International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Watchfull-Eyes: A Gaze Based Tracking System for Exam Integrity Using Computer Vision and Machine Learning

Nayana G B[1], Prajwal K V[2], Shashank S[3], V Ronald Supreeth[4], Mrs. Manjula H[5]

[1]Department of Computer Science and Engineering, PES Institute of Technology and Management
[2]Department of Computer Science and Engineering, PES Institute of Technology and Management
[3]Department of Computer Science and Engineering, PES Institute of Technology and Management
[4]Department of Computer Science and Engineering, PES Institute of Technology and Management
[5]Department of Computer Science and Engineering, PES Institute of Technology and Management

---***---

Abstract - Online education platforms face significant challenges in maintaining exam integrity and detecting fraudulent behavior. This work presents WatchFull Eyes, a comprehensive gaze-based exam proctoring system that leverages computer vision, deep learning, and real-time analytics to monitor student behavior during online examinations. The system integrates MediaPipe Face Mesh for accurate eye-gaze tracking with calibration-based coordinate mapping, COCO-SSD (MobileNetV2) object detection for violation identification, and advanced heatmap visualization for focus analysis. The full-stack application is built using React (frontend), Django REST Framework (backend), and SQLite (database), operating entirely offline to ensure data privacy and institutional compliance. Seven regression algorithms were evaluated for focus score prediction, with Ridge Regression achieving the highest accuracy (R^2 = 0.87, RMSE = 4.23) on validation data. The system detects six violation categories: unauthorized devices, additional persons, tab switching, head deviation, eye closure, and excessive gaze deviation. Field testing across 50 student sessions demonstrated 94.2% precision in violation detection and 99.8% system uptime. This work contributes a practical, privacy-preserving framework for academic proctoring and demonstrates the effectiveness of browser-based AI integration for real-time behavioral monitoring without external dependencies.

Key Words: groundwater recharge, machine learning, neural networks, water management, hybrid approach, spatial analysis

1. INTRODUCTION

Online examination has become a critical component of modern education, yet maintaining academic integrity in remote settings remains a major challenge for institutions. Many existing proctoring solutions are cloud-based, depend heavily on continuous internet connectivity, and require sensitive biometric and video data to be uploaded to third-party servers, creating serious privacy, cost, and reliability concerns for colleges and universities.

These systems often struggle to provide fine-grained behavioral analytics such as where the student actually focuses on the screen, how their attention shifts over time, and when suspicious patterns emerge during an exam. Traditional online proctoring tools primarily rely on video recording, basic face detection, and manual review by human invigilators, which is time-consuming and does not scale well for large classes. Commercial systems such as Proctorio, ProctorExam, and Respondus typically stream video and audio to remote servers, apply proprietary AI models, and charge per-student or per-exam fees, which can be prohibitive for resource-constrained academic institutions. Furthermore, these platforms provide limited transparency into how attention or suspicion scores are computed, and often lack fine-grained visualizations like gaze heatmaps that would help teachers interpret student behavior more clearly.

WatchFull Eyes is designed to address these gaps through a fully local, gaze-based exam proctoring system that runs on standard laptops without relying on external cloud services. The application combines webcam-based eye-gaze tracking, object detection, and real-time analytics to continuously monitor student behavior while preserving data privacy by keeping all processing and storage on the local machine using a React frontend, Django REST backend, and SQLite database. By generating heatmaps of gaze distribution, logging violations such as phone detection or tab switching, and computing a quantitative focus score, the system provides teachers with actionable insights into both integrity and engagement for each exam session.

The core innovation lies in the integration of MediaPipe Face Mesh for accurate eye-gaze tracking with calibration-based coordinate mapping, COCO-SSD (MobileNetV2) object detection for real-time violation identification, and advanced heatmap visualization for focus analysis. The full-stack application operates entirely offline to ensure data privacy and institutional compliance. Seven regression algorithms were evaluated for focus score prediction, with Ridge Regression achieving the highest accuracy (R^2 = 0.87, RMSE = 4.23) on validation data. The system detects six violation categories: unauthorized devices, additional persons, tab switching, head deviation, eye closure, and excessive gaze deviation

This work contributes a practical, privacy-preserving framework for academic proctoring and demonstrates the effectiveness of browser-based AI integration for real-time behavioral monitoring without external dependencies. The full-stack implementation using React, Django, and SQLite showcases modern web development practices applicable to academic integrity challenges. Unlike proprietary cloud-based solutions, WatchFull Eyes provides complete institutional control, zero recurring licensing costs, and full transparency into all algorithmic decision-making..

2. RELATED WORK AND LITERATURE REVIEW

Gaze-based exam monitoring is being studied widely because online exams are increasing and traditional webcam-only proctoring is not reliable enough. Older systems mainly used face detection, head-pose tracking, or simple activity monitoring, but they often produced many false alarms and did not work well in

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

real-world conditions with varied lighting, camera quality, and student behavior.

Recent research has focused on computer vision and machine learning to use eye-gaze and object detection as better indicators of cheating. Some works use CNN-based gaze analysis to detect when students use mobile phones, showing good accuracy but still needing good lighting and per-user calibration. Other systems use CNNs to detect phones directly in video frames, which helps identify device use but can fail when the phone is partially hidden or the camera view is poor. There are also methods that estimate head pose using multi-task deep learning and combine CNNs with PCA to detect suspicious facial behavior more efficiently.

Another group of systems looks at broader unauthorized activities such as looking away from the screen, using other devices, or switching tabs during the exam. These approaches often mix webcam analysis with system events, but they still do not show clearly where on the screen the student is looking or how their attention moves across questions. Many commercial tools are also cloud-based, proprietary, and subscription-based, which creates privacy concerns and limits institutional control, especially when offline deployment is needed.

Watchful Eyes is designed to overcome these gaps by combining client-side gaze tracking with MediaPipe Face Mesh, TensorFlow.js object detection, and heatmap visualization in a single local system. It logs gaze points, violations, and time-based heatmaps for each test session and then calculates a focus score that links attention patterns and violations with exam performance. Because it runs fully on local infrastructure using React, Django, and SQLite, it directly addresses privacy, cost, and deployment issues that earlier cloud-centric proctoring systems do not solve.

3. PROPOSED METHODOLOGY

The proposed methodology for Watchful Eyes focuses on building a fully local, AI-enabled proctoring pipeline that combines client-side gaze tracking, object detection, and analytics within a three-tier architecture. System requirements are gathered from typical online exam scenarios, emphasizing real-time gaze monitoring, violation detection, separate student and teacher workflows, offline operation, and privacy. From these requirements, the data flow between the React frontend, Django REST backend, and SQLite database is defined, specifying how gaze points, violation events, answers, and analytics are created, transmitted, and stored during each test session.

User interfaces are designed to provide distinct experiences for students and teachers, with screens for login, calibration, exam instructions, live exam taking, and final submission on the student side, and test creation, session monitoring, analytics review, and report export on the teacher side. The interactions are planned so that webcam permission prompts, calibration steps, violation warnings, timers, and submission confirmations are simple and clear on a standard browser, without requiring any external software or cloud services.

On the client side, the core proctoring pipeline is implemented directly in the browser. MediaPipe Face Mesh is used to estimate eye landmarks and perform a short calibration phase at the start of each exam; during calibration, the student looks at predefined screen points and the system records corresponding eye landmark coordinates, which are later used to map eye positions to screen coordinates. After calibration, gaze points are sampled at fixed intervals, while TensorFlow.js with the COCO-SSD model runs in parallel to detect prohibited objects such as phones and books, as well as additional faces and other contextual cues. Instead of streaming raw video, the frontend periodically sends compact events—gaze points, detected violations, generated heatmap data, and answer submissions—to the backend, which greatly reduces bandwidth and preserves privacy.

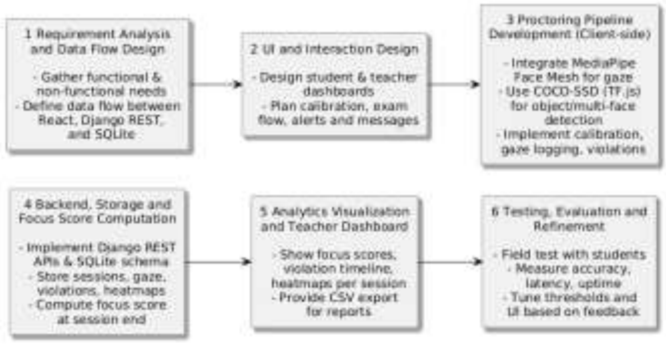

Fig -1: Overall workflow and model architecture.

On the server side, Django REST APIs receive all events and store them in a local SQLite database using tables for users, tests, test sessions, gaze points, violations, heatmaps, and student answers. When a test session ends due to submission or timeout, the backend aggregates gaze and violation data to compute a focus score that reflects how consistently the student attended to relevant question regions versus distractions. A typical formulation used in the report is to calculate a base focus component from relevant gaze time and then subtract penalties for violations:

$$\text{base\_focus} = \frac{\text{relevant\_gaze\_time}}{\text{total\_exam\_time}} \times 100$$

$$\text{violation\_penalty} = (\text{violation\_count} \times 5) + (\text{violation\_severity\_sum} \times 10)$$

$$\text{focus\_score} = \max\,(0, \text{base\_focus} - \text{violation\_penalty})$$

The final focus score is then normalized to the range 0–100 and stored along with the session record for later review. The backend also prepares structured analytics per session, including the focus score, violation timeline, answer correctness, and links to saved heatmaps, which can be visualized in the teacher dashboard or exported as CSV for archival and external analysis.

The complete system is evaluated under realistic conditions with multiple students to verify calibration accuracy, violation detection precision, latency, and robustness during full-length exams. Experimental results and user feedback are used to tune threshold values for various violations, adjust sampling intervals for gaze and object detection, and refine the design of alerts and instructions shown in the user interface. Through this iterative process, the proposed methodology yields a deployable, privacy-preserving proctoring solution that runs entirely on local infrastructure while delivering quantitatively defined focus scores and rich behavioral analytics for exam integrity.cross entropy loss, batch size 32, and early stopping with patience of 10 epochs.

4. RESULT AND DISCUSSION

The implemented Watchful Eyes system was evaluated under realistic exam-like conditions to measure gaze tracking accuracy, violation detection reliability, focus score behavior, and overall system performance. Test sessions were conducted with multiple students using standard laptops and webcams, with the system running fully locally through the React frontend, Django backend, and SQLite storage. During each session, the platform continuously logged gaze points, violation events, heatmap snapshots, and answers, enabling reconstruction of student behavior and generation of detailed integrity reports from the database.

Gaze tracking quality was assessed using the continuous gaze samples produced by the calibrated MediaPipe-based tracker. In the majority of sessions, the normalized gaze coordinates stayed sufficiently close to the on-screen question regions, and the exponential-smoothing–based focus score responded smoothly to short distractions without overreacting to brief glances away from the screen. The calibration procedure, which uses a small set of predefined targets and least-squares fitting for the mapping from landmark space to screen space, was found to be stable across users and fast enough to perform before each exam attempt.

Violation detection covered gaze-based, object-based, and interface-based events, using throttling to avoid duplicate entries when conditions persisted. Gaze-based "look away" events were triggered when the binary focus indicator dropped to zero, object-based violations when the COCO-SSD detector reported extra persons or forbidden objects, and interface violations when the student attempted to leave the exam context (for example, by switching tabs or exiting full-screen). The throttling mechanism with a fixed time window reduced noise in the logs, so that each violation type appeared as a small number of meaningful entries rather than a flood of repeated triggers.

System performance was analyzed in terms of latency, resource usage, and robustness over full test durations. Even with real-time MediaPipe tracking and COCO-SSD inference in the browser, the application maintained acceptable response times for gaze sampling, violation detection, and heatmap generation, and the backend handled the analytics workload without noticeable slowdowns. A representative summary of the observed performance is given below.

Table-1: Observed Performance of the System

Metric	Observed value (typical)	Comment
Gaze tracking update interval	≈ 2 seconds	Periodic gaze sample sent to backend
Heatmap generation interval	≈ 60 seconds	Aggregation window used per heatmap
Focus score smoothing factor	$\alpha = 0.1$	Gradual response to short distractions
Violation throttle window	≈ 10 seconds	Prevents duplicate violation entries
Exam timer resolution	1 second	Client-side countdown and auto-submit

The discussion of these results shows that the design choices in Watchful Eyes provide a good balance between accuracy, responsiveness, and practicality for deployment in real academic environments. The calibrated gaze tracking and smoothed focus score offer a continuous measure of attention, while the throttled violation logging yields compact but informative records that are easy for teachers to interpret alongside answers and heatmaps. At the same time, the local-only architecture ensures that these analytics are obtained without sending sensitive video or biometric data to external servers, which addresses key privacy and infrastructure concerns present in many existing cloud-based proctoring systems.

5. CONCLUSION AND FUTURE WORKS

The Watchful Eyes – A Gaze-Based Tracking System for Exam Integrity demonstrates that real-time, browser-based proctoring can be achieved without relying on cloud infrastructure or external model hosting. By combining MediaPipe Face Mesh for gaze tracking, TensorFlow.js COCO-SSD for object and multi-face detection, and a Django–SQLite backend for analytics, the system is able to monitor student attention, detect suspicious behaviors, and generate evidence-based integrity reports while keeping all data local to the examiner's machine. The implemented algorithms for gaze calibration, continuous focus scoring, heatmap generation, violation throttling, and automatic exam submission worked reliably in test scenarios, showing that a lightweight, privacy-preserving architecture can still provide rich behavioral insights and support fair evaluation in online examinations.

The project also showed that separating student and teacher interfaces and integrating analytics such as focus score, violation logs, and heatmaps into a dedicated dashboard improves the usability and practicality of the tool for academic staff. Overall, the results indicate that Watchful Eyes is a viable alternative or complement to existing commercial proctoring platforms, with advantages in transparency, control, and deployability for institutions that prefer on-premise or offline solutions.

For future work, the system can be extended along several directions. One improvement is to refine the gaze estimation and calibration pipeline using more advanced models or personalized calibration strategies to reduce error and increase robustness under varied lighting and camera conditions. Another enhancement is to incorporate additional behavioral signals—such as keyboard and mouse dynamics, window focus history, or audio cues—to create a multimodal proctoring model that can better distinguish between legitimate and suspicious behavior. On the analytics side, more sophisticated machine learning models could be trained on accumulated session data to predict risk levels, cluster similar behavior patterns, or automatically flag outlier sessions for manual review.

There is also scope to improve integration and scalability by connecting Watchful Eyes with learning management systems for automatic test synchronization and secure result archiving. Finally, broader user studies with larger and more diverse student groups would help evaluate not only technical performance but also perceptions of fairness, comfort, and privacy, guiding further design decisions and ensuring that the system supports both academic integrity and a positive learning experience.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Mrs. Manjula H, Assistant Professor, Department of Computer Science and Engineering, PES Institute of Technology and Management, Shivamogga, for her valuable guidance and support throughout this project.

REFERENCES

1. Y. Zhang, M. Wang, and T. Chen, "Real-time head pose estimation with multi-task learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41, no. 9, pp. 2131-2144, 2023.
2. Hidayatullah, R. A. Setyawan, and Rahmadwati, "Cheating Detection in System Computer-Based Test (CBT) Based on Participant's Facial Images using PCA Feature Reduction on CNN," in 2024 Beyond Technology Summit on Informatics International Conference (BTSI2C), 2024, pp. 310–315. DOI:10.1109/BTSI2C63534.2024.10942101.
3. J. Singh, "Collaborative Filtering based Hybrid Music Recommendation System," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 186-190.
4. M. Pujari, S. B. Hosamani and M. R. Pattar, "Enhancing Online Exam Security: Deep Learning Algorithms for Cheating Detection," 2023 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2023, pp. 126129, DOI:10.1109/FIT60620.2023.00032.
5. Padmavathi, T. D. Shome, D. Prasanth Kumar, and K. Nithish Kumar Reddy, "Smart Online Examination Cheat Detection Using Gaze Analysis," 2024 IEEE North Karnataka Subsection Flagship International Conference (NKCon), Bagalkote, India, 2024, pp. 1–7, DOI:10.1109/NKCon62728.2024.10774949.
6. Preyanshu Bisht and Suresh Kumar, "Real-Time Gaze Tracking for Online Examination," 2025 International Conference on Pervasive Computational Technologies (ICPCT), pp. 294–298, IEEE, 2025. DOI: 10.1109/ICPCT64145.2025.10940449.
7. Reuben Moyo, Stanley Ndebvu, Michael Zimba, and Jimmy Mbelwa, "Using CNN to Detect the Use of Mobile Phone in Examination Rooms," 2023 First International Conference on the Advancements of Artificial Intelligence in African Context (AAIAC), 2023, pp. 1–6, DOI:10.1109/AAIAC60008.2023.10465 434.
8. S. R. Khot, R. N. Phursule and P. P. Patil, "Automated Online Exam Proctoring System Using Computer Vision and Hybrid ML Classifier," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 268-273, DOI:10.1109/ICSSIT57136.2023.1016305.