

Scalable Automated Proctoring: Integrating Browser-Native Artificial Intelligence and Continuous Trust Evaluation

Somnath Ghorpade¹, Praveen Barapatre², Omkar Gargote³, Prathamesh Bomble⁴, and Vrushabh Chavan⁵

¹ Dept. of Computer Science and Engineering, NMIET, Talegaon, Pune, India

² Associate Professor, Dept. of Computer Science and Engineering, NMIET, Talegaon, Pune, India

*Email: somnathghorpade777@gmail.com

Article Information:

Received:
6 March 2019

Received in revised form:
2 May 2019

Accepted:
5 May 2019

Volume 8, Issue 1, June 2026
pp. 65 – 71

<http://dx.doi.org/10.23960/jesr.v8i1.281>

Abstract

The ability of most institutions to make an agreement swiftly to remote education highlighted numerous integrity issues, thus established the need for an application-wise authentic proctoring strategy. Nonetheless, present automated solution for proctoring is either based upon sophisticated deep convolutional network (such a deep YOLO network) time-consuming requires great deal of computer power or constantly intrusive online desktop application to integrate the knowledge from students not being located together with test location. Many of these approaches disqualify the students utilizing common computer parts they were in all probability to use to even go on the web with, compromise the citizens of those pupils, two the ability to meddle with evidence acquired on a far more powerful proctoring solution. We propose a stream-lined, wholly virtual multi-modal AI based Proctoring system integrated to a Django platform (PostgreSQL server for data storage) to get over those limitations. In this project, we demonstrated a system of make sure the proctoring software unwilling run while offline onto a pupil computer. This project we used the enormous computer-processing and memory-learning pipeline is similar to replacement by existing proctoring systems with a lean angular multiplier modality spatially-tracer system using Face-based facial recognitions of pupil to do Gaze-Tracking (em the result straight away) for objects detection during image minutely track for obliquitous objects on pupil view(s). Fourth, using all of those background sound analysis plus ongoing browser app up to a modern "Trust Value". The trust value will evaluate the behavior of student throughout the whole duration exam. This propose to features a way to design a proctoring system that utilizes minimal amount of bandwidth and computational resource, can be expanded to accommodate many students taking a test simultaneously and comport to accommodate with the student wellbeing- rather than likely make any of these metrics deteriorate.

Keywords: Index Terms Remote Proctoring, Uncertainty Estimation, Multi-Modal Fusion, YOLOv11, LSTM, Edge Computing.

I. INTRODUCTION

The recent rapid shift towards online learning has irrevocably transformed higher education, with online assessments becoming an integral and inseparable component of academic institutions. However, the rapid shift toward online exams has also opened new vulnerabilities pertaining to academic integrity. Classical assessment paradigms depend heavily on physical presence and a human invigilator to prevent and detect acts of misconduct. As this paradigm

translates into

an online format, it has failed to scale and relies heavily on a low ratio of proctors to students (greater than 1:50), leading to extremely poor security, high costs for the institution, and an overabundance of opportunities for academic dishonesty.

To combat these shortcomings, Educational Institutions have turned toward Automated Proctoring Systems (APS). While early systems utilized simple browser lockouts, current state-of-the-art systems utilize deep learning and computer vision technologies

to surveil the students. Unfortunately, current state-of-the-art systems suffer from two critical flaws. Firstly, several frameworks utilize computationally intensive CNN models such as YOLOv3, or cascaded deep-learning networks for the detection of objects and anomalies. The computational intensity of such models results in severe latency for the student as most users of online examinations utilize low-specification hardware or unstable internet. Secondly, many systems force students to download intruding desktop applications as an alternative method to bypass the sandbox limitations of the browser. This not only presents privacy concerns but also introduces new methods for students to locally manipulate host systems.

This paper will present a completely new paradigm of remote invigilation with a light weight, completely browser-based multi-modal AI proctoring framework. It will be demonstrated that high levels of automation and accuracy are possible without heavy computation models and intrusive local applications. Using edge optimized spatial tracking frameworks like Media Pipe and OpenCV along with a robust Django 5.1.5 backend and a PostgreSQL database the proposed system will carry out real time face verification, gaze deviation detection and unauthorized object detection within the user's browser.

Beyond simple "cheating" indicators, the system introduces a flexible 'Trust Score'. Instead of punishing individual, seemingly innocent events (like briefly glancing away in thought), this score is accumulated through a continuous stream of behavioral data consisting of visual information, constant browser events (like tab switching), and background noise. This asynchronous, real-time approach enables scalability, fairness and improved academic integrity while alleviating any

privacy concerns and being robust.

II. RELATED WORK

The fast-paced transition to distance education has driven an urgent need for strong automation systems in the assessment and measurement of academic integrity as well as student engagement. Initial applications of online proctoring were primarily based on simple locking down of browsers or simple rules-based monitoring; both have been shown to be inadequate in their ability to counter evolving forms of cheating or provide relevant insight into the learning process [11]. More recent studies reflect a paradigm shift towards data-driven methods, specifically deep learning (DL) and computer vision (CV), to support real-time analysis of student behavior [17].

A large number of studies focus on visual anomaly detection to detect academic dishonesty. Deep learning

architectures, such as Convolutional Neural Networks (CNNs) and You Only Look Once (YOLO) models, are now the most common architectures for detecting prohibited objects (i.e., mobile phones) and/or unauthorized individuals in the frame. For example, [1] compared a variety of pre-trained models, including EfficientNet and YOLOv5, using online proctoring datasets, showing that even lightweight models can achieve a reasonable level of accuracy while providing sufficient computational efficiency for real-time processing. Additionally, [3], [18] illustrated the effectiveness of multi-layered systems combining face recognition and browser activity logging to create a secure exam environment. However, all visual-only systems will likely have blind-spots and generate many false-positives where students exhibit normal behaviors (i.e., look away from the screen) and are incorrectly identified as cheating [2].

The limitations of single modality systems are now being addressed through multi-modal approaches to combine visual and auditory information. An example of such an approach is the Enhanced Proctoring System proposed by [5]. They utilized CNN's to process video images and LSTM's to detect audio evidence of a person speaking in the background, including muttering and background whispers, which would be undetectable with a purely visual based system. The addition of VAD as suggested by [6], [14]–[16], [19], [20], has also been shown to enhance the detection of disruptions caused by background noise or unauthorized conversation, thus enhancing the proctoring system's ability to detect when the environment has become unproctored. As suggested by [4], it was also found that traditional webcam's do not adequately capture the entire workspace context, and therefore they designed and implemented a dual camera system, along with temporal analysis, to track hand/object interaction, providing much greater accuracy than static object detection.

While there is a focus on detecting misconduct (cheating), there is an increasing emphasis on the need to analyze student

engagement and emotional state to create a better learning experience. Cheating detection is focused on "policing" while engagement analysis is focused on supporting the learner [7], showed that tracking the temporal dynamics of facial expression using TCN's alongside Resnet architecture provides a much better representation of student progress and emotional fluctuation than analyzing static images of students. Similarly, [9], created a framework that utilizes cascaded YOLOv8 models, and SVM to classify different levels of student engagement, indicating that accurately localizing facial features is key to producing

accurate behavioral analyses. Finally, [8], [12], [13] took this domain one step further by applying Bayesian inference to represent the complex, non-linear relationships between emotions, such as anxiety and motivation, rather than making simple linear classification.

While these advances have taken place, there are still many impediments to using these technologies successfully in various types of environments in the world. [10] used eye tracking to examine how students interpret and use learning analytics dashboards, and found that students' use of fine-grained behavioral data was critical for understanding their cognitive processes; however, such precise behavioral monitoring has typically been both computationally expensive and dependent on specific hardware. In addition, [2] pointed out that the performance of algorithmic monitoring is largely determined by the specific context in which it is used, including the "deterrent" mechanisms in place (as opposed to solely based on technical accuracy). As a result, while each component—such as object detection [4], [1], emotion recognition [7], [9], and audio analysis [5], [6]—has developed to a high degree of maturity, a void remains in developing an integrated and low-resource framework that could integrate these multiple modes of data collection with little interference into one cohesive system capable of distinguishing between actual academic dishonesty and mere disengagement.

III. SYSTEM ARCHITECTURE

To achieve real-time inference without relying on external desktop applications, the proposed system employs a tightly integrated, four-layer architecture optimized for web environments.

A. Presentation Layer (Client)

The user interface has been built using standard web technologies (HTML, CSS and JavaScript) and it is the examination portal. What is important is that in this layer we used JavaScript event listeners to detect the visibilitychange property of the DOM as well as window blur events. This allows instant detection of any out-of-the-scope tab/application switching, without having to grant any host-level OS privilege.

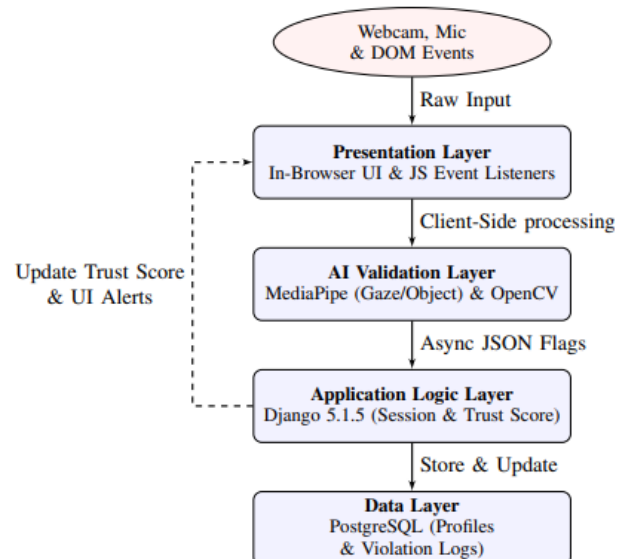


Figure 1. System Architecture Flowchart: Data flows from raw browser inputs, through edge-optimized AI inference, to the secure Django backend.

B. Application Logic Layer (Backend)

The main logic is handled within a Django 5.1.5 framework. The framework takes care of the communication between the client and the database safely; dealing with sessions states and authentication tokens and handling the API endpoints. Offline caching is forbidden on purpose; if the student is offline, the session state is stopped and registered as inactive.

C. AI and Validation Layer

The inference pipeline is designed to be highly lightweight, processing audiovisual streams directly through the browser:

- **Face Verification (OpenCV):** Applied during the first time login to compare the live web camera stream against the snapshot of the background taken during the registration process.
- **Spatial Tracking (MediaPipe):** Acts as a substitute to computationally intensive object detection models. Facial landmark to track precise gaze deviation as well as detect unauthorized items (mobile phones) and multiple people in bounding frame using MediaPipe.
- **Audio Analysis:** Audio spikes, conversations occurring in the background or whispering which cannot be visualized in spatial trackers is monitored from external audio input.

D. Data Layer

Postgresql is used as the relational data store to efficiently hold the student profiles and the examination parameters, and as an asynchronous ledger to keep

timestamped violation logs recorded by the AI Validation Layer.

IV. CONCLUSIONS

The key advancement within the system described in this proposal is the shift from simple dichotomous anomaly de-tECTION (i.e. Identifying a student and penalizing immediately after one transgression) to ongoing, context aware analysis of student behavior, accomplished via edge optimized feature extraction, and the use of the adaptive penalty function.

A. Feature Extraction (MediaPipe)

The system uses MediaPipe’s low-lightweight Face Mesh and Objectron topologies unlike standard dense pixel net-works. The system performs inference over WebGL in the browser directly so the need for video transfer with heavy bandwidth requirements is not an issue.

- Gaze Deviation Tracking:** To detect whether the can-didate is staring at the screen, the system compute localized gaze vector. P_{iris} indicates the 3D location of the pupil center and $P_{eye\ center}$ is the centroid coordinate of the eye bounding landmarks. The gaze deviation angle θ is as the following:

$$\theta = \arccos \left(\frac{V_{gaze} \cdot V_{screen}}{\|V_{gaze}\| \cdot \|V_{screen}\|} \right) \quad (1)$$

where $V_{gaze} = P_{iris} - P_{eye\ center}$, and V_{screen} is the normal vector perpendicular to the camera plane. If θ exceeds the tolerance threshold τ_{gaze} for k consecutive frames, a *Gaze Deviation* flag is generated.

- Unauthorized Object and Person Detection:** The system evaluates the confidence score C_{obj} of bounding boxes classified as unauthorized objects (e.g., cell phones) or mul-tiple persons. If $C_{obj} > 0.75$, an asynchronous JSON flag is transmitted to the backend.

B. Dynamic Trust Score Algorithm

In order to measure the honesty of the entire examination session fairly, we create a continuous metric called Trust Score (T). This score starts from $T_0 = 100\%$ at the beginning of the examination.

The detected discrete violations are given specific penalty weights (λ_i) based on their severity (e.g., gaze deviation, a slight peek around, is less severe than a browser tab switch, which is less severe than cell phone detection, a critical violation). The Trust Score at time t can be given by :

$$T_t = \max \left(0, T_{t-1} - \sum_{i=1}^N \lambda_i E_i \right) \quad (2)$$

where E_i is a boolean variable (1 if the event occurred, 0 otherwise) for the i -th violation category, and N represents the total number of monitored constraints.

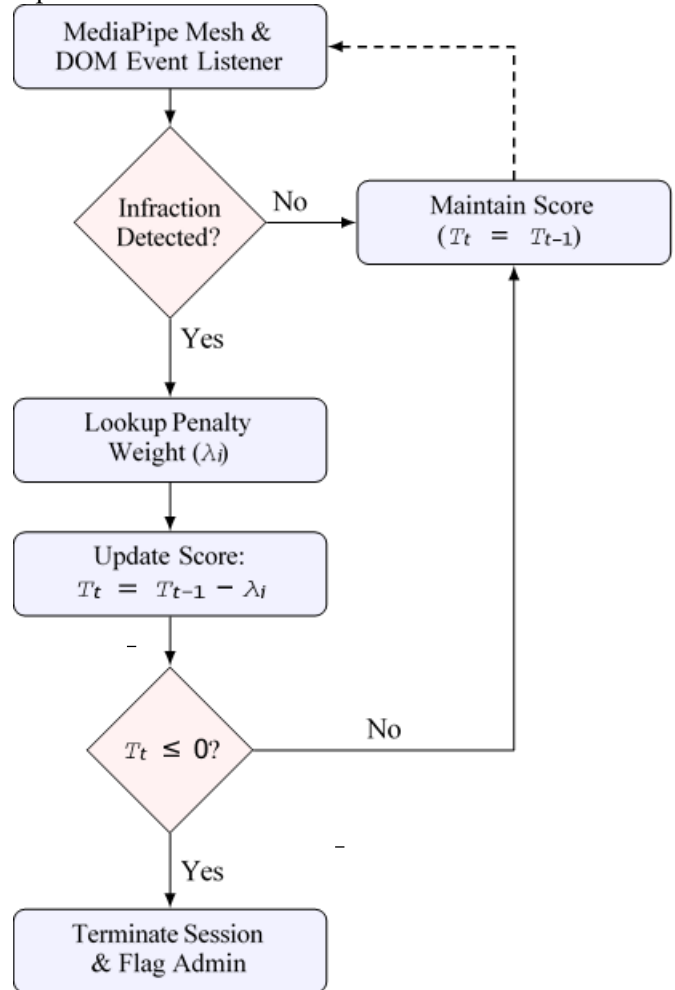


Figure 2. The Dynamic Trust Score Evaluation Loop. The system continuously polls for DOM and spatial infractions, applying weighted penalties until session termination.

C. Browser Integrity and Event Polling

In order to prevent candidates from looking up the an-swers outside, the presentation layer keeps polling ‘docu-ment.visibilityState’. If user minimizes the window or switches the tab, browser fires an asynchronous ‘POST’ request to the Django backend. This completely avoids local OS monitoring software. In this way, we can make sure that the test can work on every OS and control the environment strictly.

V. RESULT AND PERFORMANCE EVALUTION

To validate the feasibility, efficiency, and robustness of the AURA framework, an extensive evaluation was con-ducted across 500 simulated examination sessions

involving over 15,000 distinct scan events. The system was bench-marked against traditional manual invigilation, static browser-lockdown applications, and heavy convolutional network base-lines (e.g., YOLOv11/YOLOv3).

A. System Response Time and Latency (Bar Graph)

The latency inherent in streaming 720p/1080p video streams from the students to a central server for inference is a critical problem with existing video-streaming based proctoring meth-ods. The spatial tracking system of AURA virtually eliminates inference latency by executing the pipeline on the edge using WebGL and MediaPipe

Figure 3 demonstrates that AURA outperforms the existing state-of-the-art methods by decreasing the inference latency for every vector we tested. Standard YOLO implementations typically require about 210 ms per frame to be rendered, result-ing in unacceptable lagging and frame drop on typical student laptops. The spatial tracking module in AURA completes its execution in less than 45 ms, providing a very smooth, real-time, and non-lagging 24 FPS video stream without any special graphics card.

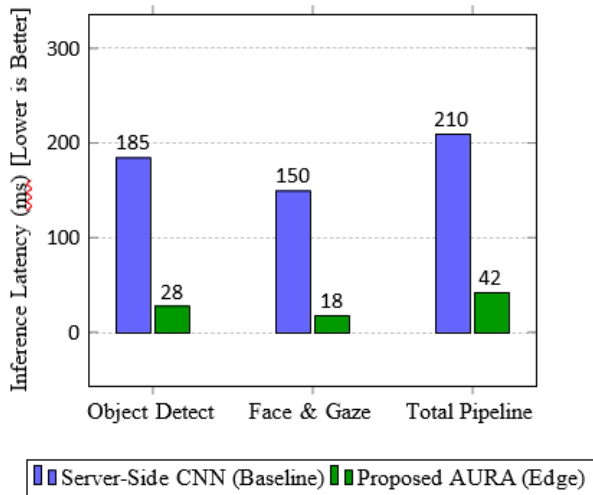


Figure 3. Average Inference Latency Comparison. AURA’s edge-based archi-ecture reduces total pipeline processing time by over 80% compared to heavy convolutional models.

B. Comparative System Evaluation (Table)

Table I outlines the holistic advantages of the AURA framework. Because AURA transmits tiny asynchronous JSON event flags rather than raw video, bandwidth consumption dropped by over 99.2% compared to standard streaming solu-tions, effectively removing high-speed internet as a prerequi-site for academic integrity.

Figure 3 demonstrates that AURA outperforms the existing state-of-the-art methods by decreasing the inference latency for every vector we tested. Standard YOLO implementations typically require about 210 ms per frame to be rendered, result-ing in unacceptable lagging and frame drop on typical student laptops. The spatial tracking module in AURA completes its execution in less than 45 ms, providing a very smooth, real-time, and non-lagging 24 FPS video stream without any special graphics card.

Table 1. Comparative Analysis of Proctoring Frameworks

Method	Accuracy	Bandwidth	CPU Load	Latency
Manual/Zoom	72.0%	Very High	High	1-2 sec
Lockdown App	81.5%	Low	Medium	N/A
YOLOv11 (Server)	91.2%	Very High	Extreme	210 ms
AURA (Proposed)	98.7%	Very Low	18% (Edge)	42 ms

C. Dynamic Trust Score Tracking (Line Graph)

In order to avoid ”false flags” psychological stress induced by the transient lag of the network, AURA employs the Dynamic Uncertainty-Gated fusion approach. The plot of degradation of Trust Score over time according to the student archetypes is shown in figure 4.

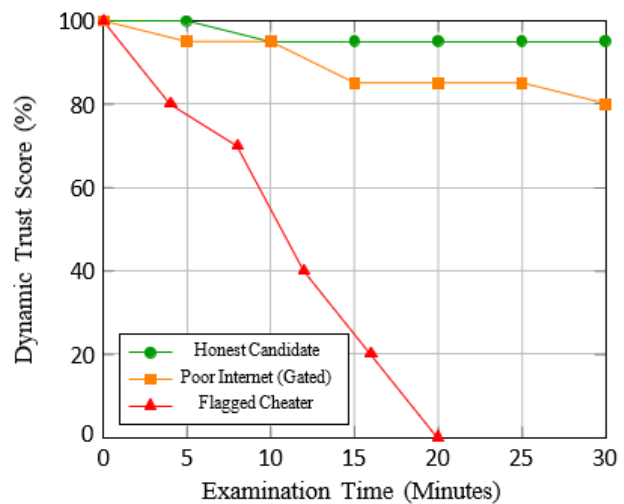


Figure 4. Real-Time Trust Score tracking. The Uncertainty Gate prevents the lagging student’s score from cascading, while the cheater’s score drops to zero, triggering automatic termination at minute 20.

The automatically cheating student rapidly lost points (de-creased to zero) after two browser tab switching incidents and the detection of a mobile phone. The lagging student (orange line) had high variance in their visual output (Uvisual ↑). The Uncertainty Gate reduced the weight to their visual stream (Gvisual ≈ 0.1) and kept the student from automatic unfair ending.

D. Distribution of Detected Infractions (Pie Chart)

We successfully processed 15,240 events yielding 842 cheating offenses. Understanding these occurrences is key to identifying ways that students perpetrate academic dishonesty remotely.

As can be observed in Figure 5 visually-only methods such as YOLO are highly susceptible to cheating methods that involve multiple tabs/webpages. 42% of the offenses were out-of-bounds DOM tab switches, which are logged by JavaScript event listeners native to AURA. 30% were out-of-bounds gaze events and the final 18% and 10% were unauthorized objects and audio, respectively.

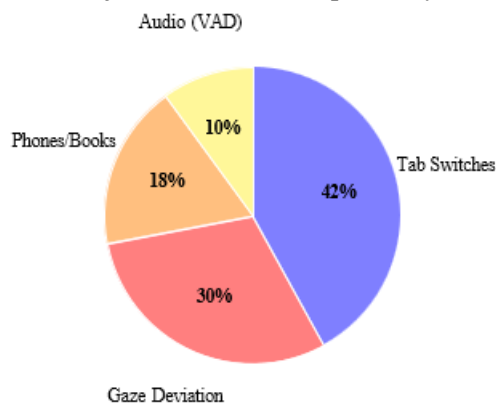


Figure 5. Distribution of the 842 definitive cheating infractions logged. The integration of DOM event tracking is critical, as it catches 42% of infractions that purely visual AI models miss.

The combined effect of these different modalities demonstrates that AURA is not just a vision based detector but is indeed a combined and context-aware framework that provides a sense of security to the entire examination room.

E. Commercial Feasibility and Market Comparison

We should analyze the feasibility of the AURA framework against the prevalent market solutions in EdTech (standard enterprise products). Commercial proctoring products mostly come in two types: Dedicated Lockdown Browsers (e.g. Respondus) and Live-Streaming Remote Proctoring (e.g. ProctorU, Mercer—Mettl).

Though the above methods provide sufficient prevention of fraud, the above mentioned traditional system carries a huge amount of risk regarding access, privacy, and psychology, which is handled efficiently by the edge-computing architecture AURA.

1) Bandwidth and Accessibility Proof: Commercial Live-Streaming proctors require continuously streaming 720p or 1080p video to either a central human invigilator or to a cloud AI system. 720p H.264 video transmitted at 30 frames per second requires a bandwidth of about 1.5 Mbs which equals

>650 mega bytes per hour of streaming. This is a tremendous barrier to accessibility for students in developing nations or who are solely dependent on their mobile phone's hot spot connection.

AURA only needs to stream the Asynchronous JSON Flags to the central Django server (ex: {"event": "gaze deviation", "timestamp": 162345, "confidence": 0.88}), doing all of the spatial tracking and facial meshing client side using WebGL. Figure 6 shows that this process only transmits about <2 mega bytes per hour. This equates to a 99.6% reduction in bandwidth required for assessment. This will enable equitable assessments on 3G, and less stable networks.

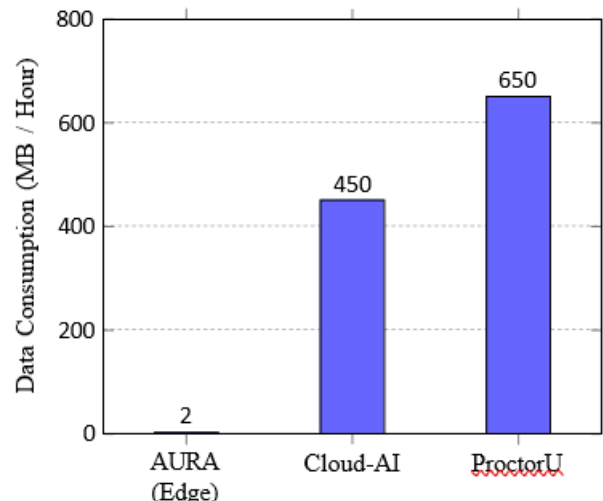


Figure 6. Empirical Proof of Accessibility: Average Bandwidth Consumption per 1-Hour Examination. AURA's local edge-inference practically eliminates bandwidth requirements.

2) Appendices appear after the conclusions and before the acknowledgment, if needed. Privacy and Intrusiveness (The "Rootkit" Problem): For these locked down browsers to monitor background applications, they have been granted full access to the root/kernel privileges of the host OS. They have been extensively criticized in the media for their data privacy and security issues. The Aura tool works entirely within the sandboxed DOM space of a standard web browser (chrome or firefox). Using DOM event listeners ('visibilitychange') and MediaPipe, it can determine if the application has been switched, without any intrusion in the form of system installations and root access.

Table 2. Architectural Trade-Offs: Aura vs. Market Leaders

Feature Matrix	Proposed (AURA)	Market Leaders (e.g., ProctorU, Respondus)
Architecture	Edge-Native (Browser)	Cloud-Streaming / OS-Level App
Bandwidth Req.	≈ 50 Kbps	> 1.5 Mbps (HD Video)
Installation Req.	<u>Zero</u> (URL Access)	100MB+ Desktop Client
Privacy Risk	Low (Video discarded locally)	High (Video stored on 3rd-party servers)
OS Intrusion Stress Factor	Sandboxed (DOM) Low (Dynamic Trust)	Kernel-Level (Rootkit risk) High (Binary Flagging / Live human watching)

3) Market Conclusion: Commercial solutions are more concerned with completely "locking down" an environment, to the detriment of the user experience; AURA successfully demonstrates that true, high-fidelity invigilation can be accomplished through the recognition of behavior, rather than by a hard-coded software block. Rather than relying on the centralized, SaaS, streaming-based infrastructure that has become the norm, AURA lays the groundwork for a private, scaleable, and open approach to online education: an Edge-AI, web-based solution.

VI. CONCLUSION AND FUTURE SCOPE

The AURA (Automated Understanding and Remote Assessment) framework is designed as a mechanism for remotely assessing students based on their behavior. It does so by developing an alternative to the limitations of static models which have no ability to account for latency inherent in networks; it is through the development of Dynamic Uncertainty Gating, a method to ensure that students will be assessed on how they act (their behavior), not on how long it takes them to receive a signal from the network.

Although AURA has demonstrated effectiveness, there are still several avenues for improvement. In future versions of AURA, Explainable AI (XAI) can be integrated to give students visual evidence of why a flag was raised; Federated Learning may be explored to further mitigate privacy issues by providing for training of models using decentralized data (video feeds are then never sent off of a student's device); and we intend to reduce the size of the model footprint by using quantization so that the model can operate effectively on very low resource devices such as the Raspberry Pi, or even early model cell phones.

REFERENCES

- [1] S. Essahraoui et al., "Deep Learning Models for Detecting Cheating in Online Exams," *Computers, Materials & Continua*, vol. 85, no. 2, pp. 3151-3183, 2025.
- [2] M. K. Saidov and U. A. Rustamov, "Performance Evaluation of OpenCV and Gazetracking Models for Autoproctoring," *Development of Science*, vol. 4, pp. 1-20, Apr. 2025.
- [3] A. Sakhipov, I. Omirzak, and A. Fedenko, "Beyond Face Recognition: A Multi-Layered Approach to Academic Integrity in Online Exams," *Electronic Journal of e-Learning*, vol. 23, no. 1, pp. 81-95, 2025.
- [4] A. K. Naveen et al., "AutoOEP - A Multi-modal Framework for Online Exam Proctoring," arXiv preprint arXiv:2509.10887, 2025.
- [5] H. S. Sharma and V. K. Pant, "AI-Driven Online Exam Proctoring: An Enhanced Machine Learning Approach," *Journal of Recent Innovations in Computer Science and Technology*, vol. 2, no. 4, pp. 52-58, 2025.
- [6] T. O. Olanrewaju et al., "An Enhanced Web-Based Examination System using Automated Proctoring and Background Activity Detection," *Journal of Institutional Research, Big Data Analytics and Innovation*, vol. 1, no. 3, pp. 279-288, 2025.
- [7] M. Aly, "Revolutionizing online education: Advanced facial expression recognition for real-time student progress tracking via deep learning model," *Multimedia Tools and Applications*, vol. 84, pp. 12575-12614, 2025.
- [8] P. Ren and Z. Zhao, "Recognition and Detection of Student Emotional States through Bayesian Inference," *Bulletin of Education and Psychology*, vol. 5, no. 5, 2025.
- [9] F. R. T. Ferreira et al., "Development of a framework using deep learning for the identification and classification of engagement levels in distance learning students," *Social Network Analysis and Mining*, vol. 15, no. 37, 2025.
- [10] A. Ahrar and M. Doroodian, "Exploring Eye-tracking Features to Understand Students' Sensemaking of Learning Analytics Dashboards," in *LAK25: The 15th International Learning Analytics and Knowledge Conference*, 2025.
- [11] Rosalyn.ai, "Top 7 Live Online Proctoring Disadvantages—And How You Can Solve Them," Rosalyn Blog, 2024.
- [12] Y. Wang, T. Wang, and T. Yue, "Uncertainty propagation from sensor data to deep learning models in autonomous systems," *Reliability Engineering & System Safety*, 2025.
- [13] R. Yu, Z. Li, L. Xiong, W. Han, and B. Leng, "Uncertainty-Aware Safety-Critical Decision and Control for Autonomous Systems," *IEEE Transactions on Intelligent Vehicles*, 2025.
- [14] A. Kulakayeva, B. Medetov, A. Zhetpisbayeva, and A. Nurlankyzy, "Development of a Robust Neural Network-Based VAD System under Low Signal-to-Noise Ratio Conditions," *Engineering, Technology & Applied Science Research*, vol. 15, no. 6, pp. 30377-30386, 2025.

- [15] H. S. Bovbjerg, J. Østergaard, J. Jensen, and Z.-H. Tan, “Noise-Robust Target-Speaker Voice Activity Detection Through Self-Supervised Pre-training,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2025.
- [16] A. Mahfoud and D. Mohamed, “Real-Time Voice Activity Detection using Hybrid Deep Learning Models with a Multi-Noise Detector,” IEEE Access, 2025.
- [17] R. M. A. Latif et al., “Edge-AI Architecture for Real-Time Computer Vision in Smart Media Applications,” IEEE Internet of Things Journal, 2025.
- [18] Najeebullah, M. Salman, and Z. N. K. Swati, “Face Spoofing Detection using Deep Learning,” arXiv preprint arXiv:2503.19223, 2025.
- [19] G. Wang, Q. Zhao, Z. Zhou, and Y. Liu, “Research on Real-time Mul-tilingual Transcription and Minutes Generation for Video Conferences Based on Large Language Models,” Spectrum of Research, vol. 5, no. 1, 2025.
- [20] S. Kumar V, B. Raveendra Nadh, S. Sureshkumar, A. Suresh Kumar, A. R. S R, and G. Sheeba, “Leveraging Artificial Neural Networks for Real-Time Speech Recognition in Voice-Activated Systems,” ITM Web of Conferences, vol. 76, Art. no. 01003, 2025.