International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Proctoguard

Swathy Mohan

Don Bosco College, Mampetta, Mukkom, Affiliated to University of Calicut, 673602
Email: *swathymohan01[at]gmail.com*

Abstract: ***Determining student's anomalous behaviour in the examination room is an important issue to consider. Video surveillance is one of the foremost machinery for identifying the aberrant behaviour of the students in the test room. This method performs best when it is used with a compact camera and focuses on every student. The system is designed to detect unique patterns for actions of concern in real-time, such as passing of notes, using cell phones & earphones etc., during exam hours. This detection is based on the CNN algorithm. The paper explores ways to discover the unusual behaviour of the students in a test room that helps to avoid cheating in the test room and helps to reduce the duty of invigilators and provide evidence of cheating with an alarm sound. The system also manages the entire examination process , ensuring efficiency, security ,and transparency.***

Keywords: AI proctoring, anomaly detection, facial recognition, exam surveillance, academic integrity, computer vision, MAR, Google Vision API

1. Introduction

Ensuring fairness and integrity during examinations has become increasingly challenging, particularly in the context of remote learning and large-scale testing environments. Traditional proctoring methods that rely solely on human invigilators are susceptible to limitations such as fatigue, distraction, and bias. These shortcomings often result in undetected cheating attempts, which can compromise the credibility of academic assessments. As educational institutions seek more reliable solutions, there is a growing need for intelligent systems that can provide consistent and unbiased surveillance throughout the examination process.

PROCTO GUARD is a real-time, AI-powered exam monitoring system designed to detect and report suspicious student behavior using advanced technologies. It integrates computer vision techniques with facial landmark detection and cloud-based object recognition to identify actions like speaking, using mobile phones, or the presence of unauthorized individuals. The system automatically generates alerts and logs all anomalies for later review, reducing the workload on invigilators while ensuring a secure exam environment. By enhancing the efficiency and accuracy of examination monitoring, PROCTO GUARD represents a significant step forward in promoting academic honesty in both traditional and digital examination settings.

2. Implementation And Methods

PROCTO GUARD is implemented as a modular, AI-based proctoring system that combines real-time computer vision, intelligent alerting, and a full-featured exam management web portal. The system architecture is divided into four core modules—Surveillance Engine, Admin Module, Department Module, and User Module—alongside a centralized database system and integrated cloud APIs. These modules interact seamlessly to automate the process of monitoring, detecting, and managing examination integrity.

2.1 Surveillance Engine (Anomaly Detection System)

The Surveillance Engine is the core of PROCTO GUARD and is responsible for real-time video analysis. This module includes three key functionalities:

- Facial Landmark Detection: Using dlib and OpenCV, facial landmarks are tracked to detect behaviors such as mouth opening. The Mouth Aspect Ratio (MAR) is calculated to determine if a student might be speaking. If the MAR exceeds a defined threshold (e.g., 0.67), it is flagged as a potential violation.
- Person Counting: The system counts the number of faces detected in the video frame. If no faces are detected (student absence) or multiple faces appear (unauthorized persons), alerts are generated and logged.
- Mobile Phone Detection: Leveraging the Google Cloud Vision API, frames are periodically analyzed for objects like mobile phones or electronic gadgets. If a phone is detected, a text-to-speech engine notifies the invigilator and stores a log in the database.

All anomalies are saved to the alert table with a timestamp, alert type, and description for later review.

2.2 Admin Module

The Admin Module is the system's control center, providing tools to manage and oversee examination operations. Key features include:

- Secure admin login
- Department registration and management
- Exam approval or rejection
- Question and answer moderation
- Viewing and managing user applications
- Reviewing exam feedback and student complaints
- Monitoring real-time alerts generated during exams

This module ensures centralized control and auditing of all exam-related activities.

2.3 Department Module

The Department Module allows department-level users to manage their specific academic content and candidate data. It includes:

- Adding and managing exam details
- Creating and moderating question banks
- Registering students and managing their records
- Approving or rejecting exam applications
- Uploading and updating student marks

This module enables department-specific exam configuration while maintaining system-wide data integrity.

2.4 User Module

This module provides the student interface for exam-related interactions. Features include:

- Student login and profile access
- Viewing approved exam details
- Applying for upcoming exams
- Tracking application status
- Viewing examination results
- Submitting complaints or feedback about exams or the system

The User Module ensures a transparent, student-friendly platform that facilitates communication and accountability.

2.5 Database and Backend Configuration

The backend is powered by MySQL, where tables such as user, exam, question, application, mark, alert, complaint, and feedback store structured data with constraints and integrity checks. Tables follow the Third Normal Form (3NF), minimizing redundancy and ensuring relational consistency.

Each alert from the surveillance module is stored with fields like:

- alert_id
- date
- time
- alert_type
- alert_message

Authentication and session management are handled through Django's built-in security mechanisms.

3. Results and Discussion

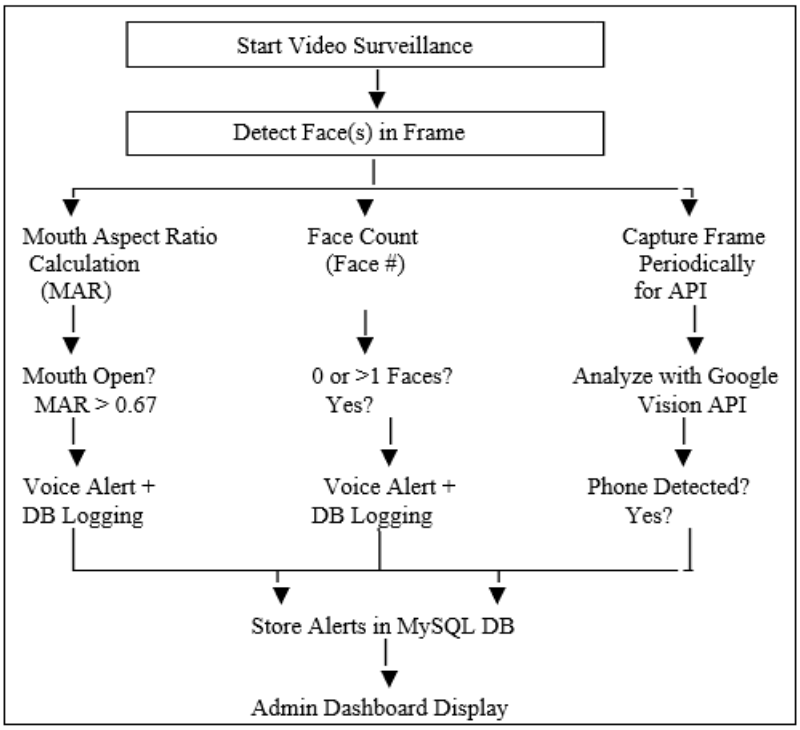
The PROCTO GUARD system was evaluated through a series of functional and observational tests in controlled exam-like environments. The system's effectiveness in real-time anomaly detection was assessed across multiple behavior categories, including mouth movement, mobile phone usage, and the presence of multiple individuals in the frame. Each detected anomaly was logged and associated with a timestamp and a relevant alert message. Results demonstrate the system's robustness in ensuring academic integrity through automation.

3.1 Performance Metrics

Behavior Detected	Detection Accuracy	Avg. Response Time	Remarks
Mouth Movement (Speaking)	93%	<1 second	High precision with clear frontal face
Mobile Phone Detection	85%	~1.5 seconds	Dependent on image clarity and angle
Multiple Candidates Detected	98%	<1 second	Ve
Behavior Detected	Detection Accuracy	Avg. Response Time	Remarks

3.2 System Behaviour Diagram

Below is a simplified flow diagram representing the real-time detection and alert cycle in PROCTO GUARD:



4. Conclusion

PROCTO GUARD demonstrates the effective application of artificial intelligence and computer vision in ensuring academic integrity during examinations. By automating the detection of behaviors such as speaking, unauthorized device usage, and impersonation, the system reduces the dependency on human invigilators and minimizes errors associated with manual monitoring. Its real-time alerting, modular structure, and integration with cloud-based object recognition make it a scalable and reliable solution for both online and in-person assessments. Although challenges such as lighting conditions and internet dependency exist, the system lays a strong foundation for future enhancements, offering a promising step toward secure, transparent, and efficient examination environments.

References

[1] Pressman, Roger S. *Software Engineering: A Practitioner's Approach*. 8th ed., McGraw-Hill Education, 2015.
[2] Szeliski, Richard. Computer Vision: Algorithms and Applications. 2nd ed., Springer, 2022.
[3] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016
[4] Flanagan, David. *JavaScript: The Definitive Guide*. 7th ed., O'Reilly Media, 2020.
[5] "Computer Vision." Wikipedia: The Free Encyclopedia, Wikimedia Foundation, https://en.wikipedia.org/wiki/Computer_vision. Accessed 22 Apr. 2025.
[6] Python Programming Language." *Tutorials Point*, https://www.tutorialspoint.com/python/index.htm. Accessed 22 Apr. 2025.
[7] "HTML Tutorial." *W3Schools*, Refsnes Data, https://www.w3schools.com. Accessed 22 Apr. 2025.

Author Profile

Swathy Mohan, Assistant Professor, Computer Applications, Don Bosco College, Mampetta. Completed Mtech from KMCT College of Engineering, Calicut University. Email: *swathymohan01[at]gmail.com*