



International Journal of Advanced Research in Education and Technology (IJARETY)

Volume 13, Issue 3, May - June 2026

Impact Factor: 9.971



Exam Monitoring System

Dhanush Gowda C D, Gnaneswari Gnanaguru

Department of Master of Computer Applications, CMR Institute of Technology, Bengaluru, Karnataka, India

Associate Professor, Department of Computer Applications, CMR Institute of Technology, Bengaluru,
Karnataka, India

ABSTRACT: Online learning's rise pushes more need for trust-worthy exam setups over the internet. Human supervisors watch exams now, or videos get checked afterward - this way burns money, struggles to grow, misses cheating fast. Instead of old ways, a system driven by artificial intelligence steps in, using smart image reading and pattern learning tools. It keeps eyes on test takers without pause when they take quizzes online. A camera keeps watching the test taker without stopping. Instead of pausing, it checks where the person is looking. Eyes that close too often get flagged by the software. Movement around the lips might signal talking. Even slight tilts of the head are measured closely. Suspicious signs pop up when patterns seem off. Another method scans the room at once using fast object spotting. Phones hidden nearby show up through instant detection layers. Books left in view trigger alerts just as quickly. Electronic gadgets glow on screen even if small. Every frame gets broken down before the next arrives. Monitoring runs quietly in the background throughout. Python powers the system, connecting tools like OpenCV, Dlib, and NumPy alongside machine learning models to handle live data. Instead of human supervisors, it runs checks automatically - keeping exams fair while cutting down on oversight expenses and handling bigger test events smoothly. This method works well under pressure, delivering steady results for remote assessments; schools and credential groups across the globe could begin using it right away..

KEYWORDS: Artificial Intelligence, Computer Vision, Online Exam Proctoring, Deep Learning, OpenCV, YOLO, Face Detection.

I. INTRODUCTION

Nowadays, studying happens more through websites where lessons and tests are online. Schools plus training programs use digital exams to check how much learners know. Still, making sure those tests stay fair and trustworthy? That's tough. Watching students during tests usually means having people supervise them in person or online. When only a few take the exam, that system holds up fairly well. Yet when numbers climb into the hundreds or more, keeping track gets messy fast. Attention fades over time. Mistakes slip through. One watcher might catch something another misses. From cameras to smart alerts, machines now spot odd actions in real time. Watching footage, they catch what might escape a person's eye. Instead of relying on someone always looking, software learns patterns over time. Without needing breaks, these tools stay alert through hours of video. Sudden movements or unusual stops get flagged automatically. Learning from data, the system grows sharper with each scene it sees. Watching how people act during tests online, this tool uses smart tech based on deep learning. Starting by spotting faces, it follows where eyes move, how often they blink, also which way the head turns. Focus matters - if attention drifts from the screen, the system notices. Instead of guessing, it checks physical cues to judge engagement throughout the exam. Starting off, software spots forbidden items during tests by recognizing objects. Instead of relying on just one method, it layers different ways to watch over things - keeping everything honest without favoring anyone. Aiming to build an artificial intelligence tool that runs smoothly on everyday devices, this study focuses on speed and simplicity. Real-time performance matters most, so the design stays lean through careful choices in structure and process. Instead of relying on powerful machines, it uses smart optimization to keep demands low. Efficiency becomes possible by cutting unnecessary layers without losing accuracy. The outcome should respond quickly, yet fit within common computing limits found in typical setups.

II. LITERATURE REVIEW

A fresh look at online exams shows scientists testing new ways to lock things down. Ways that might block cheats pop up now and then through trial runs. Some methods shift how data moves during tests. Others watch behavior like typing speed or eye movement. Each idea builds on past flaws found in digital setups. Progress comes slow but steady with each test round. Back then, exam monitoring tools mostly saved videos for later checking. Instead of real time

alerts, someone had to watch each clip after the test ended. Because of that, spotting cheating took longer. Work piled up quickly when every session needed eyes on screen. Today's cameras spot faces by themselves, thanks to smarter software built on visual learning methods. From sudden movements to still frames, tracking happens instantly when OpenCV steps into the loop. Instead of humans watching screens, machines now interpret expressions, guided by pattern-based decisions. Real-world scenes get broken down fast - each pixel examined without delay through live video streams. Using Dlib means tapping into strong tools that learn patterns in faces. Because it detects precise spots - like where eyes sit or how a nose angles - it supports deeper study of movement. From there, following eye direction becomes possible. Even guessing which way someone turns their head comes within reach. All of it runs on smart math built to adapt. What shows up as dots on screen are actually decision points shaped by training data.

A single glance is all it needs - models like YOLO process entire images at once, spotting many items without slowing down. Because of how fast they work while staying precise, these systems now help keep watch where constant attention matters. Faces plus objects get spotted together in smart exam monitoring, research shows. Still, most setups need heavy computing muscle or special gear to run. A fresh take on the problem swaps heavy design for something leaner. This setup runs live checks without needing high-end machines. Instead of complex layers, it relies on simplicity to keep pace. Speed comes not from power but smarter structure. Ordinary hardware handles the task just fine. Efficiency steps in where brute force might fail.

III. METHODOLOGY

One step at a time, the AI-powered exam supervision tool checks what test takers do while they take tests online. Each piece works on its own task, lined up like stations along an assembly line. Built this way, changes in one area won't break anything elsewhere. Parts can be swapped or upgraded later, since they don't rely too heavily on each other. From a webcam feed, video snapshots move through stages - first cleaned up, then checked for faces, scanned for actions, followed by spotting objects nearby. Meaning begins to form at each stop along the path, passed forward like a note handed quietly between workers. One phase builds on what came before without rushing ahead, details pulled out slowly, fed into what follows. Information flows not all at once but piece by piece, shaped as it moves from raw pixels toward understanding. Each module does its part while keeping space open for what comes next in line. Processed bits travel onward, never stuck, always moving under quiet logic. Nothing gets forced; everything unfolds just after its turn arrives. Steps link together not with noise but with steady rhythm beneath. Clarity grows not fast, yet surely, guided by order hidden in sequence. What enters blurry becomes defined only because earlier parts did their job well. Most times, eyes drift elsewhere - the software catches that. Talking breaks the quiet exam norm, it flags those moments too. Hands reach for hidden things, sensors notice. Light shifts might confuse older systems, here they get filtered out fast. Motion happens, sure, but only odd patterns trigger checks. False alarms fade when context proves innocent. Aim is smooth operation while things happen live, built to stay steady even under pressure. Runs fast enough that regular machines keep up, no extra gear needed just to make it go. Toughness matters here, so does speed, both packed into everyday setups without fuss.

A. Overall System Workflow

The overall workflow consists of the following sequential stages:

- 1) Live Video Acquisition
- 2) Image Preprocessing
- 3) Face Detection and Landmark Extraction
- 4) Eye Gaze Tracking and Blink Detection
- 5) Head Pose Estimation
- 6) Object Detection using YOLO
- 7) Suspicious Behavior Detection

B. Video Acquisition

A single webcam links to the machine, feeding live visuals into the setup. From there, snapshots stream without pause through OpenCV's tools. Not just snapshots, but live footage flows through the system. One after another, frames get pulled out and sent ahead - to be studied for actions and movements. The stream never pauses; each piece feeds into analysis without delay. A steady stream of frames lets the system follow how actions unfold, instead of guessing from single snapshots. Because of this, spotting unusual behavior becomes more consistent. When light levels shift, so does how well images come through. Facing the lens straight on helps the system catch details more reliably. Brightness matters a lot, yet too much glare causes trouble just as shadows do. Placement in view plays a role alongside how sharp each frame appears.

C. Image Preprocessing

Before classification, each frame undergoes preprocessing to standardize input format and improve model inference accuracy. The preprocessing pipeline includes:

- Resizing image to 224×224 pixels
 - Converting color format from BGR to RGB
 - Normalizing pixel intensity values
 - Expanding dimensions to match batch input format
- Normalization is mathematically defined as:

$$I_{norm}(x, y) = \frac{I(x, y)}{255}$$

where $I(x, y)$ represents the original pixel intensity and $I_{norm}(x, y)$ represents the normalized value.

D. Face Detection and Facial Landmark Extraction

Finding faces happens through OpenCV, working alongside Dlib's tool for spotting facial features. Tools team up here - OpenCV handles locating faces while Dlib maps key points on them.

The system identifies 68 facial landmark points on the candidate's face, representing key facial features such as:

- Eyes
- Nose
- Mouth
- Jawline

Faces shift slightly when people move them - tiny spots help catch every change through testing. Each point marks a spot that shifts just enough to show what's happening across the face during checks. Facial landmark detection might look like this when drawn out F consists of points named p one through p sixty eight F stands for the collection of facial points found, while p shows where each one sits on the face. Facing forward, the system tracks where eyes move, how often they blink, while also watching which way the head turns using those numbers later on

E. Eye Gaze Tracking and Blink Detection

Focusing on the exam screen? That's what eye tracking checks for in candidates. How they look at things gives it away - subtle but clear. Starting with facial points, the software measures space around the eyes to find a value called EAR. This number shows when someone blinks or shuts their eyelids. By tracking changes in those gaps, it knows if the eyes are open or closed. The Eye Aspect Ratio Calculation The distance between point two and six, combined with the gap from three to five, shapes part of the expression. This sum then divides by twice the span linking one and four. A ratio emerges through these paired lengths. Not every segment weighs equally here. Each measurement plays its role without extra influence. The formula holds steady regardless of how it is viewed Here p1 through p6 stand for positions of key points on the eye. If the EAR measurement drops beneath a set level, that moment counts as an eye blink. A blink gets registered once the EAR slips under its preset limit. Only when EAR goes lower than the fixed point does it mark a blink. The instant EAR dips past its baseline, blinking is detected. Blinking shows up whenever EAR sinks below a specific cutoff. A blink that comes too often might mean someone is tired. Or perhaps they are up to something odd while being watched closely. A quick shift of the eyes could signal unease. Sometimes it's just exhaustion showing through. Other times it hints at behavior worth noticing. The way a person looks away suddenly can tell more than words. Tiredness creeps into small movements like these. Suspicion grows when patterns feel off without reason.

F. Head Pose Position

Looking at key points on the face helps figure out which way the person is facing. The direction comes clear when those markers are mapped across the image. On a line through space, rotation shows up in one direction. Another path crosses it, bringing a second measure into view. A third follows, standing separate yet connected by motion

- Pitch – up and down movement
- Yaw – left and right movement
- Roll – head tilting

A slight turn of the head can tip off the software. When eyes drift sideways, attention seems elsewhere. Glancing around too much raises flags. The camera notices where a person is looking. Staring at the screen matters most. Off-

center gazes get recorded. Focus shifts are tracked frame by frame. When the head points away from the screen too long, it gets marked automatically. A glance elsewhere might seem small - yet the software notices. Over time, if facing changes stay off track, alerts follow. Not every shift matters - but lasting ones do. Direction shifts may slip by once, yet patterns raise questions. Staying turned beyond normal triggers a check later.

G. Object Detection using YOLO

A single glance is all it takes - thanks to a smart algorithm built right in. This one spots odd items around the test taker without delay. Instead of multiple passes, it works on first view alone. Detection happens fast, no second chance needed. Hidden tools show up instantly when they shouldn't be there. Bounding boxes appear alongside class chances once the picture gets split into squares. From each section, predictions form based on where objects likely sit. Grid cells guess what's inside while outlining possible shapes. Once divided, every part contributes to locating items across the frame. One way to write the detection part is O is made up of items that look like x, y, w, h, then c where x, y represent the bounding box center Measurements labeled w stand for width, while h indicates height c represents class confidence score. The model identifies objects such as:

- Mobile phones
- Books
- Electronic devices

Around each found item, a box appears - marking its place clearly. Labels then attach themselves nearby, naming what was spotted.

H. Suspicious Behavior Detection

A web of signals comes together, feeding alerts when odd patterns show up. Modules pass along their pieces - sudden shifts catch attention faster that way. Examples of suspicious behaviors include:

- Looking away from the screen frequently
- Talking or excessive mouth movement
- Presence of unauthorized objects
- Absence of face from camera

A single moment isn't enough to trigger a warning - checks spread over several glimpses confirm if something's actually there. Only then does the alert rise.

IV. RESULTS AND DISCUSSION

The proposed system was implemented using Python and tested in a simulated online examination environment. The system successfully detected:

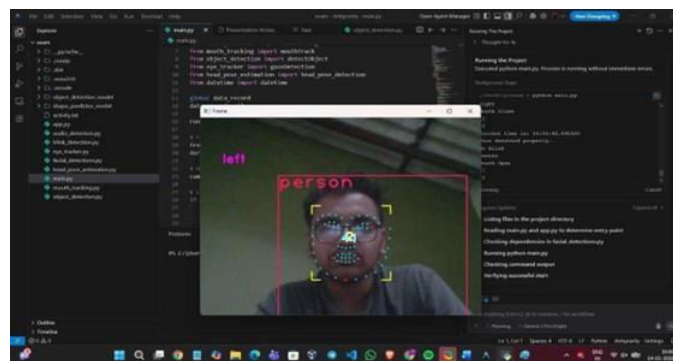


Fig. 1. Person Detection

V. CONCLUSION

A new kind of exam watch system using artificial intel-ligence helps keep online tests fair. Built on smart tools, it spots unusual behavior without slowing things down. Instead

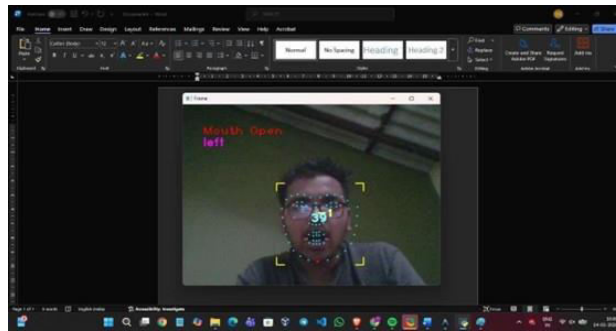


Fig. 2. Mouth Detection

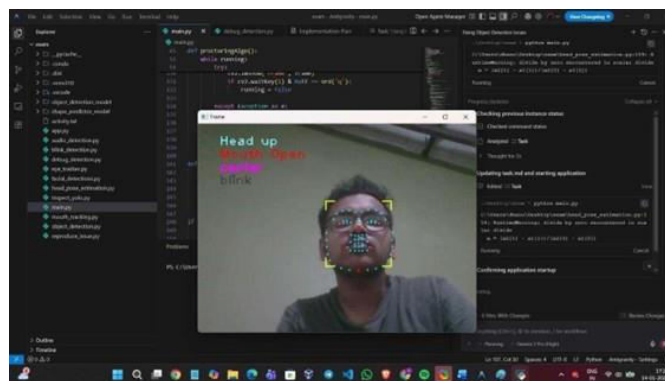


Fig. 3. Speed Breaker Detection

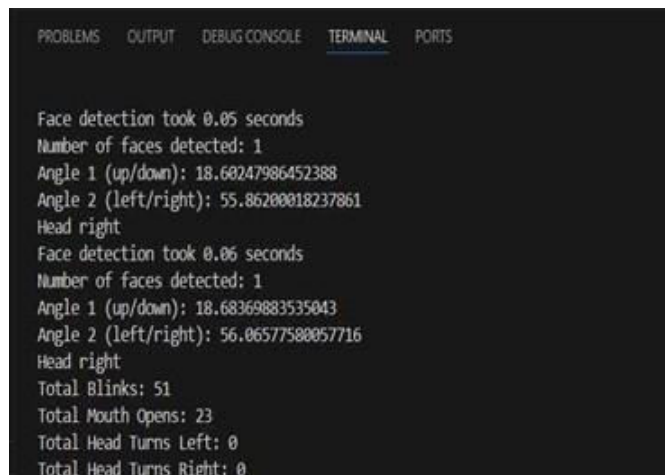


Fig. 4. Number of movement

TABLE I
PERFORMANCE EVALUATION

Parameter	Observation
Detection Accuracy	High
Response Time	1–2 seconds
False Detection Rate	Low
System Stability	Stable

of people watching every move, software checks actions in real time. This setup reduces errors while staying alert throughout each session. With constant learning, the method gets better at telling normal from suspicious activity. Facing the camera, software spots faces while following where eyes move. Instead of guessing, it checks head angles to judge attention. Watching these cues helps track what a person focuses on. Objects around them also get recognized without delay. All of this happens as events unfold. A stripped-down design runs easily on regular machines. No extra gear needed just to get it working.

Advantages

- Real-time monitoring
- Automated cheating detection
- Reduced need for human invigilators
- Disadvantages
- Lights that aren't bright enough cause problems. When illumination drops, performance dips too. Weak light throws off accuracy. Details get lost without strong visibility. Function suffers under dim settings
- Requires webcam access
- Limited training dataset

A. Future Scope

Future improvements may include Integration with cloud monitoring systems Advanced deep learning models Multi-camera monitoring Speech detection for audio monitoring

REFERENCES

- [1] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
- [2] D. King, "Dlib Machine Learning Toolkit," Journal of Machine Learning Research, 2009.
- [3] J. Redmon et al., "You Only Look Once: Unified Real-Time Object Detection," CVPR, 2016.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, 2015.
- [5] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
- [6] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2010.
- [7] A. Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks," NIPS, 2012.
- [8] K. He et al., "Deep Residual Learning for Image Recognition," CVPR, 2016.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975

Impact Factor: 9.971