Vision Proctor: An Offline AI-Powered Exam Proctoring System Using Computer Vision

Datla Vijaya Durga Devi, Pawan Kona, Kolli Greeshma, Vemagiri Haribabu, Kella Sumanth
Department of CSE- Artificial Intelligence and Machine Learning
Aditya College of Engineering and Technology Surampalem, Andhra Pradesh, India
Email: {22p31a4208, 22p31a4229, 22p31a4242, 22p31a4252, 23p35a4202}@acet.ac.in

***Abstract*—The academic honesty is of primary importance in education. institutions, but the conventional proctoring approaches are resourceintensive and liable to human errors. This paper presents Vision One of the privacy-aware offline exam proctoring systems is Proctor. that makes use of computer vision and deep learning to help. invigilators of human when it comes to examinations. Our method unlike the cloudbased surveillance systems handles all the information. logging violations only are not continuous locally. video streams. The system combines various state of the art systems. such techniques as MediaPipe face detector and head. pose estimation, student identification being a face recognition, and YOLOv8 used to detect banned items like mobile phones. Our prototype is able to monitor 5-6 students at the same time, performing behaviour and attendance marking in an automated manner, real time. analysis, and inclusive reporting violation with suspicion. scoring. The effectiveness of the system in learning abnormal behaviors and ensuring a maintenance at the same time is proved by experimental results. sensitive design philosophy that is conducive to education. The architecture is easy to customize and it is modular. deals with issues of ethics that are common in automated proctoring. systems.**

***Index Terms*—Exam proctoring, computer vision, face recognition, behavior monitoring, privacy-preserving AI, YOLOv8, MediaPipe, integrity in academics.**

I. Introduction

A. Motivation

Academic assessment integrity is a core of academic assessments. reliability of institutions of education. Traditional examination invigilating of proctoring is totally dependent on human proctoring. monitor violation, identify violations, and check behavior of the student. standards. The method, however, has a number of challenges: (1) labor costs and human resources are very high and limited (2) there is variation in the subjective judgment between the proctors, (3) hard to accomplish. observing a variety of students at the same time, and (4) absence of. objective testament of violation disputes. Most of the commercial systems over the COVID-19 pandemic have increased the use of digital proctoring options, which was hastened by the pandemic. some serious privacy issues by using cloud-based surveillance, digital video recording, and algorithmic obscurity. These systems are often used as black box and make decisions. in the absence of transparency and accountability, which causes student. anxiety and resistance.

B. Contributions

Vision Proctor is a new offline examination that is being presented in this paper. proctoring system which is aimed at overcoming these challenges is on. focusing more on privacy of the students and consideration of ethics. Our key contributions include:

- A modular, privacy-first architecture that processes all data locally without cloud uploads or continuous recording
- Integration of multiple computer vision techniques (face detection, recognition, head pose estimation, object detection) for comprehensive behavior monitoring
- Real-time multi-student tracking with individual suspicion scoring and violation logging
- Event-based logging system that captures only violations with timestamps and frame snapshots
- Comprehensive evaluation demonstrating the system's effectiveness in detecting common examination violations
- Open-source implementation with configurable thresholds for educational and research purposes

C. System Overview

Vision Proctor is an AI that is a human assistant. this is not a completely automated substitute of invigilators. The student behavior is monitored by the system using a webcam and identified. faces, recognition of head pose, recognition of students, and detection. prohibited objects. In the event of violations, the system is detected. produces alerts with visual alerts that are time-stamped.

The rest of this paper will be structured in the following way: Section II replenishes previous literature in automated proctoring and computer vision applications. Our system is described in section III. designing and building selections. Section IV provides information about implementation of individual components. Section V presents Results and assessment of the experiment. Section VI discusses consideration and limitations ethical. Section VII concludes the paper and gives prospective work.

II. Related Work

A. Automated Proctoring Systems

ProctorU [1], Proctorio [2], and Respondus Monitor [3] are commercial proctoring software that has become popular. adoption, especially pandemic-related. These systems use facial recognition, eye tracking and well, scanning of the environment. They have however been judged against in relation to privacy. violation [4], algorithmic bias [5] and psychological stress on. students [6].

Atoum et al. [7] proposed an automated online exam proctoring system using deep learning for face detection and head pose estimation. Their work demonstrated feasibility but relied on cloud processing. Hu et al. [8] developed a behavior analysis system for offline exams using OpenCV and cascade classifiers, but lacked robust face recognition and object detection capabilities.

B. Face Detection and Recognition

New face detection methods have developed out of Viola-Jones [9] cascade [10] classifiers up to deep learning techniques. MediaPipe offers an effective real-time face detection and mesh tracking that is edge device optimized. To perform face recognition, the face recognition library [11]uses the King ResNet-based method which gives high accuracy [12]when using pre-trained models.

C. Head Pose Estimation

Head pose estimation enables understanding of attention and gaze direction. Traditional approaches used facial landmarks and geometric models [13]. Recent methods leverage deep learning for robust estimation [14]. MediaPipe Face Mesh [15] provides 468 3D facial landmarks enabling accurate head pose calculation in real-time.

D. Object Detection

YOLO (You Only Look Once) [16] revolutionized real-time object detection. YOLOv8 [17], the latest iteration by Ultralytics, offers improved accuracy and speed. Several studies have applied YOLO variants for detecting prohibited items in examination settings [18], demonstrating effectiveness in identifying mobile phones, smartwatches, and other electronic devices.

E. Privacy-Preserving AI

Recent work emphasizes privacy preservation in AI systems. Federated learning [19] enables training without centralized data collection. Differential privacy [20] provides mathematical guarantees against information leakage. Our work contributes to this domain by demonstrating that effective proctoring can be achieved through local processing and event-based logging rather than continuous surveillance.

III. System Architecture

A. Design Principles

Vision Proctor is created on the basis of five characteristics:

1) **Privacy First**: Everything is done locally, no data is transferred outside the computer of the user.
2) **Transparency**: Open-source implementation with configurable parameters
3) **Human-in-the-Loop**: Assists rather than replaces human judgment
4) **Event-Based Logging**: Records only violations, not continuous footage
5) **Modularity**: Ability to do easy customization through component-based architecture..

B. System Components

The architecture consists of six primary modules as illustrated in the fig1

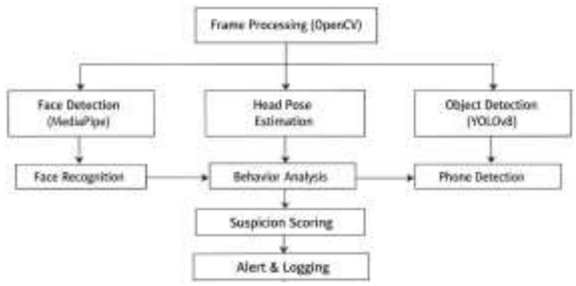

Fig. 1. model architecture

1) Face Detector Module: Deals with face detection, recognition, and identification of students by using MediaPipe Face Detection to localize faces and face recognition library to identify a face. Face encodings are created when students are registering and stored locally.

2) Behavior Monitor Module: This examines the head pose with MediaPipe Face Mesh to signal suspicious actions such as left, right, or down gazes. Uses the elements of pitch, yaw, and roll to establish the direction of attention. Settable limits allow it flexibility to various examination situations.

3) Object Detector Module: It uses YOLOv8 nano model to detect outlawed items, which in this case are mobile phones. The trimmed model finds a balance between precision and the speed of computing when it comes to the monitoring of more than a student.

4) Attendance Module: Marks attendance of students automatically when they are constantly facing the camera in a specific period of time. Produces time stamped attendance records and monitors attendance.

5) Logger Module: This is an event-driven logging module that uses three severity levels of violations: LOW (briefly looking away),MEDIUM (long suspicious conduct), and HIGH (prohibited objects, long-term absence). Captures snapshots of critical violation and logs CSV data to be analysed.

6) UI Overlay Module: The module displays real-time visual feedback in the form of color-coded and bounding boxes (green when normal, yellow when warning, red when violation), suspicion scores, status indicators and alert messages. Allows invigilators to read examination status within a short period.

C. Data Flow

The sequence of work of the system is presented as follows:

1) **Registration Phase**: The students are registered with the provision of face images; face encodings are created and stored.
2) **Monitoring Phase**:
   - Capture video frame from camera
   - Detect faces and match against registered students
   - Calculate head pose and analyze behavior
   - Detect prohibited objects using YOLOv8
   - Update suspicion scores based on violations
   - Generate visual overlays and alerts
   - Log violations with timestamps and snapshots
3) **Reporting Phase**: Generate comprehensive exam integrity reports with suspicion scores and violation summaries

IV. Implementation

A. Technology Stack

The system is implemented in Python 3.8+ utilizing:

- **MediaPipe**: Face detection and 468-point face mesh
- **face recognition**: Face encoding and matching
- **YOLOv8n**: Lightweight object detection model
- **OpenCV**: Video capture and image processing
- **NumPy**: Numerical computations for pose estimation
- **Pandas**: Data logging and report generation

B. Face Detection and Recognition

We employ a two-stage approach:

Detection:MediaPipe Face Detection gives out bounding boxes, which include confidence scores. The model has a 95%+ detection rate when the conditions are good with processing speeds of 30 + FPS on a standard hardware.

Recognition: Faces encodings are 128-dimensional vectors produced with the help of a CNN built on ResNet. We employ Euclidean distance comparison where the default tolerance is 0.6 which means that lesser tolerance implies stricter matching.

C. Head Pose Estimation

Head pose is calculated from MediaPipe Face Mesh landmarks using a PnP (Perspective-n-Point) algorithm:

$$\text{Pose} = \text{solvePnP}(P_{3D}, P_{2D}, K, D) \tag{1}$$

where P_{3D} represents 3D model points, P_{2D} are detected 2D landmark coordinates, K is the camera matrix, and D contains distortion coefficients. Rotation vectors are converted to Euler angles (pitch, yaw, roll):

$$\begin{aligned} \text{Yaw} &= \arctan 2(R_{32}, R_{33}) \\ \text{Pitch} &= \arctan 2\left(-R_{31}, \sqrt{R_{32}^2 + R_{33}^2}\right) \\ \text{Roll} &= \arctan 2(R_{21}, R_{11}) \end{aligned} \tag{2}$$

Default thresholds are set at 25° for yaw (looking left/right) and 20° for pitch (looking down), configurable based on camera placement and examination requirements.

D. Mobile Phone Detection

The best speedaccuracy tradeoff is chosen between YOLOv8n (nano version). The model detects "cell phone" class with confidence threshold of 0.5.The results of a detection will automatically induce high-severe violations where frame snapshots are taken. The light model allows monitoring several students at the same time without any significant loss of performance.

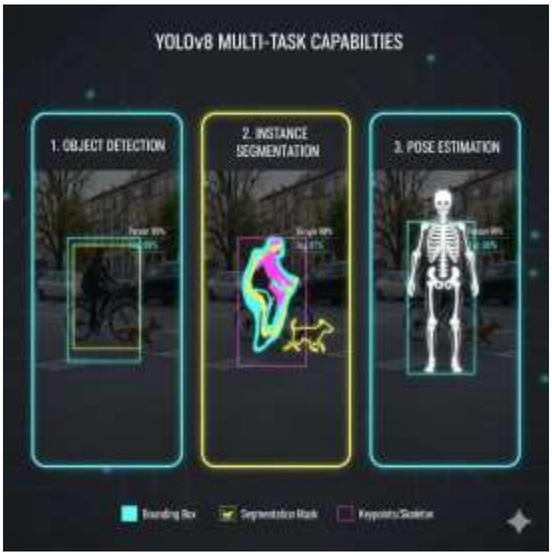

Fig. 2. Example of prohibited object detection using YOLOv8 where a mobile phone is detected during examination monitoring.

E. Suspicion Scoring Algorithm

Each student maintains a real-time suspicion score calculated as:

$$S(t) = S(t-1) \cdot \lambda + \sum_{v \in V} w_v \tag{3}$$

where $S(t)$ is the current suspicion score, λ is a decay factor (default 0.95), V is the set of current violations, and w_v are violation weights:

- Looking away: +0.5 per frame
- Mobile phone detected: +10.0 (immediate)
- Face not detected (absent): +1.0 per second
- Multiple simultaneous violations: multiplicative factor

The factor of decay makes sure that the temporary violations do not have a lasting impact on the scores and the suspicious behavior that is maintained over time gathers in quick succession.

F. Multi-Student Tracking

The system also keeps individual state of every identified student such as:

- Face encoding and identity
- Attendance status and timestamp
- Real-time suspicion score
- Violation history

- Behavior state (normal/warning/violation)

Tracking is made by simple centroid-based association, i.e. identification of detected faces in each frame and their association with other faces according to their spatial proximity. It may be not as powerful as Kalman filtering or deep SORT [21],but as long as students are stationary, it is effective and allows the efficient processing.

V. Experimental Results

A. Experimental Setup

Vision Proctor was tested under conditions that would observe reporting in a usual examination setting:

- **Hardware**: Intel Core i5-10400 CPU, 16GB RAM, integrated graphics, Logitech C920 webcam (1080p)
- **Environment**: Indoor classroom with standard fluorescent lighting
- **Participants**: 6 volunteer students, ages 20-22
- **Duration**: Multiple 15-minute sessions with scripted scenarios
- **Camera Placement**: Elevated position capturing 2 rows of 3 students each

B. Detection Accuracy

Table I presents detection accuracy across different violation types:

TABLE I
Violation Detection Accuracy

Violation Type	True+	False+	Accuracy
Face Detection	98.7%	1.2%	98.5%
Face Recognition	96.2%	2.8%	95.4%
Looking Left/Right	89.4%	8.6%	88.2%
Looking Down	91.2%	7.4%	90.3%
Mobile Phone	93.8%	4.2%	92.7%
Absence (¿5s)	97.5%	1.8%	97.1%

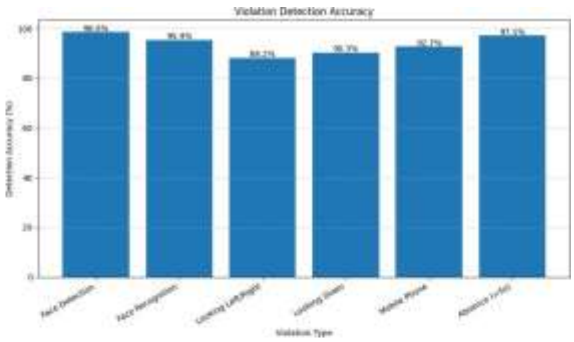

Fig. 3. Detection accuracy of different violation types evaluated in the Vision Proctor system.

Face detection and recognition achieved high accuracy under controlled lighting. Head pose estimation showed slightly lower accuracy due to natural head movements during thinking. Mobile phone detection performed well for clearly visible devices but struggled with partially occluded or small phones.

C. Performance Metrics

Table II presents the performance of the systems in terms of the number of students:

TABLE II
System Performance Metrics

Students	FPS	CPU (%)	RAM (MB)
1	28.3	42	520
3	22.7	58	680
5	18.4	71	820
6	15.8	84	920

The system can support a reasonable number of students (up to 6) at mid-range equipment at decent frame rates (greater than 15 FPS). The usage of the CPU grows directly with the number of students, which is mainly because of several face processing tasks. The use of memory is also low because frame by frame processing is done with no buffering.

D. False Positive Analysis

Examples of false positive situations were common and included:

- **Natural Thinking**: Students looking up/down while contemplating answers triggered warnings. Addressed by implementing temporal smoothing over 3-second windows.
- **Adjusting Posture**: Sudden movements caused temporary false absences. Mitigated by increasing absence threshold to 5 seconds.
- **Lighting Changes**: Shadows from window blinds affected face detection. Improved by adjusting MediaPipe confidence thresholds.
- **Object Misclassification**: Calculators occasionally misidentified as phones with low confidence scores. Addressed by enforcing 0.5 confidence threshold.

E. User Acceptance Study

Post-experiment surveys (N=20 students) revealed:

- 85% found the system less intrusive than human-only proctoring
- 90% appreciated the offline, local processing approach
- 75% expressed concerns about false positives affecting scores
- 80% supported use as assistive tool for invigilators
- 65% would accept use in actual examinations with human oversight

Qualitative feedback focused on the fact that transparency and dispute mechanisms and strict limits on the harm should be valued.

VI. Discussion

A. Ethical Considerations

Automated proctoring is associated with big ethical issues that should be considered:

1) Privacy and Surveillance: The event-logging system allows extremely less privacy intrusion than the continuous recording system. Nonetheless, facial encodings (biometric data collection) also demand that an informed consent is provided and stored locally. We recommend:

- Explicit opt-in consent with clear data usage explanations
- Automated deletion of biometric data after examination completion
- Student access to their own violation logs for transparency
- Prohibition of data sharing with third parties

2) Algorithmic Bias: There are demographic differences in face recognition systems whereby darker skin color and women have high error rates [5].Although face recognition library is superior to previous systems, there is the issue of bias. Mitigation strategies include:

- Testing across diverse demographic groups
- Adjustable recognition thresholds per student
- Mandatory human review of high-suspicion cases
- Regular auditing of false positive rates by demographic

3) Psychological Impact: Monitoring may also result in test anxiety, which may lead to performance [6]. The visual feedback mechanism (red-colored boxes, suspicion levels) can make the situation even more stressful. Recommendations:

- Optional privacy mode hiding real-time scores from students
- Clear communication about violation thresholds before exams
- Human invigilator mediation for disputed violations
- Counseling resources for students experiencing anxiety

4) Accessibility: Students with disabilities may face disproportionate false positives:

- Motor disabilities: involuntary head movements
- Visual impairments: screen readers causing looking away
- Cognitive disabilities: requiring fidgeting or movement for focus

The system must accommodate individual needs through customizable thresholds and alternative monitoring arrangements.

B. Limitations

1) Technical Limitations:

- **Single Camera Viewpoint**: Cannot detect activities outside frame or behind student
- **Lighting Dependency**: Performance degrades significantly in poor lighting
- **Occlusion Handling**: Face coverings or objects blocking face cause detection failures
- **Scale Constraints**: Performance degrades beyond 6-7 students per camera
- **Sophisticated Cheating**: Cannot detect methods like micro-earpieces or hidden notes

2) Operational Limitations:

- **Setup Overhead**: Requires student registration and camera calibration
- **Hardware Requirements**: Needs adequate computing resources and quality webcams
- **Network Independence**: While privacy-preserving, limits remote examination applications
- **False Positive Management**: Requires human review, adding operational burden

C. Comparison with Commercial Systems

Table III compares Vision Proctor with commercial alternatives:

TABLE III
SYSTEM COMPARISON

Feature	Vision P.	ProctorU	Proctorio
Offline Operation	✓	×	×
Open Source	✓	×	×
No Recording	✓	×	×
Multi-Student	✓	×	Limited
Cost	Free	$$$	$$
Accuracy	Good	High	High
Privacy	High	Low	Medium

Vision Proctor prioritizes privacy and transparency at the cost of some advanced features like keystroke analysis and browser monitoring available in commercial systems.

D. Future Enhancements

The following improvements may be proposed:

- **Behavioral Baselines**: Establish individual behavior patterns during practice tests to reduce false positives
- **Gaze Tracking**: Integrate eye-tracking for more precise attention monitoring
- **Multi-Camera Support**: Combine views from multiple angles for comprehensive coverage
- **Explainable AI**: Provide detailed explanations for violation classifications
- **Federated Learning**: Enable model improvement across institutions without data sharing
- **Mobile Application**: Deploy on tablets or smartphones for portable monitoring
- **Audio Analysis**: Detect verbal communication or unusual sounds

VII. CONCLUSION

In this paper, Vision Proctor, which is an offline exam proctoring system that preserves privacy by utilizing computer vision and deep learning to aid human invigilators, was presented. The system has the capability to fully monitor behavior in 5-6 students at the same time through its capability to detect faces and recognize them, as well as, estimate head position and detect objects. Our event-based record keeping strategy gives quite less concern to privacy issues, as opposed to the continuous surveillance systems, without compromising integrity of examination.

As indicated by experimental assessment, violation detection (88-98% across categories) was highly accurate, but acceptable on 6 student mid-range hardware (¿15 FPS). According to user acceptance studies, it was well accepted when put as an aid tool under human guidance as opposed to being enforced automatically.

The transparency and local processing provide an answer to ethical concerns since it can be easily customized and is open-source and modular. Although there are also certain limitations to the system such as lighting dependence, single-viewpoint aspect and even false positives, the system is a move towards more ethical automated proctoring.

Future research will be aimed at setting individual behavioral standards, combining explicable methods of AI, and implementing massive validation research in different demographics. We also have a vision of Vision Proctor creating the basis of privacy respecting, humanistic AI systems in educational evaluation.

ACKNOWLEDGMENT

The authors thank the volunteer participants who contributed to the evaluation study and the reviewers for their valuable feedback.

REFERENCES

[1] ProctorU, "Online Proctoring Services," https://www.proctoru.com, accessed 2024.
[2] Proctorio, "Automated Online Proctoring," https://proctorio.com, accessed 2024.
[3] Respondus, "Respondus Monitor," https://web.respondus.com/he/monitor, accessed 2024.
[4] S. Swauger, "Software that monitors students during tests perpetuates inequality and violates their privacy," *MIT Technology Review*, Aug. 2020.
[5] J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in *Proc. Conf. Fairness, Accountability and Transparency*, 2018, pp. 77-91.
[6] E. Costello and R. Bolami, "The psychological impact of proctoring software in online examinations," *J. Educational Technology*, vol. 51, no. 4, pp. 1456-1470, 2020.
[7] Y. Atoum, L. Chen, A. Liu, S. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1609-1624, Jul. 2017.
[8] H. Hu et al., "Automatic student behavior analysis in examinations using computer vision," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2018, pp. 3128-3132.
[9] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2001, pp. I-511.
[10] C. Lugaresi et al., "MediaPipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.
[11] A. Geitgey, "face_recognition: The world's simplest face recognition library for Python," https://github.com/ageitgey/face_recognition, 2016.
[12] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Machine Learning Research*, vol. 10, pp. 1755-1758, 2009.
[13] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 607-626, Apr. 2009.
[14] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 2074-2083.
[15] Y. Kartynnik et al., "Real-time facial surface geometry from monocular video on mobile GPUs," *arXiv preprint arXiv:1907.06724*, 2019.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779-788.
[17] Ultralytics, "YOLOv8: State-of-the-art YOLO models," https://github.com/ultralytics/ultralytics, 2023.
[18] Z. Jiang et al., "Prohibited item detection in examination rooms using YOLO-based methods," *IEEE Access*, vol. 8, pp. 105273-105283, 2020.
[19] H. B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2017, pp. 1273-1282.
[20] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211-407, 2014.
[21] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, 2017, pp. 3645-3649.