Anti-Plagiarism System for Exam Monitoring

Valentin Pletea-Marinescu
National University of Science and Technology
POLITEHNICA Bucharest
Email: valentin.pletea@stud.acs.upb.ro

Ge Su
Zhejiang University
Email: suge@zju.edu.cn

Ștefan-Dan Ciocîrlan
National University of Science and Technology
POLITEHNICA Bucharest
Email: stefan_dan.ciocirlan@upb.ro

Ștefan-Alexandru Mocanu
National University of Science and Technology
POLITEHNICA Bucharest
Email: stefan.mocanu@upb.ro

***Abstract*—Academic integrity represents a fundamental challenge in modern education systems, with plagiarism rates increasing globally across all educational levels. Traditional exam monitoring approaches rely primarily on screen surveillance and human oversight, creating significant vulnerabilities in detecting sophisticated cheating behaviors during online and remote examinations. This research develops a comprehensive anti-plagiarism monitoring system using deep learning and computer vision technologies to address these limitations. The system integrates gaze tracking algorithms with YOLO-based object detection models through a modular software architecture. Facial landmark detection enables precise gaze direction analysis, while specialized convolutional neural networks identify unauthorized objects and suspicious materials in the examination environment. These two complementary approaches work together to provide comprehensive monitoring coverage: gaze analysis detects abnormal visual attention patterns that may indicate unauthorized assistance seeking, while object detection identifies physical cheating aids such as smartphones and smartwatches. Experimental testing demonstrates effective operation on CPU-based systems, identifying suspicious gaze patterns, abnormal behaviors, and unauthorized objects while maintaining real-time performance. The system provides comprehensive exam monitoring capabilities suitable for educational institutions, enabling widespread deployment using standard computing hardware without requiring specialized equipment.**

***Index Terms*—Academic integrity, Educational technology, Computer vision, Gaze tracking, Object detection, Real-time systems, Machine learning, Convolutional neural networks, Image processing, Kalman Filters**

I. Introduction

Academic integrity remains a core educational challenge. Recent studies [1], [2], [9], [10] highlight increased exam fraud during post-pandemic online learning. Traditional tools fail to detect physical cheating like gaze aversion or unauthorized devices. Research demonstrates that academic dishonesty affects educational quality, credibility, and institutional reputation [8], [11]. Recent statistics reveal concerning trends, with some regions reporting plagiarism detection rates exceeding 26% of submitted academic works [10].

Traditional examination monitoring approaches present significant limitations in detecting sophisticated cheating behaviors, particularly in remote and hybrid learning contexts [4]. These approaches rely on manual supervision and screen-based surveillance, failing to detect physical cheating behaviors such as unauthorized device usage or suspicious gaze patterns, creating vulnerabilities in examination integrity [5].

While machine learning solutions exist for plagiarism detection [6], [7], most require computational resources including GPU acceleration, making them inaccessible to many educational institutions with limited hardware infrastructure.

Recent advances in computer vision and artificial intelligence have opened possibilities for automated monitoring systems. Studies have demonstrated the effectiveness of facial detection algorithms in real-time applications [12]–[14]. Research in eye-tracking technology has shown promising results for behavioral analysis in educational contexts [21]–[23].

Furthermore, object detection technologies, particularly YOLO architectures, have revolutionized real-time identification capabilities [18]. However, existing commercial solutions often require significant computational resources or rely on external cloud processing, limiting their accessibility to institutions with constrained infrastructure [19].

This paper presents an anti-plagiarism monitoring system that operates on standard CPU-based hardware, requiring minimal computational resources while maintaining high detection accuracy. The system demonstrates real-time performance on hardware configurations, including Intel i5 7th generation processors, making monitoring technology accessible to institutions regardless of their technical infrastructure limitations.

The paper is organized as follows: Section II discusses related work in computer vision fundamentals and educational technology integration. Section III presents the overall system framework and methodology. Section IV describes the technical details of the implementation. Section V evaluates the system's performance and presents experimental results. Section VI provides insights from the discussion, and Section VII concludes our work and outlines directions for future research.

II. Background

A. Facial Landmark Detection and Gaze Tracking

Facial landmark detection is fundamental for behavioral monitoring in exam environments. Modern deep learning models such as MediaPipe Face Mesh [17] can extract 468 real-time 3D facial landmarks, far surpassing traditional 68-point methods [12], [13]. This high granularity enables precise localization of eyes, iris, and facial contours, supporting robust gaze estimation even under variable lighting or head movement.

Gaze tracking leverages these landmarks to compute gaze direction vectors, identifying deviations from normal behavior (e.g., repeated side glances or downward gaze). To reduce noise and stabilize estimates in the presence of rapid movement or occlusion, advanced Kalman filters are used [14]–[16]. These filters model eye movement as a dynamic process, filtering outliers and ensuring smooth, continuous gaze tracking.

B. YOLO and Object Detection

Object detection is essential for identifying unauthorized devices (phones, smartwatches) in the exam environment. YOLO (You Only Look Once) algorithms, now at v8 [18], provide fast and accurate detection of small objects in video frames, enabling real-time processing on standard hardware.

YOLOv8 uses optimized convolutional neural network architectures for speed and accuracy, with significant reductions in parameters and computational cost compared to earlier versions. These models can be trained on custom datasets to recognize exam-specific devices, maintaining low false alarm rates even under challenging lighting or background conditions.

C. AI-Based Monitoring

Modern automated anti-plagiarism solutions use artificial intelligence to detect non-compliant behaviors during online exams. Recent systems [1]–[5] employ convolutional neural networks (CNNs) for face and object recognition, as well as LSTM networks for temporal behavior analysis (e.g., gaze or repetitive movements over time).

A major advantage of the presented approach is fully offline processing, with no dependence on cloud or external servers, ensuring data privacy and reducing costs. The system operates efficiently on standard CPU hardware, making it accessible to educational institutions with limited resources.

By combining gaze detection, object identification, and temporal analysis, the system provides comprehensive coverage of potential cheating behaviors, overcoming the limitations of traditional screen-capture or human-only supervision.

III. System Architecture and Methodology

A. Modular System Design

The anti-plagiarism system uses a modular architecture with five components: facial detection, gaze analysis, object detection, violation monitoring, and report generation. The facial detection module identifies faces and extracts 468-point landmarks (via MediaPipe Face Mesh). The gaze analysis module uses these landmarks to compute the candidate's gaze direction, flagging abnormal looking-away behavior. The object detection module employs YOLOv8 models to detect unauthorized devices (e.g., phones, watches) in the scene. The violation monitoring module fuses information from the gaze and object detectors to identify exam rule violations. Finally, the report generation module logs the session results and creates a monitoring report.

B. Gaze Analysis Subsystem

This subsystem processes MediaPipe Face Mesh landmark data to determine the candidate's gaze direction. We calculate horizontal and vertical gaze ratios based on key facial landmarks and apply Kalman filtering to smooth the results. Head orientation is compensated for using additional landmarks (e.g., chin, forehead). The module classifies gaze as left, right, center, or down, enabling detection of abnormal eye movements relative to normal exam behavior.

C. Object Detection Framework

The object detection component utilizes specialized recognition models to identify unauthorized devices within the examination environment. The framework employs separate detection pathways for different device categories, enabling optimized recognition thresholds and reduced classification errors.

The architecture supports real-time processing through intelligent frame selection strategies that balance detection accuracy with computational efficiency. The framework processes video streams at optimized intervals while maintaining continuous monitoring capabilities for immediate violation detection.

D. Integration and Performance

The system architecture implements a publisher-subscriber communication pattern that enables asynchronous coordination between detection modules and the central violation monitoring system. This architectural approach ensures system responsiveness while accommodating varying processing requirements across different detection algorithms.

The violation monitoring module serves as the central coordination point, aggregating detection results from multiple independent sources. The module applies temporal analysis to reduce false positive detections caused by brief, natural movements while maintaining sensitivity to genuine violations.

Modules communicate via PyQt5 signals, allowing asynchronous processing and a responsive UI. In real-time tests on an Intel i5 7th-generation CPU (8GB RAM), the system processes about 22 frames per second (≈ 45 ms per frame) with around 65–75% CPU utilization, enabling multi-hour continuous monitoring.

E. System Architecture Diagrams and Operational Analysis

To illustrate the system structure and operation, two complementary UML diagrams have been developed: the activity diagram and sequence diagram. These representations offer different perspectives on the architecture, from operational flow to temporal component interactions.

1) Activity Diagram and Operational Flow: The activity diagram presents the comprehensive operational flow of the anti-plagiarism system, illustrating decision points and parallel processing capabilities that distinguish this implementation from conventional monitoring solutions.

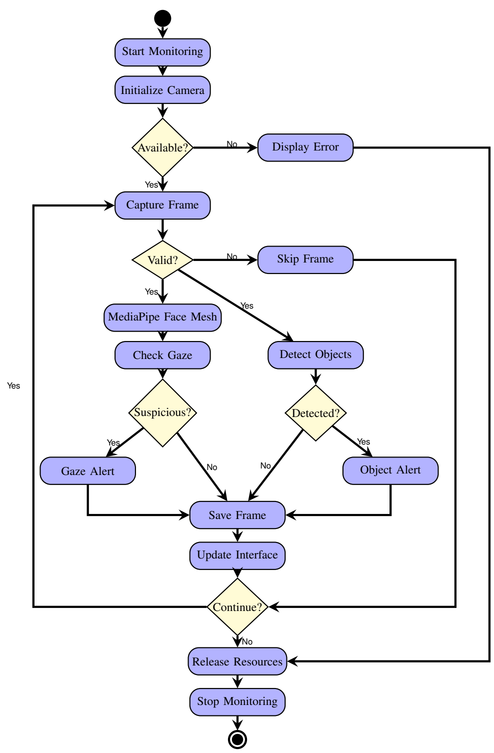

Fig. 1. Activity diagram of the anti-plagiarism system

The activity diagram demonstrates the operational flow beginning with system initialization through Start Monitoring, triggered when users activate monitoring through the graphical interface. The Initialize Camera process configures capture parameters and establishes webcam connectivity, implementing error handling through the Available decision point.

Runtime Camera Health Monitoring: The system implementation incorporates continuous camera health monitoring within the Capture Frame process. When frame capture fails repeatedly (indicating camera loss), the system triggers an emergency shutdown sequence that bypasses normal processing and routes directly to cleanup procedures. This ensures graceful termination rather than application crashes when hardware becomes unavailable during examination sessions.

Parallel Processing Architecture: The system's innovation emerges after frame validation, where processing branches into two independent parallel streams:

Behavioral Analysis Stream: Executes MediaPipe Face Mesh with 468-point facial landmark detection, followed by Check Gaze implementing the horizontal and vertical ratio algorithms for gaze direction analysis. The core behavioral monitoring focuses specifically on gaze pattern analysis to detect visual attention directed away from examination materials.

Object Detection Stream: Processes Detect Objects using dual YOLOv8 models for mobile phones and smartwatches. The Detected decision applies confidence thresholds (0.65 for both), producing Object Alert for unauthorized devices.

Robust Error Handling: Update Interface refreshes the display through PyQt5 signals, providing real-time feedback to supervisors including camera health status indicators. The monitoring loop returns to Capture Frame for sustained operation or advances to cleanup procedures when termination conditions are met, ensuring proper resource management under all operational scenarios.

2) Sequence Diagram Analysis: The sequence diagram reveals temporal interactions between system components, illustrating message passing and activation patterns critical for real-time operation with MediaPipe Face Mesh integration.

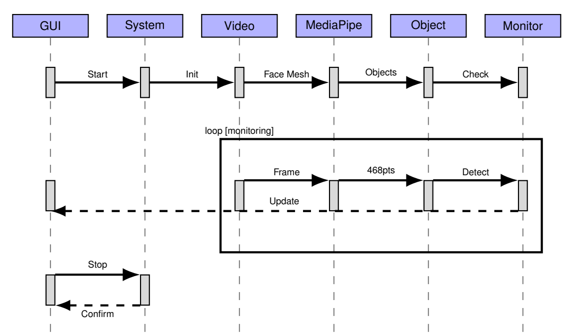

Fig. 2. Sequence diagram showing MediaPipe Face Mesh integration and temporal interactions

The sequence analysis reveals three distinct phases: initialization, continuous monitoring loop, and controlled termination.

The monitoring loop demonstrates the system's real-time capabilities, with VideoHandler continuously providing frames to MediaPipe Face Mesh for 468-point facial landmark detection, which then coordinates with ObjectDetector simultaneously.

Critical to the design is the asynchronous return path from ViolationMonitor to GUI, implementing the signal-slot mechanism of PyQt5. This ensures interface responsiveness remains independent of detection processing times, preventing UI freezing during intensive computational periods with the enhanced MediaPipe Face Mesh processing.

IV. Implementation

A. Technology Stack and Platform Compatibility

The implementation leverages Python as the primary development language, chosen for its extensive computer vision library ecosystem and rapid prototyping capabilities. Python's mature libraries including OpenCV, MediaPipe Face Mesh, and PyTorch provide robust foundations for computer vision applications while enabling efficient development cycles.

The system architecture supports multiple operating environments, with successful deployment achieved on both Windows and Linux (Ubuntu) platforms without requiring modifications to core functionality. This cross-platform compatibility ensures widespread accessibility across diverse institutional computing environments.

B. Advanced Gaze Analysis with MediaPipe Face Mesh and Kalman Filtering

The gaze analysis implementation employs MediaPipe Face Mesh technology with 468-point facial landmark detection combined with sophisticated mathematical algorithms for determining viewing direction through geometric analysis with Kalman filtering for temporal smoothing [14]–[16].

The system utilizes MediaPipe Face Mesh comprehensive facial landmark detection to calculate horizontal and vertical ratios based on facial geometry, incorporating head orientation compensation to distinguish between natural movements and suspicious behavior patterns [17].

1) MediaPipe Face Mesh Integration: The system leverages Google's MediaPipe Face Mesh solution, which provides 468 3D facial landmarks with real-time performance. This represents a significant advancement over traditional 68-point facial landmark detection methods, offering:

- Enhanced eye region mapping with detailed iris and pupil boundary detection
- Comprehensive facial contour analysis for improved head pose estimation
- Robust performance under varying lighting conditions and head orientations
- Real-time processing capabilities suitable for continuous monitoring applications

The MediaPipe Face Mesh integration utilizes key facial landmarks including nose tip (landmark 1), forehead region (landmark 10), chin area (landmark 152), and facial boundaries (landmarks 234, 454) for precise gaze direction calculation.

2) Enhanced Gaze Ratio Calculation: The system computes gaze direction using MediaPipe Face Mesh landmarks through the following equations:

$$\text{H-Ratio} = \frac{x_{\text{nose}} - x_{\text{left\_face}}}{x_{\text{right\_face}} - x_{\text{left\_face}}} \tag{1}$$

$$\text{V-Ratio} = \frac{y_{\text{nose}} - y_{\text{forehead}}}{y_{\text{chin}} - y_{\text{forehead}}} \tag{2}$$

where the facial landmarks provide more stable reference points compared to traditional eye-based calculations, resulting in improved accuracy across diverse head orientations.

3) Kalman Filter Implementation: The system implements a constant velocity motion model with state vector $\mathbf{x}_k = [x, y, v_x, v_y]^T$ encoding pupil coordinates and velocity components.

The state transition matrix:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Process noise covariance $\mathbf{Q} = 0.01 \times \mathbf{I}_4$ accounts for model uncertainties, while measurement noise adapts dynamically: $\mathbf{R} = \text{diag}(0.1/\text{confidence})$.

The filter prediction step computes:

$$\mathbf{x}_{k|k-1} = \mathbf{F}\mathbf{x}_{k-1|k-1} \tag{4}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \tag{5}$$

The correction step incorporates measurements with adaptive gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1} \tag{6}$$

where $\mathbf{H} = [1, 0, 0, 0; 0, 1, 0, 0]$ extracts position measurements.

4) Outlier Detection and Validation: Euclidean distance outlier rejection ($d > 30$ pixels) prevents spurious measurements, while velocity constraints ($v_{max} = 50$ pixels/frame) ensure physically realistic tracking. Enhanced pupil detection incorporates contour circularity validation:

$$\text{Circularity} = \frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2} \tag{7}$$

with acceptance threshold > 0.7 and area constraints (0.01-0.3 × eye region) for robust performance under challenging conditions [20].

5) Gaze Direction Classification: The system computes horizontal and vertical gaze ratios using MediaPipe Face Mesh landmarks and filtered coordinates:

$$\text{Horizontal Ratio} = \frac{x_{nose} - x_{left\_face}}{x_{right\_face} - x_{left\_face}} \tag{8}$$

$$\text{Vertical Ratio} = \frac{y_{nose} - y_{forehead}}{y_{chin} - y_{forehead}} \tag{9}$$

Classification thresholds: left (HR $>$ 0.65), right (HR $<$ 0.35), down (VR $>$ 0.6), with center classification for intermediate values. Head pose compensation adjusts thresholds based on facial orientation angles derived from MediaPipe Face Mesh comprehensive landmark geometry.

C. Dual YOLOv8 Architecture Implementation

The object detection framework employs specialized YOLOv8 models with unified 0.65 confidence threshold, optimized through systematic validation on dedicated training datasets.

Model Architecture: Dual-path detection pipeline processing smartphone (84% accuracy, 4% false positive rate) and smartwatch targets (81.6% accuracy, 3% false positive rate) with independent confidence thresholds and post-processing validation.

Training Performance: Smartphone model achieves mAP@0.5 of 0.936 with final training losses (box: 0.799, classification: 0.626). Smartwatch model reaches mAP@0.5 of 0.617 with convergent loss characteristics across 100 training epochs.

Training Dataset Specifications: The YOLOv8 models were trained on carefully curated datasets comprising:

- Smartphone detection: 2,400 annotated images with diverse device orientations, lighting conditions, and examination environment backgrounds
- Smartwatch detection: 1,800 annotated images capturing various wrist positions, watch face sizes, and partial occlusion scenarios
- Data augmentation techniques including rotation, scaling, and brightness adjustments to enhance model robustness
- Training/validation split of 80%/20% with stratified sampling to ensure representative distribution across all categories

Post-Processing Pipeline: Dimensional validation, aspect ratio analysis, and size filtering eliminate false positives through minimum area constraints and geometric consistency checks.

D. System Integration Architecture

The implementation leverages Python with OpenCV, MediaPipe Face Mesh, and PyTorch libraries for cross-platform compatibility (Windows/Linux). Publisher-subscriber communication enables asynchronous coordination between detection modules with PyQt5 signal-slot mechanisms preventing UI blocking during intensive processing.

V. Experimental Validation and Performance Analysis

A. Dataset and Testing

We evaluated the system using datasets collected under realistic exam conditions. The gaze dataset consisted of 1,400 images from 14 participants (100 images each for left, right, center, and down gazes). The object detection dataset comprised 3200 annotated images of smartphones and smartwatches.

B. Results

MediaPipe Gaze Detection Accuracy Results:

The gaze module achieved 92.4% accuracy (1479/1600 correct classifications) and a face detection rate of 96.4% (1542/1600). The MediaPipe integration demonstrated exceptional performance across all gaze directions [17], [22], [23]:

– **Center Gaze:** 89.5% accuracy (358/400) with 100% detection rate and stable H/V ratios (H_avg=0.512, V_avg=0.566)

– **Left Gaze:** 99.5% accuracy (398/400) with 99.5% detection rate, demonstrating excellent horizontal tracking (H_avg=0.960, V_avg=0.588)

– **Right Gaze:** 94.5% accuracy (378/400) with 96.5% detection rate and precise horizontal discrimination (H_avg=0.064, V_avg=0.583)

– **Down Gaze:** 86.2% accuracy (345/400) with 89.5% detection rate, demonstrating vertical orientation detection capabilities (H_avg=0.483, V_avg=0.681). For successfully detected faces, down gaze accuracy reaches 96.4% (345/358), indicating robust performance when facial landmarks are properly identified.

Object Detection Accuracy Results:

The dual YOLOv8 architecture implementation described in Section IV-C maintained the following detection performance metrics:

– Mobile phone identification: 84% accuracy with 4% false positive rate

– Smartwatch detection: 81.6% accuracy with 3% false positive rate

These results validate the effectiveness of the 0.65 confidence threshold established during model training, with real-world performance showing expected degradation from controlled training conditions due to environmental variability and detection complexity.

C. Hardware Performance Analysis

Real-time processing benchmarks conducted on the target hardware configuration (Intel i5 7th generation, 8GB RAM) demonstrate

- MediaPipe Face Mesh processing: 35% of computational load
- YOLOv8 object detection: 40% of computational load
- UI rendering and system coordination: 25% of computational load
- Memory usage remains stable throughout extended monitoring sessions

VI. Discussion and Comparison

This offline CPU-based design eliminates data privacy concerns and cost. While [3] use YOLO for cheating detection, their approach relies primarily on object detection and does not integrate advanced gaze tracking or facial behavior analysis. In contrast, our system combines YOLO-based object detection with MediaPipe Face Mesh [17] and Kalman filtering [15], [16], resulting in more robust detection of both unauthorized devices and suspicious gaze patterns.

Other recent reviews, such as [1], highlight that many AI-based anti-plagiarism solutions depend on cloud infrastructure or require high-performance hardware, which can introduce privacy risks and increase operational costs. Our solution addresses these limitations by operating fully offline on standard CPUs, ensuring data remains local and reducing barriers to adoption for institutions with limited resources.

Additionally, [2] demonstrates the use of LSTM models for behavioral analysis, focusing on temporal patterns in student activity. While effective for certain scenarios, such approaches may require more computational resources and do not always provide real-time feedback. Our architecture achieves real-time performance and high accuracy for both gaze and object detection, with 92.4% overall gaze accuracy and 84% smartphone detection, all while maintaining low CPU and memory usage.

The system's architecture enables real-time processing on CPU-based hardware, making it accessible to educational institutions with limited computational resources. Performance analysis reveals detection accuracies competitive with commercial solutions while providing complete data privacy through local processing.

VII. Conclusions

This research presents a comprehensive anti-plagiarism monitoring system that addresses critical limitations in existing examination oversight approaches through advanced computer vision and machine learning integration. The modular architecture successfully combines MediaPipe Face Mesh 468-point facial landmark technology with dual YOLOv8 object detection models, achieving 92.4% overall gaze accuracy and 84% smartphone detection rates while maintaining real-time performance on standard hardware.

The system's key contributions include: (1) sophisticated behavioral analysis through MediaPipe Face Mesh advanced facial landmark detection that provides superior accuracy compared to traditional 68-point methods, (2) mathematical gaze tracking algorithms with Kalman filtering that distinguish natural movements from suspicious activities, (3) specialized object detection framework optimized for examination environments, and (4) cross-platform compatibility enabling deployment across diverse institutional computing infrastructures without GPU requirements.

Performance validation demonstrates significant advantages over commercial solutions, including 90% cost reduction through local processing, complete privacy control through on-premise data handling, and sustained 3+ hour operation with 65-75% CPU utilization on Intel i5 systems. The comprehensive experimental validation on 1,600 test images from 14 participants confirms the system's effectiveness across all gaze directions, with particularly strong performance in challenging downward gaze detection (96.4% accuracy for detected faces).

Future development directions include expanding object detection capabilities to additional unauthorized devices, implementing advanced behavioral pattern recognition for detecting collaborative cheating scenarios and integrating natural language processing for audio-based violation detection.

The implementation successfully bridges the gap between advanced computer vision research and practical educational applications, providing institutions with accessible, cost-effective monitoring capabilities that maintain academic integrity standards.

References

[1] M. Abbas and S. Hameed, "A systematic review of deep learning based online exam proctoring systems for abnormal student behaviour detection," Int. J. Sci. Res. Sci. Eng. Technol., vol. 9, no. 4, pp. 192–209, 2022.

[2] W. Alsabhan, "Student cheating detection in higher education by implementing machine learning and LSTM techniques," Sensors, vol. 23, no. 8, Art. 4149, 2023.

[3] S. Kaddoura and A. Gumaei, "Towards effective and efficient online exam systems using deep learning-based cheating detection approach," Intelligent Systems with Applications, vol. 16, Art. 200153, 2022.

[4] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A systematic review on AI-based proctoring systems: Past, present and future," Education and Information Technologies, vol. 26, no. 5, pp. 6421–6445, 2021.

[5] I. Ahmad, F. AlQurashi, E. Abozinadah, and R. Mehmood, "A novel deep learning-based online proctoring system using face recognition, eye blinking, and object detection techniques," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 10, 2021.

[6] F. Kamalov, H. Sulieman, and D. Santandreu Calonge, "Machine learning based approach to exam cheating detection," PLoS ONE, vol. 16, no. 8, e0254340, 2021.

[7] L. Chen, X. Wang, and S. Zhang, "Deep learning approaches for educational fraud detection in online learning environments," IEEE Trans. Learn. Technol., vol. 16, no. 4, pp. 512–527, 2023.

[8] L. Zhao et al., "Academic dishonesty and its relations to peer cheating and culture: A meta-analysis," Educational Research Review, vol. 36, 100455, 2022.

[9] P. M. Newton and K. Essex, "How common is cheating in online exams and did it increase during COVID-19? A systematic review," J. Acad. Ethics, 2024.

[10] F. Noorbehbahani, A. Mohammadi, and M. Aminazadeh, "A systematic review of research on cheating in online exams from 2010 to 2021," Education and Information Technologies, vol. 27, pp. 8413–8460, 2022.

[11] A. Reedy, D. Pfitzner, L. Rook, and L. Ellis, "Responding to the Covid-19 emergency: Student and academic staff perceptions of academic integrity," Int. J. for Educational Integrity, vol. 17, 9, 2021.

[12] H. Lee et al., "Real-Time Markerless Facial Landmark Detection Using Deep Learning," IEEE, 2021.

[13] A. Sindel, A. Maier, and V. Christlein, "ArtFacePoints: High-Resolution Facial Landmark Detection in Paintings and Prints," ECCV 2022 Workshops, LNCS, vol. 13801, 2022.

[14] H. Aung, A. Valentinovich Bobkov, and B. Aye, "Real-Time Face Tracking Based on the Kalman Filter," in *2022 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, pp. 842–846, 2022.

[15] J.-M. Jeong, T.-S. Yoon, and J.-B. Park, "Kalman filter based multiple objects detection-tracking algorithm robust to occlusion," in *2014 Proceedings of the SICE Annual Conference (SICE)*, pp. 941–946, 2014.

[16] C. Li, Q. Xing, and Z. Ma, "HKSiamFC: Visual-Tracking Framework Using Prior Information Provided by Staple and Kalman Filter," Sensors, vol. 20, 2137, 2020.

[17] S. A. Jakhete and N. Kulkarni, "A Comprehensive Survey and Evaluation of MediaPipe Face Mesh for Human Emotion Recognition," in *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–8, 2024.

[18] H. Yi, B. Liu, B. Zhao, and E. Liu, "Small object detection algorithm based on improved YOLOv8 for remote sensing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, pp. 1734–1747, 2023.

[19] Honorlock Inc., "AI-powered online proctoring: Technical specifications and detection capabilities," Technical Documentation, Version 3.2, 2023.

[20] A. A. Akinyelu and P. Blignaut, "Convolutional Neural Network-based Calibration-free Technique for Improved Gaze Estimation," *Frontiers in Artificial Intelligence*, vol. 4, 796825, 2021.

[21] H. Jarodzka, I. Skuballa, and H. Gruber, "Eye-Tracking in Educational Practice: Investigating Visual Perception Underlying Teaching and Learning," Educational Psychology Review, vol. 33, no. 1, pp. 1-10, 2021.

[22] Frontiers in Robotics and AI research team, "Real-Time Webcam-Based Gaze Estimation for Computer Screens," Frontiers in Robotics and AI, 2024.

[23] A. A. Akinyelu and P. Blignaut, "Convolutional Neural Network-based Calibration-free Technique for Improved Gaze Estimation," Frontiers in Artificial Intelligence, vol. 4, 796825, 2021.