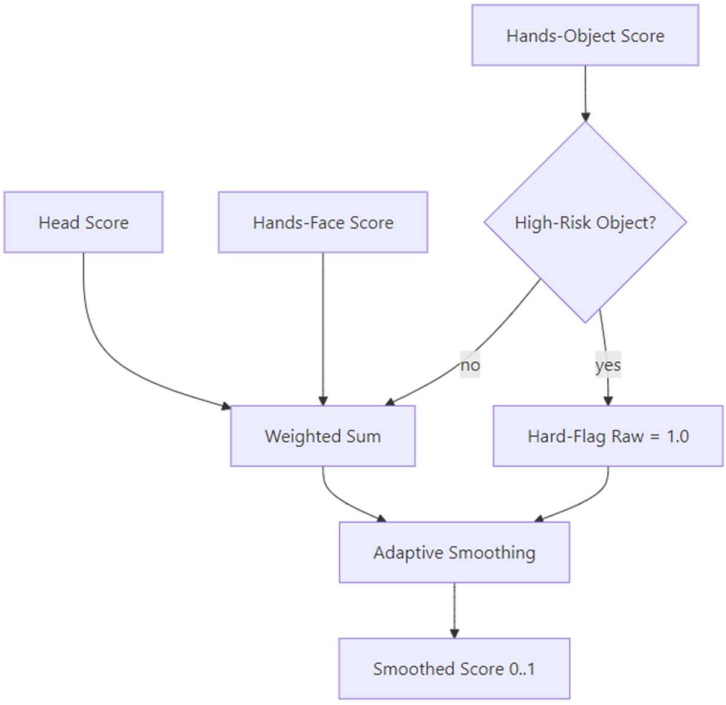


Figure 4: Weighted suspicion scoring with hard-flag and adaptive smoothing.

Figure 4 simplifies how scoring components accumulate to represent a normalized number for the final decision to decide whether the current person is cheating or in neutral form except when a high-suspicion object interaction triggers a hard flag, setting the score to 1.0 to indicate a 100% cheating action .

3.1.6. Temporal smoothing and deduplication

We apply modified Exponential Moving Average (EMA) to the raw score to reduce jitter:

$$\text{smoothed} = \alpha \cdot \text{raw} + (1-\alpha) \cdot \text{previous smoothed} \quad (\alpha = 0.3)$$

Then we check a deduplication map. If the same student has already triggered an alert for the same behavior type within a short window while the same behavior type is still pending in the alert panel, we skip sending another alert. This prevents flooding the proctor with repeated notifications.

3.1.7. Privacy-first evidence storage

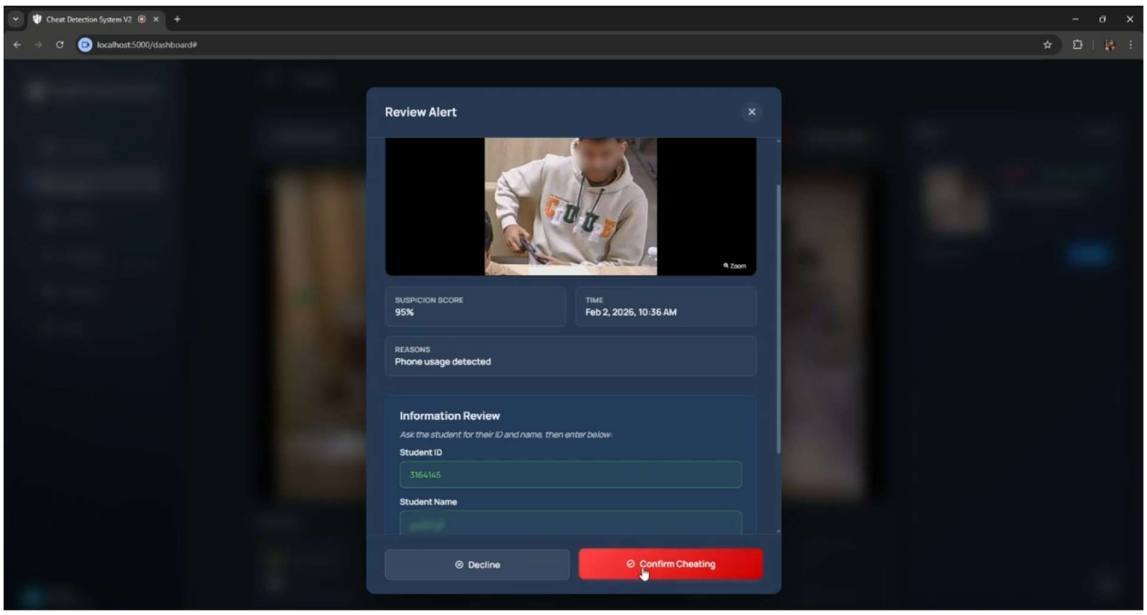


Figure 5: Evidence card generated by the cheating detection system, displaying a flagged alert with suspicion score, timestamp, and reason (phone usage) for review and confirmation

When an alert is raised, we save only the flagged frame (full frame) and a crop of the student's face/body, as shown in **Figure 5**. We do not store the raw video stream. All evidence is automatically deleted after seven days. This policy is our answer to the privacy-surveillance paradox that most existing systems ignore (European Union, 2016).

3.2 Ethical Consideration and Privacy Protection

3.2.1 Ethical Approval and Consent

Data collection for this study was conducted with strict adherence to ethical standards regarding participant privacy. All video recordings used to test and evaluate the proposed real-time cheat detection system were collected inside the university from a group of student volunteers who agreed in advance to participate in a pre-arranged test of the system, under explicit informed consent. No third-party or covertly recorded footage was used.

3.2.2 Privacy Preservation and Anonymization

To address the severe privacy risks associated with automated face recognition and monitoring systems, the system applies the principle of data minimization: it does not store the raw video stream, and retains only the minimal evidence required for review (the flagged frame and a cropped region), which is automatically deleted after a seven-day retention period, in line with GDPR principles (European Union, 2016). The pipeline itself does not perform automatic face-blurring; for this publication, the faces shown in the figures were blurred manually to protect the privacy of the participating students.

3.2.3 Legal and Regulatory Compliance

The deployment of AI-based surveillance technologies in educational settings must align with international and regional data-protection regulations such as the GDPR(European Union, 2016). To this end, the system processes all video streams locally, in real time, on the institution's own hardware, with no cloud transmission of student imagery. Student identification relies on a small, local, access-controlled face-embedding database used only to match enrolled students during the exam session; these embeddings are not shared externally and cannot be accessed or reverted to its original state. This local, edge-style processing significantly reduces the risk of data leaks and unauthorized access to personal biometric information, and no unauthorized recording or surveillance was performed during the study.

3.3 Technical challenges and our solutions

Implementing a real-time computer vision and deep learning framework for exam monitoring in practical environments entails several critical engineering hurdles. To build a robust system capable of operating under these constraints, we addressed these key obstacles through the following targeted solutions:

- **False positives from natural movements**

In the base configuration, a student raising a hand or briefly looking at a wristwatch could be mistaken for holding a phone. This happened because the hand-object interaction term was too sensitive.

Our fix: We increased the minimum confidence threshold for phone detection and tightened the hand-object interaction gate, while keeping the component weights at their configured values. This is why Round 3 achieved zero false positives without losing true positives.

- **Long-tail latency in face recognition.**

Most frames had negligible face recognition cost (median 0 ms). However, the log shows that sometimes the system retries face matching (attempt 1/3, 2/3) and occasionally fails to recognise the student (confidence 0.00). In rare frames, recognition took up to 644 ms.

Our mitigation: We increased the cache duration and made face recognition asynchronous – the suspicion score uses the cached ID; if the cache is stale, the system continues with the Track ID while updating in the background. This reduced the visible latency spikes, but they still appear in the 95th percentile.

- **Temporal smoothing did not dramatically improve clip-level decisions**

We initially expected EMA smoothing to significantly reduce false alarms. On our curated set, the raw scores and smoothed scores agreed on every clip. The real improvement came from threshold tuning and confidence gates, not from the temporal filter.

What we learned: Smoothing is still useful for continuous video streams because it makes the score less jittery, but it is not a substitute for careful threshold setting. We kept it in the pipeline because it stabilises the display on the proctor dashboard.

- **Resolution and speed trade-off**

We tested three capture resolutions: 640×480, 1280×720, and 1920×1080. Surprisingly, 640×480 was the slowest (14.19 FPS, 68 ms latency), while 720p gave the best balance (16.98 FPS, 42 ms). 1080p delivered almost the same frame rate as 720p (16.94 FPS) but with higher latency (54 ms).

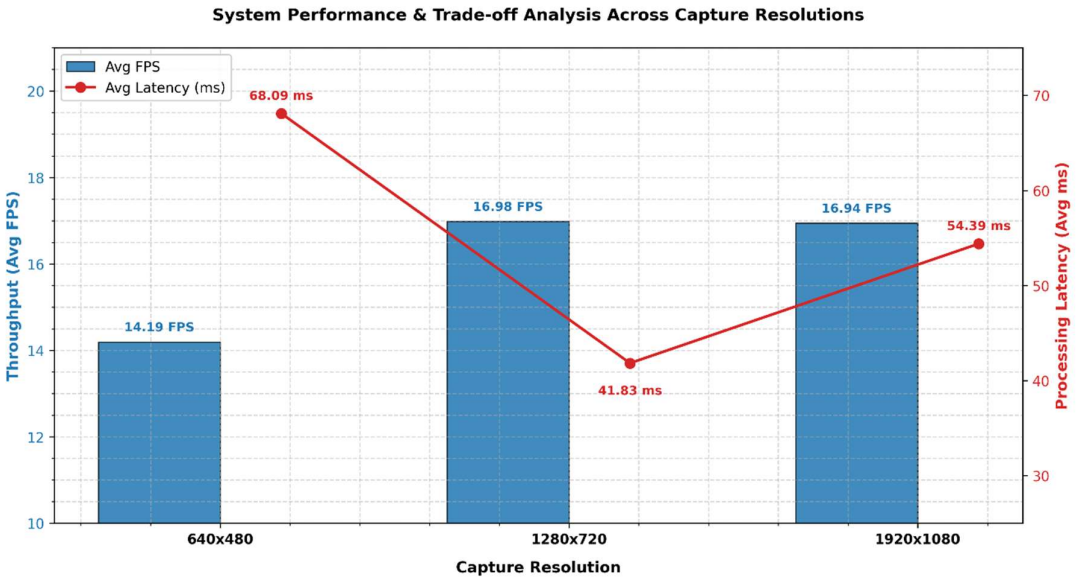


Figure 6: System throughput (average FPS) and processing latency (average ms) measured at runtime across three capture resolutions (640×480, 1280×720, 1920×1080). 1280×720 provides the best speed–latency balance and is used as the default configuration.

Conclusion: We chose 720p as the default. The lower resolution was slower because the detector spent relatively more time on CPU/GPU synchronization, not less.

- **Warm-up delay**

```
Warming up AI Pipeline (pre-allocating GPU memory)...
YOLO Engine Warm
Pose Estimation Warm
Pipeline Ready (Warmup took 1.36s)
```

Figure 7: Initialization log of the AI pipeline showing GPU memory pre-allocation, and readiness confirmation

Before the first frame is processed, the system loads the YOLO and Media Pipe models into GPU memory, as shown in **Figure 7**, This short warm-up prepares the pipeline to begin the detection session immediately at peak performance, instead of starting cold and risking missed early events. For a long exam session, 1.36 seconds is acceptable and practically invisible to the proctor.

What we would do differently

Looking back, we would improve:

- Asynchronous face recognition from the start. The current implementation sometimes blocks the main pipeline when the cache misses. A fully asynchronous design would hide the long-tail latency completely.

4. Results and Analysis

We evaluated the system in two stages: a clean-start runtime benchmark and a tuning study on a curated labeled clip set. Our goal was to measure how far the system could be improved through configuration changes alone, without changing the core detection pipeline.

4.1 Experimental Setup Summary

This section summarizes the environment used for the benchmark round.

4.1.1 Hardware and Software

Table 1: Hardware and software environment used for benchmark evaluation.

GPU	CPU	RAM (GB)	OS	Python	Libraries
NVIDIA RTX 3070	AMD Ryzen 5 5600X	16.0	Windows-11	3.11.9	YOLOv11, MediaPipe BlazePose, DeepFace, OpenCV, psutil, NVML

4.1.2. Stress Benchmark base round test

Table 2: Stress benchmark results for the base round, showing frame throughput and skip rate over a 60-second run.

Round	Duration (s)	Frames Read	Frames Processed	Avg Stream FPS	Avg Pipeline FPS	Avg Skip Rate
Round 1	60.00	1028	1019	17.13	16.98	0.88%

4.1.3 Evaluation protocol

Table 3: Evaluation protocol parameters, including capture resolution, model configuration, and lighting conditions.

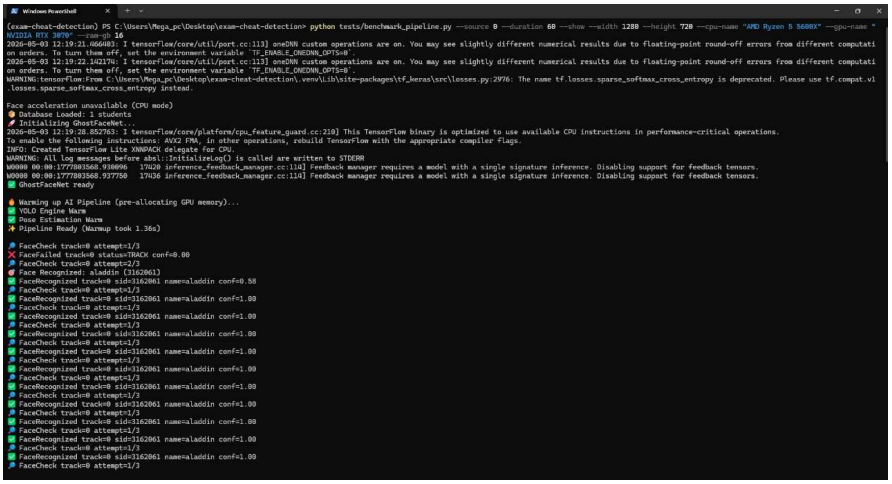
Test Duration (min)	Capture Resolution	Camera FPS	Model Family	Model Size	Image Size	Sessions	Lighting
1.00	1280x720	30.0	YOLOv11 large	large	1024	1	Normal indoor lighting

We ran the benchmark after a clean restart to reduce background noise. GPU utilization and memory usage were captured automatically, and pose estimation was included in the reported pipeline timing.

4.2. Performance Metrics

4.2.1. Measurement instrumentation

All performance and accuracy metrics reported in this section were produced by a custom evaluation script (tests/benchmark_pipeline.py) developed for this study. The script executes the full pipeline against a specified capture source and resolution, logs per-frame timing for each pipeline stage, samples system resource utilisation, and writes the results to a timestamped metrics file for offline analysis. Hardware utilisation was captured using psutil for CPU and RAM and NVIDIA NVML for GPU load and memory. Precision, recall, F1-score, and false-positive rate were computed directly from the TP/FP/TN/FN counts using their standard definitions rather than through an external machine-learning



library.

Figure 8: A terminal script that shows the way system running the pipeline on the annotated clips to generate performance matrix

Ground-truth labels for the evaluation clips were produced by trained annotators using the standardized codebook described in Section 3.1.1, which records behavior type, onset time, and duration for each cheating event.

4.2.2. Throughput and Latency

Table 4: Overall throughput, latency, and resource utilisation measured during the benchmark run.

Scenario	Stream FPS	Pipeline FPS	Median latency (ms)	P95 latency (ms)	GPU utilization (%)	CPU utilization (%)
Overall	17.13	16.98	39.54	43.60	34.84	30.23

4.2.3. Latency distribution

Table 5: Distribution of total pipeline latency across processed frames.

Metric	Mean (ms)	Stdev (ms)	Min (ms)	Max (ms)	Median (ms)	P95 (ms)
Total pipeline	41.83	33.98	25.57	775.81	39.54	43.60

4.2.4 Model Inference Timing

Table 6: Average per-frame inference time for each pipeline stage.

SCENARIO	YOLO (MS)	MEDIAPIPE (MS)	TRACKING (MS)	FACE RECOGNITION (MS)	BEHAVIOR + SCORING (MS)	EVIDENCE (MS)
AVG PER FRAME	23.81	16.01	0.07	1.73	0.05	0.00

4.2.5. Component timing statistics

Table 7: Detailed timing statistics for each pipeline component.

Component	Mean (ms)	Stdev (ms)	Min (ms)	Max (ms)	Median (ms)	P95 (ms)
YOLO inference	23.81	5.68	22.13	203.71	23.30	25.15
Postprocess	15.63	2.72	0.02	57.26	15.82	16.59
Filter overlaps	0.00	0.00	0.00	0.01	0.00	0.00
Tracking	0.07	0.03	0.00	0.16	0.06	0.12
MediaPipe pose	16.01	1.50	0.00	41.95	15.90	16.62
Face recognition	1.73	31.84	0.00	644.74	0.00	0.11
Behavior + scoring	0.05	0.02	0.00	0.22	0.04	0.09
Evidence saving	0.00	0.00	0.00	0.00	0.00	0.00

The runtime profile is dominated by detector inference and post processing. The remaining stages are comparatively small, so throughput is driven mainly by the object detector rather than the scoring logic.

4.3. Detection Accuracy and Tuning Study

Assessing the real-world viability of our deep learning-based proctoring system requires a multi-faceted performance analysis. The following Detection Accuracy and Tuning Study is structured to thoroughly examine system behavior.

4.3.1. Curated evaluation set

Table 8: Composition of the curated 20-clip evaluation set by clip group.

Clip group	Count	Percentage
Hard normal	10	50.0%
Normal	3	15.0%
Cheating	7	35.0%
Total	20	100.0%

To evaluate the system under rigorous and challenging conditions, we constructed a curated evaluation set comprising 20 video clips, as detailed in Table 8. The dataset features a deliberate distribution: **Hard normal** clips account for 50.0% (10 clips) of the data, alongside standard **Normal** cases at 15.0% (3 clips) and **Cheating** scenarios at 35.0% (7 clips).

This distribution ensures the model is tested against 'hard negatives' (challenging benign cases) providing a more rigorous measure of the false-positive rate, rather than just easy normal cases. That makes the false-positive rate more meaningful. The inclusion of adversarially recorded hard clips was a deliberate design choice: rather than measuring performance only on clearly separable cases, the set was constructed to probe the system's behaviour at the decision boundary, where benign actions superficially resemble cheating and cheating attempts are deliberately concealed.

4.3.2. Detection Accuracy and Processing Speed during Evaluation Clips

We used round-based naming for the tuning passes to provide a clear iterative progression of the system’s optimization. Round 1 removed most false positives compared with the base configuration, but Round 2 was too strict and lost recall. Round 3 restored recall while preserving zero false positives, which is the strongest operating point we reached.

The evaluation set comprised 20 video clips. The four columns reported above correspond to four configuration passes of the same pipeline (Base and Rounds 1–3), not to twenty distinct models; no change was made to the underlying model architecture between passes, and all differences arise from threshold, image-size, and confidence-gate settings.

Metric	Base	Round 1	Round 2	Round 3
Image Size	1024	1280	1280	1280
Alert Threshold	30	20	20	20
TP	7	7	4	7
FP	5	1	0	0
TN	11	15	16	16
FN	0	0	3	0
Precision	0.5833	0.8750	1.0000	1.0000
Recall	1.0000	1.0000	0.5714	1.0000
F1 Score	0.7368	0.9333	0.7273	1.0000
FP Rate	0.3125	0.0625	0.0000	0.0000
Avg FPS	21.2941	12.8857	13.7842	15.1271
Avg Proc MS	45.5423	90.8228	72.9891	81.4725
Total Flagged Frames	5034	2206	209	1088

Table 9: Detection accuracy and processing speed across the four configuration passes (Base and Rounds 1–3).

4.3.3. Threshold sweep on the tuned configuration

We also swept the alert threshold around the tuned configuration to confirm the stable operating band.

Table 10: Threshold sweep around the tuned configuration, showing the stable operating band.

Threshold band	TP	FP	TN	FN	Precision	Recall	F1	FP rate
10-21	7	1	15	0	0.8750	1.0000	0.9333	0.0625
22	6	1	15	1	0.8571	0.8571	0.8571	0.0625

This shows that the useful operating band is broad around 20, but performance begins to drop once the threshold is pushed too high.

4.3.4. Confusion matrix at the final operating point

Table 11: Confusion matrix at the final operating point for raw and smoothed scoring.

Mode	TP	FP	TN	FN
Raw score	7	0	16	0
Smoothed score	7	0	16	0

At the final operating point, raw and smoothed scoring produced the same clip-level decision.

4.3.5. Component contribution

Table 12: Final component weights and their observed contribution to the suspicion score.

Component	Final weight	Observed impact
Head orientation	0.75	Remained the dominant contributor to the suspicion score.
Hand-face proximity	0.15	Helped capture distraction behavior, but stayed secondary.
Hand-object interaction	0.10	Gated by a higher object-confidence threshold to suppress weak phone-like false positives.

The final improvement did not come from changing the core detector. It came from tuning the object-confidence gates and the scoring thresholds.

4.3.6. Qualitative output examples

Figure 5 shows a representative true-positive detection at the operating point used for the final configuration, presenting the flagged frame together with the computed suspicion score, timestamp, and the triggering reason surfaced to the proctor. Evidence frames generated during the earlier base configuration, which produced five false positives, were removed under the seven-day automatic retention policy described in Section 3.2.2 and are therefore not reproduced here.

4.4. Temporal Smoothing Impact

Temporal smoothing did not change the clip-level decision on this curated set. The raw and smoothed scores agreed at the final operating point, which means the main gains came from threshold and confidence tuning rather than temporal filtering alone.

4.5. Resolution Sensitivity

We ran separate resolution tests to compare throughput and latency under different capture sizes.

Table 13: Throughput and latency sensitivity across the three tested capture resolutions.

Resolution	Avg FPS	Avg Latency (ms)	Frames	Videos
640x480	14.19	68.09	21,690	20
1280x720	16.98	41.83	21,690	20
1920x1080	16.94	54.39	21,690	20

The 720p base benchmark remained the reference point for the main runtime section, while the 640x480 and 1920x1080 measurements show how the same pipeline behaves when the capture resolution is lowered or increased.

4.6. Hardware usage during run

Table 14: Hardware resource utilisation recorded during the benchmark run.

Resource	Mean	Stdev	Min	Max	Median	P95
CPU utilization (%)	30.23	3.12	19.3	36.2	30.9	34.63
RAM utilization (%)	55.30	0.48	51.9	55.6	55.4	55.60
GPU utilization (%)	34.84	9.76	0.0	41.0	39.0	41.0
GPU memory used (MB)	1897.43	19.40	1751.0	1900.0	1900.0	1900.0

4.7 Error Analysis

The main runtime irregularity was the long-tail face-recognition latency. The median face-recognition cost was effectively near zero, but rare frames reached several hundred milliseconds but these spikes do not affect the overall real time feel of the system, which suggests occasional expensive recognition calls or retries. The overall pipeline latency stayed stable enough for real-time processing, with the bulk of frames clustered around the low-40 ms range.

CPU and RAM usage were more stable. GPU utilization averaged about 35%, while GPU memory usage stayed around 1.9 GB, which indicates that the detector was running on CUDA with moderate load rather than saturating the device.

4.8. Discussion

The results demonstrate that the pipeline maintains a sufficient frame rate for real-time surveillance applications at 1280x720 capture resolution with the large YOLO configuration. The initial base configuration was overly noisy, especially on benign clips where a raised hand could resemble a phone. After tuning, the false positives dropped sharply, and the final configuration recovered full recall while preserving zero false positives on the curated evaluation set.

The strongest result is the final round: it preserved TP=7 and removed all false positives, which gives Precision=1.0, Recall=1.0, and F1=1.0 on the curated set. That is the clearest evidence that the system is tunable through configuration alone.

Table 15: Quantitative comparison between the proposed system and prior work. Throughput measured on the clean-start 720p benchmark (Table 4). The tuned Round 3 configuration, which produced the reported accuracy figures, averaged 15.13 FPS at 1280 px input.

Study	Approach	Accuracy / mAP	Precision	Recall	F1	FPS
Hu et al. (2024)	SwinTransformer + YOLOv5-CA, 3 cameras/student	95% accuracy	—	—	—	35
Zuo et al. (2024)	YOLOv8 + Efficient Channel Attention	85.67% mAP50	—	—	—	27
Xu et al. (2025)	Frame differencing + MLP-YOLOv8-SENetV2 + ResNet	96% mAP50	—	—	—	45
Adhikari (2024)	YOLO + face recognition	89% accuracy	91%	83%	—	not reported
Abdulkarim et al. (2025)	YOLOv8 + OpenPose, rule-based fusion	—	92%	88%	—	~29*
Shiddiq & Kurniasih (2025)	YOLOv5 + MobileOne backbone	98.1% accuracy	—	—	—	not reported
This work	YOLOv11-L + MediaPipe BlazePose + weighted scoring	—	1.00	1.00	1.00	16.98

As summarised in Table 15, the proposed system attains the highest precision and recall of the compared approaches on its evaluation set, at the cost of the lowest frame rate. Two caveats should be noted when interpreting this comparison. First, the reported metrics are not measured on a common benchmark: each system was evaluated on its own dataset, so the values indicate general performance levels rather than a controlled head-to-head result. Second, the perfect precision and recall reported here were obtained on a curated 20-clip set, which is substantially smaller than the datasets used in several of the compared studies, and the throughput figure reflects the use of the large YOLOv11 variant at 1280×720 rather than a lightweight configuration optimised for speed. The comparison therefore positions the present work as favouring detection reliability and graduated scoring over raw throughput.

Finally, the evaluation phase was conducted on a limited set of 20 video clips. While the preliminary results confirm the efficiency of the YOLOv11–MediaPipe pipeline, a larger sample size is required to comprehensively validate the system's scalability and generalizability across diverse academic settings.

5. Conclusion

This study presented a real-time cheating detection system that combines object detection, pose estimation, head-pose analysis, and face recognition within a single configurable pipeline. The experimental results showed that the system can be tuned effectively through thresholds, confidence gates, and resolution settings to balance sensitivity and false alarms without modifying the core architecture. The final tuned configuration achieved the best balance we observed in this study. Compared with the base round, the pipeline became more selective, then more conservative, and finally reached a configuration that preserved all positive clips without raising false alerts on the curated evaluation set. In practical terms, the system can be tuned through thresholds, confidence gates, and resolution settings to trade-off between sensitivity and false alarms without changing the model architecture.

and it also sustained real-time operation on standard hardware. The resolution tests further showed that 720p provides the best practical trade-off between speed and latency, while the privacy-aware evidence handling ensures that only suspicious frames are retained and deleted automatically after the retention period. Overall, the proposed system offers a practical and effective approach for improving academic integrity in exam environments.

6. Future Recommendations

While the proposed system demonstrates strong performance in real-time cheating detection, several directions remain open for future exploration:

- **Dataset Expansion and Fine-Tuning:** Collect larger, more diverse datasets from real exam environments to fine-tune YOLOv11 and MediaPipe models, improving robustness across lighting, cultural, and behavioral variations.
- **Multi-Camera Integration:** Explore multi-perspective setups to reduce blind spots and enhance detection accuracy without significantly increasing hardware costs.
- **Adaptive Weight Optimization:** Replace manual suspicion weight tuning with automated methods such as reinforcement learning or Bayesian optimization to dynamically adjust scoring in different exam contexts.
- **Explainable AI (XAI):** Incorporate interpretable models or visualization tools to help proctors understand why a particular behavior was flagged, increasing trust and transparency.
- **Privacy-Enhancing Techniques:** Investigate real-time anonymization methods (e.g., face blurring or differential privacy) to further reduce ethical concerns while maintaining detection accuracy.

- **Integration with Institutional Systems:** Link alerts and evidence storage with existing Learning Management Systems (LMS) for streamlined reporting and auditing.
- **MAR (Mouth Aspect Ratio):** Incorporate MAR analysis to detect subtle mouth movements (e.g., whispering, speaking to hidden devices, or signaling), which could further strengthen behavioral monitoring in exam settings

7. References

1. Abdulkarim, A., Kuliya, M., & Adam, A. B. (2025). Real-time AI-driven exam cheating detection using YOLO and pose estimation: A multi-modal deep learning approach. Iconic *Research and* Engineering Journals*,* *9*(6), 847–856. https://doi.org/10.64388/IREV9I6-1712742
2. Adhikari, A. (2024). YOLO-based face recognition for automatic cheating detection in examination environments. International Journal of Electrical and Data Communication, *5*(2), 29–32.
3. Alansari, M., Hay, O. A., Javed, S., Shoufan, A., Zweiri, Y., & Werghi, N. (2023). GhostFaceNets: Lightweight face recognition model from cheap operations. IEEE Access, 11, 35429–35446. https://doi.org/10.1109/ACCESS.2023.3266068
4. European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L 119, 1–88. https://eur-lex.europa.eu/eli/reg/2016/679/oj
5. Hu, Z., Jing, Y., Wu, G., & Wang, H. (2024). Multi-perspective adaptive paperless examination cheating detection system based on image recognition. Applied Sciences*,* 14(10), 4048. https://doi.org/10.3390/app14104048
6. Imah, E. M., Puspitasari, R. D. I., Annisa, F. Q., Prayogi, H., & Fitriyaningsih, I. (2025). Smart CCTV exam monitoring to detect and alert suspicious activities using deep transfer learning. Procedia Computer Science, 269, 805–814. https://doi.org/10.1016/j.procs.2025.09.023
7. Shiddiq, M. I., & Kurniasih, A. (2025). Enhancing exam cheating detection with MobileOne: A YOLOv5 algorithm approach to abnormal behavior recognition. In Proceedings of the 2024 Brawijaya International Conference (BIC 2024*)* (pp. 3–16). Atlantis Press. https://doi.org/10.2991/978-94-6463-854-7_2
8. Xu, E., Lu, J., Xu, S., & Wang, J. (2025). Cheating recognition in examination halls based on improved YOLOv8. Discover Computing*,* *28*(1), 256. https://doi.org/10.1007/s10791-025-09747-3
9. Zuo, Y., Chai, S. S., & Goh, K. L. (2024). Cheating detection in examinations using improved YOLOv8 with attention mechanism. Journal of Computer Science, 20(12), 1668–1680. https://doi.org/10.3844/jcssp.2024.1668.1680