International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

AI-Based Online Exam Proctoring System Using Machine Learning

Ms. Harshada Patil
GODAVARI INSTITUTE OF MANAGEMENT AND RESEARCH, JALGAON
harshup1314@gmail.com | +91 9325402093

Prof. Mr. Prashant Bahekar
GODAVARI INSTITUTE OF MANAGEMENT AND RESEARCH, JALGAON
prashant@gmail.com | +91 8007900047

Abstract
With the rapid expansion of digital learning, maintaining academic integrity in remote assessments has become a critical requirement. The AI-Based Online Exam Proctoring System is a Flask-based web application designed to support invigilation using artificial intelligence, machine learning, and computer vision. The system provides a web-based login and quiz workflow while monitoring selected candidate activities through webcam and microphone input. It incorporates facial and facial-landmark analysis to observe face presence, eye and head movements, and uses Eye Aspect Ratio (EAR) calculations as an indicator of eye closure patterns. Object-detection models such as YOLO can be configured to identify selected prohibited items, while audio-processing techniques can monitor sound activity and possible speech anomalies. When configured conditions are detected, the system can display alerts through dialog messages and system sounds for review. Developed using Python, Flask, OpenCV, MediaPipe, machine-learning/deep-learning libraries, and JavaScript, the proposed system aims to make remote examination monitoring more organized and accessible. The paper presents the system study, methodology, AI architecture, performance-evaluation approach, privacy considerations, applications, and future scope. Automated alerts are treated as indicators for human review rather than conclusive proof of misconduct.

Keywords: AI Proctoring, Computer Vision, Deep Learning, Face Recognition, MediaPipe, Gaze Tracking, Head Pose Estimation, YOLO Object Detection, Online Exam Security, Flask Framework, Precision Invigilation.

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

1. INTRODUCTION

Online examination platforms have expanded significantly in higher education, distance learning, and corporate recruitment. They allow candidates to take assessments from different locations and reduce the need for physical examination centres. However, remote assessments introduce challenges related to examination security, candidate authentication, academic honesty, and supervision. Traditional invigilation depends on human proctors observing candidates through live video feeds. This approach can be resource-intensive, difficult to scale for large candidate groups, and vulnerable to human oversight when several video streams must be monitored simultaneously.

With advancements in Artificial Intelligence (AI), Machine Learning (ML), and Computer Vision, automated proctoring solutions can assist invigilators by analyzing selected visual and audio signals. Facial analysis can help determine whether a face is visible in the camera frame. Facial landmarks can be used to estimate eye and head movement, while object-detection models can identify configured classes of objects. Audio-processing techniques can measure sound activity and may help identify possible speech or unusual background noise. These techniques can provide additional information during an examination, although their outputs can be affected by lighting, camera position, background conditions, and individual behavior.

The AI-Based Online Exam Proctoring System is a web-based examination and monitoring platform developed using Python and the Flask framework. The candidate accesses a login or sign-in page through a browser. After successful login, the quiz page opens and the camera is activated alongside the examination for proctoring. The monitoring components analyze available camera and audio input. When a configured condition is detected, an alert can be displayed in a dialog box and accompanied by a system sound. The main aim is to provide timely indications of events that may require attention and reduce the amount of continuous manual observation.

The system combines functional modules for user authentication, quiz delivery, face and facial-landmark analysis, eye and head movement monitoring, object detection, audio activity monitoring, and alert generation. The exact functions available depend on the installed libraries, configured model files, device permissions, and the implementation enabled in the application. The system is designed as an assistance tool; final decisions about examination misconduct should remain subject to human review and institutional rules.

The main objectives are to provide a simple browser-based examination workflow, support camera-based observation during a quiz, analyze selected visual or audio events, notify the relevant user when configured conditions occur, and maintain a structured application process. A further objective is to study how AI-based monitoring can be integrated with a conventional web application without treating automated outputs as final judgments.

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

2. SYSTEM STUDY AND ANALYSIS

2.1 Problem Statement and Motivation

Remote examinations face several security risks, including candidate impersonation, unauthorized reference material, off-screen communication, the presence of additional people, and external verbal assistance. A candidate may also move away from the screen or use a prohibited device without being immediately noticed. Human invigilators who monitor many candidates at the same time can experience fatigue, and important events may be missed. Reviewing recorded examination sessions after completion can require substantial time and delay administrative decisions.

The motivation for the proposed system is to combine a web-based quiz environment with automated monitoring support. By analyzing selected camera and microphone signals during an examination, the application can raise alerts when predefined conditions occur. These alerts can help an invigilator focus attention on relevant moments. The system does not remove the need for human supervision; rather, it aims to support it through an organized and technology-assisted workflow.

A practical proctoring application must also consider technical constraints. Candidates may use laptops with different camera resolutions, older processors, unstable internet connections, or different operating systems. Camera permissions may be denied, microphone input may be unavailable, and low light can reduce face-detection quality. The application should therefore distinguish technical failures from possible examination events and communicate errors clearly.

2.2 Existing Solutions vs. Proposed System

Existing online examination approaches commonly use password-based login, timed quizzes, browser restrictions, live human supervision, or post-examination video review. Password authentication controls access to the portal but does not continuously observe the examination environment. Manual proctoring can provide contextual judgment but requires staff time and attention. Basic lockdown tools may restrict some computer activities but may not analyze facial movement, visible objects, or sound activity.

The proposed system combines the login and quiz process with camera-based observation and optional audio analysis. It uses computer-vision components to examine face presence and selected movements, and a configured object-detection model to identify supported object classes. Audio features can be used to identify sound activity. Alerts are presented during the examination so that possible events can be reviewed without relying exclusively on lengthy post-examination recordings.

Parameter	Manual Human Proctoring	Basic Digital Lockdown	Proposed AI Proctoring System
Monitoring	Human observation of live video	Restricts selected device/browser actions	Automated analysis of selected camera and audio signals
Scalability	Needs more invigilators as candidate numbers	Depends on platform capacity	Designed to assist concurrent monitoring; capacity needs

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

	grow		testing
Detection	Depends on attention and judgment	May not analyze candidate behaviour or surroundings	Can flag configured face, movement, object and audio events
Response	Manual intervention	Platform restrictions or logs	Dialog messages and sound alerts for review
Limitations	Human fatigue and resource requirements	Limited environmental awareness	False alerts, device quality, model and privacy constraints

The comparison indicates that the proposed approach adds automated observation to the normal examination workflow. It should not be interpreted as a replacement for a human invigilator. The value of the system depends on whether its alerts are accurate, understandable, and useful in the context of the examination.

3. RESEARCH METHODOLOGY

The research methodology integrates webcam-based computer-vision processing, selected machine-learning and deep-learning techniques, audio signal analysis, and a Flask web application. The operational process begins when a candidate accesses the portal and completes authentication. After login, the quiz page is displayed and the monitoring component is initialized. Camera frames and, where enabled, microphone input are processed by their respective modules. The system checks configured conditions and generates alerts for events that may need review.

The methodology follows a modular design. Each component has a defined responsibility: the web interface collects user actions, the Flask backend manages application flow, the database stores required records, the monitoring modules process input, and the alert component communicates events. A modular design makes it easier to test individual functions and replace a model or library without redesigning the complete application.

3.1 System Pipeline & Machine Learning Modules

1. **Candidate Authentication:** The user submits login information through the portal. The Flask backend validates the credentials against the application records and allows an authenticated user to access the quiz. Secure password hashing and controlled session handling should be used. Face recognition should only be described as an active identity-verification feature if the corresponding face-matching component is implemented and enabled in the deployed version.

2. **Face Detection and Landmark Analysis:** The camera feed is processed to locate a face and identify facial landmarks. MediaPipe Face Mesh can provide landmark coordinates around the eyes, nose, mouth, and face contour. These coordinates support later analysis of eye and head movement. The quality of the result depends on the camera feed, face visibility, lighting, and processing configuration.

3. **Gaze and Head Pose Estimation:** Facial landmark positions across frames can be used to estimate head orientation, commonly described using pitch, yaw, and roll. Repeated or sustained movement away from the expected screen-facing direction may be treated as a possible event. A

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

single movement is not sufficient to conclude that a candidate is cheating, because normal examination behavior can include looking away briefly.

4. **Eye Aspect Ratio (EAR) and Mouth Landmark Calculation:** EAR compares vertical eye-opening distances with horizontal eye width using selected eye landmarks. A sustained reduction in EAR can indicate eye closure and may be used to study blinking or drowsiness-related patterns. Mouth landmark distances can be observed as an additional signal, but they cannot reliably prove speaking without suitable validation and context.

5. **Object Detection (YOLO Framework):** A configured YOLO model can inspect video frames and identify object classes included in its model. Depending on the model, classes may include mobile phones, books, or other selected objects. The system can use confidence thresholds to decide when to show an alert. Detection is limited to the classes supported by the model and may be affected by object size, camera quality, occlusion, and lighting.

6. **Acoustic Anomaly Detection:** Audio captured through a microphone can be analyzed using signal features such as Root Mean Square (RMS) energy. MFCC features or a trained classifier may be used in more advanced implementations to characterize audio. A sound-level event can be flagged for review, but background noise and environmental sounds can also trigger alerts.

3.2 Processing and Alert Workflow

The monitoring pipeline receives frames or audio segments, performs the relevant processing, and compares the resulting values with configured thresholds. When a condition is met, the application creates an alert message. A cooldown period can reduce repeated alerts for one continuous event. The alert may be shown in a dialog box and accompanied by a system sound. If the camera or microphone is unavailable, the application should report the technical issue clearly rather than interpreting missing input as suspicious activity.

A useful alert should identify the type of event and, where appropriate, the time at which it occurred. The alert wording should remain neutral, such as 'Face not detected for the configured interval' or 'Sound activity detected'. It should avoid statements that directly accuse a candidate. The reviewer can then consider the alert alongside the examination rules and other available information.

Threshold selection is an important part of this workflow. A threshold that is too sensitive may generate frequent false alerts, while a threshold that is too relaxed may miss events. The values should be selected through repeated testing in realistic conditions. A cooldown interval can prevent a continuous event from producing a large number of identical messages, improving the readability of the alert interface.

3.3 Web Application Integration

The user interface is developed using HTML, CSS, and JavaScript, while Flask handles server-side routes and application logic. The login and sign-in pages provide access to the portal. After authentication, the quiz interface presents questions and answer options. The camera monitoring component runs alongside the quiz according to the application's configuration. JavaScript supports interface interactions and communication with backend routes, while Python libraries

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

perform the configured monitoring operations.

The browser-based design separates the candidate-facing interface from the server-side processing. This separation helps organize the code and allows the quiz interface to be updated without changing the underlying monitoring logic. The interface should provide clear instructions before camera or microphone access is requested. It should also show the candidate when the quiz is active and provide understandable messages if a required device cannot be accessed.

3.4 Database and User Records

The backend uses database helper functions to manage user-related records and application data. Database connectivity must be configured for the target environment, and the application should handle connection failures safely. User credentials should be stored using secure password hashing. Queries should use parameterized statements, and access to user information should be restricted to authorized application functions.

Only information required for account management and examination operation should be stored. If examination events are recorded, each record should contain only necessary details, such as event category and timestamp. Database errors should not expose passwords, connection strings, or other sensitive configuration values to the browser. Backup and retention procedures should be defined by the institution operating the application.

3.5 System Testing

Testing should cover the complete workflow, including registration or sign-in, valid and invalid login attempts, quiz loading, question navigation, camera activation, alert display, and sound output. Functional testing checks whether individual modules work as intended. Integration testing checks communication between the browser, Flask routes, database helper, and monitoring libraries. Device testing checks behavior when the camera or microphone is unavailable or permission is denied. Performance testing should measure processing load and responsiveness under realistic conditions.

Representative test cases should include a candidate remaining in front of the camera, temporarily moving out of frame, changing head direction, a supported object appearing in the camera view, background noise, and microphone permission being denied. Each case should be repeated several times. The tester should record whether the expected alert appeared, whether an incorrect alert appeared, and whether the application continued to operate normally.

Testing should also include negative cases. For example, a brief natural glance away from the screen should not automatically be treated as misconduct. A face-detection failure caused by poor lighting should be recorded as a technical limitation. These cases help determine whether the alert rules are practical and whether users can understand the difference between a technical warning and a possible examination event.

4. AI ARCHITECTURE & PERFORMANCE METRICS

The system architecture combines a web interface, Flask application logic, input-capture components, AI-based processing modules, and an alert interface. The presentation layer contains

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

the login and quiz pages. The application layer manages authentication and examination flow. The monitoring layer receives camera and audio input. The analysis layer applies face detection, facial landmark analysis, EAR calculations, object detection, and audio feature extraction. The alert layer presents the resulting events to the user or invigilator for review.

The data flow begins with the candidate's interaction with the browser. Authentication information is sent to the Flask application for validation. Once the user is authorized, the quiz is delivered. During the quiz, the monitoring process obtains permitted input from available devices. The analysis modules return measurements or detection results, and the alert logic determines whether any configured condition should be reported. This layered architecture supports separation of responsibilities and makes individual modules easier to maintain.

4.1 Face and Facial Landmark Analysis

Face detection identifies whether a face is present in a frame. Facial landmark analysis provides coordinates that can be used to study the relative positions of facial features. These positions may support head movement and eye-region analysis. The processing pipeline should maintain a consistent frame size and handle missing or low-confidence landmarks. Changes in camera angle, lighting, and face occlusion can affect the output.

The application can examine face presence over a short time interval rather than relying on one isolated frame. This helps reduce unnecessary alerts caused by momentary detection failures. Landmark coordinates should be checked for validity before calculations are performed. If the model cannot locate the required points, the application should skip that calculation safely and continue processing later frames.

4.2 Eye and Head Movement Analysis

The eye-monitoring component can calculate EAR from selected landmark points. Head movement can be estimated from facial geometry over time. A practical implementation should use a time window rather than relying on one frame, because brief movements and blinks are normal. Thresholds should be selected through testing and should be configurable for different camera conditions.

For head-pose analysis, changes in the estimated orientation can be compared across consecutive frames. The system may record sustained or repeated movement as an event for review. The interpretation must remain cautious: candidates may look away to think, read permitted material, adjust their seating position, or respond to environmental distractions. A movement score alone cannot establish intent.

4.3 Object Detection Architecture

The object-detection component processes selected video frames using a configured YOLO model. The model returns class labels, confidence scores, and bounding boxes. The application can filter detections using a confidence threshold and display an alert when a configured class is found. The model's training classes and version should be documented so that the system's actual detection capability is clear.

Detection quality depends on the size and visibility of an object in the frame. A phone partly

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

hidden behind another object may not be detected, while an object with a similar appearance may be classified incorrectly. Frame sampling can reduce processing load, but the sampling interval should be selected carefully. The system should not claim to detect every prohibited object unless those classes have been included in the model and tested.

4.4 Audio Monitoring Architecture

The audio component captures short segments from the microphone and calculates signal features. RMS energy can be used to estimate the strength of a sound signal. More advanced implementations may use MFCC features with a classifier to distinguish speech-like activity from other sounds. The component should be tested with different room conditions and should not treat every sound as evidence of misconduct.

Audio monitoring is particularly sensitive to environmental conditions. Fans, traffic, other household members, keyboard sounds, and microphone quality can affect the signal. If speech classification is used, it requires a suitable labelled dataset and evaluation. The system should process only the audio needed for the examination purpose and should clearly explain whether audio is analyzed live or retained.

4.5 Performance Evaluation Metrics

The system should be evaluated using module-specific measures. For face detection and object detection, precision, recall, and false-positive rate can describe detection quality. For object detection, mean Average Precision (mAP) and Intersection over Union (IoU) may be reported when a labeled test set is available. For real-time monitoring, frames per second and processing delay can describe responsiveness. For audio monitoring, precision and recall on labeled audio segments can be useful. No verified benchmark values are available in the supplied project material, so numerical accuracy or latency results should be added only after controlled testing.

Precision measures the proportion of reported detections that are correct, while recall measures the proportion of relevant events that the system successfully identifies. A high false-positive rate can reduce user trust because genuine candidates may receive repeated alerts. A high false-negative rate can reduce the usefulness of monitoring because relevant events may be missed. Therefore, evaluation should consider both types of error rather than reporting a single accuracy value.

For a real-time application, responsiveness is also important. Processing delay describes the time between receiving input and displaying an alert. Frames per second indicates how frequently video frames can be processed. These measures should be recorded on the hardware expected to run the application. Results from one computer should not automatically be generalized to all candidate devices.

4.6 System Performance Factors

The overall performance depends on hardware capability, camera resolution, frame rate, model size, number of active processing modules, and network conditions. Running several AI components simultaneously may increase CPU or GPU usage. The application can be optimized by processing selected frames, resizing input images where appropriate, and avoiding duplicate

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

processing of the same frame. Any optimization should be checked to ensure that it does not reduce detection quality beyond an acceptable level.

Memory use and application stability should also be monitored during longer examination sessions. Resources that are not needed should be released when the quiz ends. Camera and microphone streams should be closed properly, and background threads should be stopped in a controlled manner. These practices help prevent unnecessary resource consumption and reduce the risk of device conflicts during later sessions.

4.7 Evaluation and Validation

A structured evaluation should include normal candidate behavior and controlled scenarios representing events that the system is intended to detect. Examples include a face temporarily leaving the frame, repeated head turns, the appearance of a supported object class, and background speech or noise. Each scenario should be repeated under different lighting and camera positions. Results should record true detections, missed events, false alerts, processing time, and system resource use. Independent test data should be used where possible to avoid evaluating a model only on data used during training.

The validation report should identify the software versions, hardware configuration, camera settings, model files, thresholds, and test conditions. This information makes the evaluation reproducible and helps explain differences between test environments. If a module has not been evaluated, the paper should describe it as a proposed or configured component rather than claiming a measured level of reliability.

5. DATA PRIVACY & ETHICAL CONSIDERATIONS

Deploying an AI-based proctoring system requires careful attention to privacy, transparency, fairness, and security. Webcam and microphone streams may contain personal information about candidates and their surroundings. The institution operating the system should explain why monitoring is required, what data is captured, how the data is processed, who can access it, and how long any records are retained. Collection should be limited to information necessary for the examination purpose.

Where possible, processing should be performed without retaining raw video or audio longer than necessary. If incident records are stored, they should be protected by access controls and secure transmission. Authentication information should be protected using password hashing, and application sessions should be managed securely. API keys, database credentials, and model files should not be exposed through public client-side code.

Automated alerts may be inaccurate. A candidate may look away to think, move naturally, adjust their position, or be affected by lighting and camera limitations. Background sounds may be unrelated to examination misconduct. Differences in devices, disabilities, assistive technologies, and personal environments can affect the system's output. Therefore, alerts should be treated as indications that require human review, not as automatic proof of cheating.

5.1 Fairness, Transparency, and Human Review

The system should provide clear information about monitoring before an examination begins. Institutions should define consistent examination rules and a fair process for reviewing incidents. Candidates should have an opportunity to explain technical problems or challenge an alert. Models and thresholds should be tested across diverse users and environments, and known limitations should be documented. Human oversight is necessary for decisions that may affect examination results.

Transparency also requires understandable alert descriptions. A candidate should know whether the system checks face visibility, movement, objects, or sound. The institution should provide contact information for technical support and explain what happens if a device stops working. Clear procedures reduce confusion and help candidates prepare their devices before the examination begins.

5.2 Security and Responsible Data Handling

Security measures should include secure password storage, restricted database access, protected network communication, controlled access to incident records, and appropriate retention policies. Developers should avoid storing unnecessary personal data and should test the application for common web security weaknesses. Monitoring should stop when the examination ends, unless a clearly communicated and justified process requires otherwise.

The application should validate user input on the server side and avoid displaying detailed internal errors to candidates. Database credentials should be maintained in protected configuration rather than embedded in publicly accessible files. Access to stored records should be limited according to user roles. Institutions should periodically review access permissions and remove records when the approved retention period expires.

6. REAL-WORLD APPLICATIONS & FUTURE SCOPE

The AI-Based Online Exam Proctoring System may be useful in university and college examinations, online quizzes, distance-learning assessments, competitive examination practice, corporate IT recruitment tests, and professional certification activities. Its browser-based workflow can support candidates who take assessments from different locations. The monitoring alerts can assist invigilators by highlighting selected events that may need attention. The system's suitability for high-stakes examinations depends on reliable testing, institutional approval, privacy safeguards, and appropriate technical infrastructure.

6.1 Educational Institutions

Colleges and universities can use online examination platforms for internal tests, practice assessments, and selected remote evaluations. A proctoring support system can provide an additional layer of observation when candidates are not physically present in an examination hall. Institutions should configure the monitoring rules according to the assessment type and provide support for candidates who face device or connectivity issues.

For classroom practice tests, the system may help teachers understand how a remote quiz workflow operates and identify technical problems before larger assessments. Institutions can begin with low-stakes use, collect feedback from students and invigilators, and improve the

International Journal of Convergent Technologies and Future Systems (IJCTFS)
Website: https://www.maulikavyanshufoundation.org/IJCTFS/index.php
Email: maulikavyanshuijctfs@gmail.com | kavyanshufoundation@gmail.com
Volume 02 Issue-06 November- December 2026 ISSN NO: 3108-1312

interface and monitoring rules before considering wider deployment.

6.2 Recruitment and Certification

Organizations conducting remote screening tests or certification assessments may use automated alerts to help invigilators review candidate sessions. The system can support the examination workflow by combining account access, timed quizzes, and monitoring. Before use in recruitment or certification decisions, the system should be validated and its alerts should be reviewed through a transparent process.

In recruitment settings, the system should be applied consistently to all candidates and should not be the only basis for rejecting an applicant. Technical difficulties should be handled through a documented support process. Human reviewers should consider the context of an alert and provide a fair opportunity for clarification.

6.3 Future Scope

Future improvements may include stronger identity-verification workflows, more robust liveness checks, improved gaze and head-pose estimation, and object-detection models trained on examination-specific data. Audio classification may be improved to distinguish speech from common background sounds. A dedicated invigilator dashboard could display event summaries and allow authorized reviewers to examine alerts in context. Secure incident logging and role-based access control could improve administration.

Further work may explore efficient edge processing for environments with limited internet connectivity, provided that device capability and privacy requirements are considered. The system could also be enhanced with better accessibility support, configurable exam policies, and clearer recovery procedures for camera, microphone, or network interruptions. Any new module should be evaluated independently and as part of the integrated application before being relied upon.

Another useful direction is to develop a configurable examination policy. Different assessments have different requirements: some may permit calculators or printed notes, while others may not. A configurable policy would allow institutions to define which events should generate alerts for a particular examination. The system could also provide a summary report showing event categories and timestamps, while protecting candidate information and avoiding unsupported conclusions. Future evaluation should include feedback from students, teachers, and examination administrators. Such feedback can identify confusing alerts, accessibility barriers, and workflow problems that may not be visible through technical testing alone. Improvements should be prioritized according to reliability, usability, privacy, and fairness.

7. CONCLUSION

The AI-Based Online Exam Proctoring System Using Machine Learning demonstrates an approach to combining web-based examination delivery with automated monitoring support. Developed around Python and Flask, the application provides a login and quiz workflow and can activate camera-based proctoring alongside the examination. Its monitoring architecture brings together face and facial-landmark analysis, eye and head movement observation, configured

object detection, audio activity analysis, and alert presentation.

The integration of computer-vision and audio-processing techniques can help identify events that may require an invigilator's attention. MediaPipe facial landmarks can support analysis of eye and head movement, EAR can provide an indicator of eye closure patterns, YOLO-based models can detect configured object classes, and audio features can characterize sound activity. Dialog messages and sound alerts can make the monitoring process more noticeable during an active examination.

The system also has important limitations. Detection quality depends on device and environmental conditions, model configuration, and the accuracy of thresholds. False alerts may occur, and automated signals cannot establish misconduct on their own. Privacy, fairness, accessibility, and secure data handling must be considered throughout development and deployment. The system should therefore be used as an assistance tool with human review rather than as an autonomous decision-maker.

Further evaluation using documented test scenarios, representative data, and measurable performance metrics is required to establish the reliability of each module. With careful validation, improved security, and transparent institutional procedures, the proposed system can contribute to more organized and accessible remote examination supervision. The work provides a foundation for further development of reliable, responsible, and user-friendly online assessment support.

REFERENCES

[1] Malhotra, M., & Chhabra, I. (2026). Ensuring academic integrity through automated online exam proctoring: A decade-long systematic review. Discover Education, 5, Article 207.

[2] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[3] Lugaresi, C., et al. (2019). MediaPipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172.

[4] Schroff, F., Kalenichenko, V., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815–823.

[5] Flask Documentation. Flask Web Development Framework. https://flask.palletsprojects.com/

[6] OpenCV Documentation. Open Source Computer Vision Library. https://docs.opencv.org/

[7] Google. MediaPipe Solutions Guide. https://ai.google.dev/edge/mediapipe/solutions/guide