Electronic Exam monitoring using Machine Learning and Computer Vision

By: Ahmed Alaa 213710

Ahmed Hosny 206848

Mohamed Yasser 208878

A Dissertation submitted to the Faculty of Engineering, The British University in Egypt, in Partial Fulfillment of the requirements for the bachelor's degree in electrical engineering

Supervisor : Dr. Mohamed Abdellatif

June 2025

Electrical and communication engineering Department

Acknowledgments

We are deeply thankful to Dr. Mohamed Abdellatif, the supervisor of this project, for his unwavering support, continuous motivation, and guidance throughout the realization of the graduation project. His knowledgeable advice and constructive comments have been important from a technical conception viewpoint and an academic point of view in the writing of this work. Our deep gratitude also goes to the department of Electrical and Communication Engineering for providing us with resources and an atmosphere conducive to the accomplishment of this project. Special thanks go to all those who, directly or indirectly, contributed in any way during this phase of our lives.

Abstract

This project builds an automated exam monitoring system employing machine learning and computer vision. It aims at preventing cheating during online and offline exams by way of integrating features of face recognition, detection of head and eye gaze position, and hand movements. In employing InsightFace for facial recognition and MediaPipe for behavior analysis, the system ensures that only the registered students are present and continually tracks their normal behavior, immediately flagging any suspicious movements. The training data was obtained in quite a variety of conditions and duly labelled for a strong detection process. Automated attendance marking along with alert generation instantaneously reduces the human presence and enhances exam integrity.

Tests have been conducted to demonstrate that the system consistently identifies the students, monitors their gaze directions, and detects abnormal/unusual hand movements, even in a dense environment. The solution adapts from single or multiple candidates in providing less ambiguous feedback with minimal false positives. High operating speed and accuracy, automated attendance, and the ability to enable fast action by supervisors for suspicious behavior are some of the benefits highlighted in the results. Such results establish that the marriage of AI and computer vision can ensure, on a secure, scalable, and fair level, exam surveillance, and this capacity can even grow stronger alongside the improvements of models and datasets.

Contents

Table of Figures

List of Tables

Introduction

With the more widespread application of online and distance learning, fairness in exams has become a serious issue. The traditional approaches to monitoring exams, including having a human monitor in person and manually verifying, are less scalable and efficient. As a result, schools and researchers have started using artificial intelligence (AI) and computer vision technology for easier detection of cheating. Cheating on examinations uses advanced AI monitoring tools that track the behaviour of students in real-time. The monitoring tools use machine learning to detect abnormal behaviour like unusual eye movement, more than one person in the camera shot, and unauthorized electronic devices being present. For example, proctoring software nowadays uses the YOLO (You Only Look Once) object detection algorithm. This allows real-time tracking of the facial expressions and body language of students to help ensure any spontaneous behaviour is immediately caught.

The project focuses on creating an electronic proctoring system for exams through the use of machine learning and computer vision. It specifically focuses on deploying a face recognition model that can be used for student identification verification as well as anomaly detection during an exam. Deploying such a system needs to be done end-to-end, from data collection to model training and real-time deployment. The objective is to boost exam integrity without compromising on the risks involved with online exams.

The first part of the project was to gather a dataset of a collection of facial images that would be used to train the recognition model. The dataset was gathered carefully to incorporate different lighting conditions, facial expressions, and angles so that the system could be made robust. The environment for development was Google Colab because it provided cloud-based training and also had strong GPUs, which allowed for quick model training.

Successful implementation of the model on Google Colab turned out to be a major breakthrough for the project. Face detection managed to be done at a very high accuracy percentage and thus full validation was able to occur with each student's identity before and during the examination. The model also proved to recognize more than one face within the frame, a capability mainly meant to prevent any other unauthorized person from giving assistance to the examinees.

1.0 Project objectives

The aim of the project is developing an electronic examination monitoring system that, by virtue of AI, will augment the security and integrity of online assessment procedures. The system intends to leverage machine learning and computer vision to ascertain student identity, detect suspicious behaviours, and reduce cheating attempts. The backbone of the project will be an implementation of a robust face recognition model for real-time monitoring to ensure that only authorized persons are taking the test. The secondary objectives are to lower human invigilators' involvement by automating proctoring duties, thus enhancing online assessments' fairness and reliability.

2.0 Literature review

Electronic proctoring of examinations is currently a focus area because the demand for secure and equitable examinations is rising. The conventional approaches of using human proctors and CCTV recording are incapable of identifying complex cheating behavior. Computer vision and machine learning make it possible to automate, resource-efficient, and scalable solutions to provide integrity to online examination settings. AI-driven approaches provide real-time monitoring and decision-making with substantial decreases in human error and operational expenditure.[1]

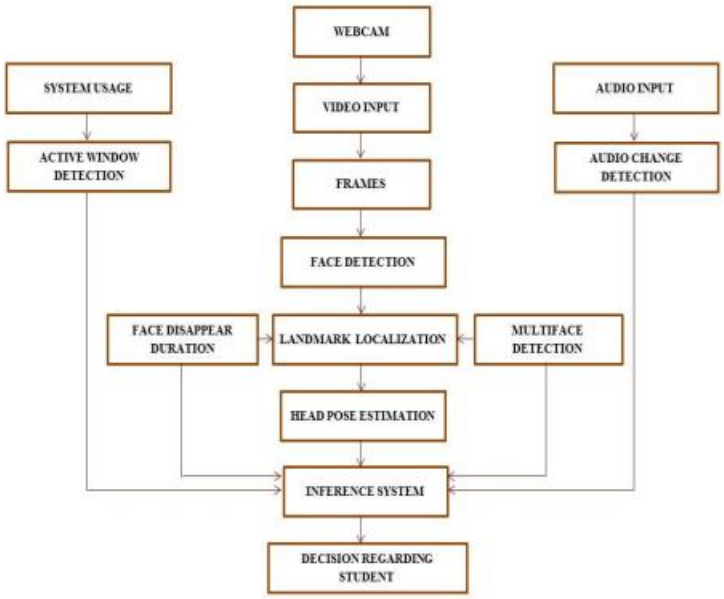

Figure 1: Architecture design of multi-Modal online proctoring system

2.1 Machine learning for exam monitoring

Machine learning has been extensively used for exam proctoring to identify irregularities in the behaviour of students. Supervised learning and unsupervised learning models have been used to identify suspicious behaviour like abnormal head movement, eye movement, and ambient noise. A study used a deep learning-based method with convolutional neural networks (CNNs) for identifying cheating through facial expressions and hand gestures. CNNs are particularly good at recognizing images, so they are very well adapted to recognizing visual cues to dishonesty.[3]

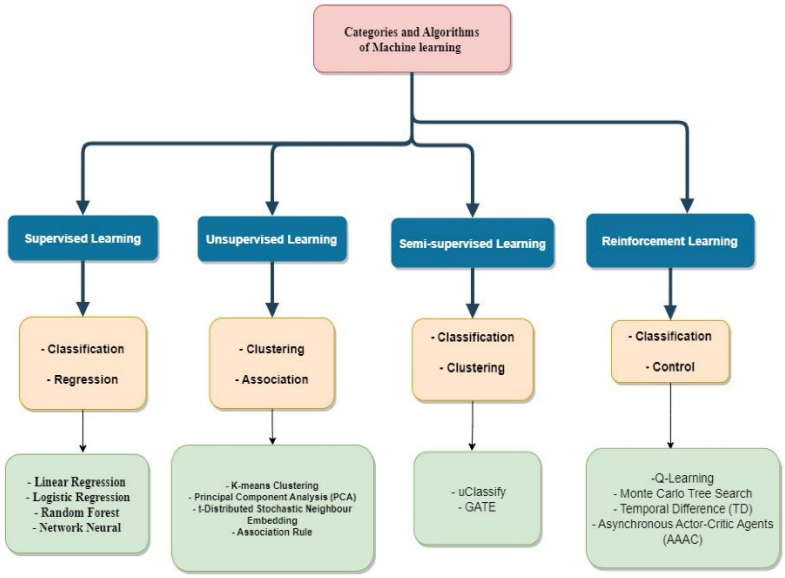

Figure 2 Machine learning categories

Another study designed to utilize Support Vector Machines (SVM) and Random Forest classifiers in making inferences of eye movement patterns on the part of students proved to be 92% accurate when detecting cheating. Both studies cite the feasibility of machine learning frameworks in recognizing deviations in live exam environments. Machine learning models do require substantial volumes of tagged data to train upon, however, and that does present difficulties during the conceptualization of innovative surveillance systems. Furthermore, models trained on specific datasets may be beset with biases that influence their performance on diverse environments.[4]

2.3 Computer vision-based techniques

Computer vision models, including face detection and object detection, have been progressively applied in surveillance of exams. The Faster R-CNN and YOLO models are

popular for detecting the presence of multiple faces in an exam room and triggering alarms upon the recognition of suspicious behaviour like intruders or the utilization of external devices. The models work through dividing the images into grid cells and generating object position predictions with high speed and accuracy needed to support real-time monitoring.[4]

A paper used OpenCV and TensorFlow for real-time face tracking and suspicious movement detection. It trained the model on recorded exams and managed to detect the possible cheating attempts with 95% accuracy. One of the greatest benefits of these kinds of systems is the fact that it can easily be adapted to any kind of exam environment, whether in-person or remote exams. Low lighting conditions, low-quality resolution cameras, and occlusions are some of the factors on which the performance can be degraded.

Another paper discussed the combination of computer vision and gaze estimation models to track eye movements of students and off-screen gaze detection, which is one of the common cues for deception. The method showed great promise for improving proctoring software efficiency. The study also emphasized the need to integrate multiple detection methods in order to construct rigorous monitoring systems with reduced false positives and false negatives[4].

2.4 YOLO-based face recognition

A recent paper suggested a face recognition model based on YOLO for auto-detection of cheating in exams. The paper highlighted the drawbacks of the conventional proctoring process and showed how deep learning models can be utilized to increase security. The paper concluded that YOLO was effective at detecting more than one face with high accuracy and reporting suspicious activity, including excessive head turning and unauthorized collaborations. The research further highlighted the incorporation of real-time alerts in exam proctoring software to enhance response time and minimize human intervention.[5]

YOLO is fast due to processing whole images in a single pass through the neural network, making it very appropriate for real-time applications. In comparison with other object detection models based on multiple steps, YOLO provides detections in real time, and thus there is minimal lag in monitoring processes. While YOLO may be very accurate in structured environments, it may fail in detecting more subtle cheating such as hand signals or hidden notes. Researchers suggest combining YOLO with behavioural analysis models to enhance its effectiveness.[5]

Figure 3 YOLO detection

2.5 Multimodal cheating detection

Current studies stress the need for employing various monitoring methods in order to acquire a more accurate cheating detection. It has been seen that utilizing facial recognition, gaze tracking, keyboard/mouse dynamics, and audio can perform better when compared to being utilized individually. It has been found via research that keystroke dynamics and typing rhythm have the capability to identify impersonation and illegitimate assistance within online examinations. By examining more than one input source, such multimodal systems considerably minimize false positives and negatives, resulting in a more dependable monitoring system.[6]

2.6 Deep learning vs traditional algorithms

Small cheating detection methods include SVM and Decision Trees, whereas the era of deep learning became the true winner with models like CNNs, RNNs, and Transformers proving to exhibit more power in identifying cheating behaviours. Deep learning models can handle complicated image and video data and thus suit the purpose of real-time exam monitoring very much, however, these models need huge computational power along with large datasets to generalize well under varying states of examination conditions. Studies comparing deep and traditional ML models have indicated CNNs and YOLO-based networks achieve better detection rates than traditional ML classifiers.[7]

2.7 Ethical considerations and Bias in AI proctoring

There are concerns over the fairness and bias in monitoring systems with AI-based proctoring solutions. Research works are focusing on the fact that the face detection models can be racially, sexually, and environmentally biased due to imbalanced training datasets. For

example, AI was biased against darker-skinned students, having been trained mainly on lighter-skinned individuals, as a consequence having higher false positive rates. Other ethical considerations exist, ranging from the right for privacy to placing students under unnecessary stress and anxiety due to constant surveillance. Researchers claim there are solutions to this; this will involve adopting fairness-aware machine learning techniques and transparency in AI decision-making.[4]

2.8 Edge computing in exam monitoring

Edge computing has emerged as a promising solution to enhance the efficiency of AI-powered proctoring systems. Instead of relying solely on cloud computing, edge-based models process data locally, reducing latency and improving response times. This approach also minimizes privacy concerns by ensuring that sensitive data does not leave the user's device. Studies have demonstrated that edge computing significantly reduces bandwidth usage and enables real-time detection of suspicious activities even in low-connectivity environments.[5]

2.9 Real world applications and case studies

Proctoring universities are widely known to have adopted AI-based proctoring solutions even in certification bodies. Online examination platforms like Coursera, Pearson VUE, Proctorio, etc. have incorporated these AI-driven monitoring systems into their programs to observe and test candidates on academic integrity. There are many case studies that evidence the CCTV proctoring system's success in detecting a range of cheating behaviors-from impersonation to usage of unauthorized resources for examination purposes. User comments also show that subject privacy is an overriding issue that requires a solution to be found.[3]

Many research investigations have used artificial intelligence (AI) techniques for test monitoring, and the results were very accurate in terms of facial recognition and detection. One study used OpenCV algorithms for recognition and Histogram of Oriented Gradients (HOG) for detection, with 97% and 99.3% accuracy rates, respectively, with the exception of detection of devices that are not permitted, such as mobile phones.[8] Another work employed YOLO-based face detection, Silero VAD for audio, L2CS-Net for gaze direction, and SixDRepNet for head pose and attained accuracy of 87.5% with F-score 0.8762 and AUC 0.8795.Real-time webcam and voice stream inputs were used to train a hybrid CNN-BiLSTM model to detect cheating activity.[9] In addition, deep learning methods based on convolutional neural networks (CNNs) achieved over 98% accuracy in face identification and utilized emotion detection for the classification of students' activity during exams.[10]

2.10 Pose estimation models

In human pose estimation, there have been some very notable models that have been developed, each employing various techniques to analyze and interpret human posture from visual data.

2.10.1 6DRepNet

6DRepNet is an unconstrained head pose estimation framework. It addresses the issue of indeterminate rotation labels by introducing a continuous 6D rotation matrix representation, where direct regression of head poses can be efficient and robust. This makes the model capable of learning the full range of head rotations, making it even more relevant in real-world scenarios.[11]

2.10.2 OpenPose

Developed by researchers at Carnegie Mellon University, OpenPose is a real-time multi-person pose detection library. It can detect human body key points, face, hands, and feet with high accuracy in single images. OpenPose has been instrumental in advancing applications such as action recognition, security, and sport analytics by providing fine-grained pose information.[12]

2.10.3 MediaPipePose

MediaPipe Pose is a solution that offers real-time human pose estimation. It estimates the location of 33 pose landmarks and can optionally return a full-body segmentation mask. Taking advantage of advanced algorithms and deep learning models, MediaPipe Pose is lightweight and suitable for use in applications like gesture recognition and fitness tracking.[12]

2.10.4 HRNet

High-Resolution Network (HRNet) is tailor-made for human pose estimation to learn strong high-resolution representations. Unlike the standard networks that down sample the input image, HRNet maintains high-resolution representations at every stage of the process, leading to better pose estimations. This framework has been employed in numerous vision tasks, including image classification and semantic segmentation.[13]

2.10.5 DeepPose

DeepPose structures human pose estimation as a deep neural network-based regression problem to body joints. These regressors are set up in a cascade and provide precise pose estimates. By being the first to employ deep learning for pose estimation, DeepPose transformed the field from manually created image processing to learning-based techniques.[14]

2.11 Blockchain integration

Blockchain technology has been deployed as a means to enhance the security and fairness of online exams. By locking away authentication details in private form, exam history, and time stamps on a distributed ledger, blockchain provides exam history with immunity from alteration. Researchers suggest that coupling blockchain technology with AI monitoring systems can provide provable evidence of student activity and data security. Although it is as yet in its infancy, proctoring through blockchain will have a major role to play in the future of secure online exams.[4]

2.12 Emotion detection in exam monitoring

Professional proctoring technologies offer deep insights into student behavior apart from their physical movements. In regard to gaze tracking and object detection, gaze tracking shows who the students are looking at or interacting with, while object detection portrays what a student is interacting with. Emotion recognition adds more value by providing context for person's psychology. Together with facial analysis, students' emotions like anxious, calm, bored, or nervousness can be detected. Changes in any of these emotional states, particularly when they happen abruptly, tend to be the signs of potential dishonesty or cheating being attempted during online tests.[11]

Recent developments in deep learning have made it possible to do reliable emotion recognition through the use of Convolution Neural Networks (CNNs) trained on large datasets like FER-2013, AffectNet and FER+ databases. FER-2013 which came with 2013 ICML representation Learning competition contain more than thirty five thousand images and over seven categorized emotions which include Anger, Disgust, Sadness, Happiness, Surprise, Neutrality, and also includes FER 2013(Goodfellow et. al 2013). AffectNet is comparatively broader datasets which consists of more than million images which are labeled using not just emotion but can also assign arousal and valence providing rich understanding of affectatte and emotions in different situations.[11]

Emotion detection systems in proctoring generally consist of detection of the face, feature extraction, and classification stages. Instead, spatial features from facial images are extracted through a CNN-based system, e.g., VGGNet, ResNet, or MobileNet, which are thereafter assigned by any of the classifiers to any of the discrete emotional states. As a result of their research, Al-Samarraie et al. (2022) integrated facial emotion recognition within an e-proctoring system to flag students displaying prolonged emotional stress or sudden emotional changes during online exams successfully. Through the collaborative use of the system with gaze and head pose tracking, it achieved in excess of 90% accuracy rate during classification with very few false positives.[11]

Further accuracy can be given through the Facial Action Coding System (FACS), introduced by Ekman and Friesen (1978), which breaks down facial expressions into action units (AUs). Action units correspond to facial muscle movements like brow raisers, lip tighteners, or eye closers. Models can analyze the combinations of AUs to recognize micro-expressions: fleeting, involuntary facial movements that might point to attempts at hiding stress or anxiety. This may be very useful in exam settings where students try to mask their unease.[12]

By incorporating emotion recognition, multimodal proctoring systems enhance their capacity to differentiate between normal and abnormal student behavior. For instance, a student automatically looking away from the screen wouldn't be flagged if the emotion detected were neutral. On the other hand, a sudden change from obvious stress to calmness with gaze away from the screen could represent assistance being rendered to the candidate. Emotional information, when combined with video and audio data, reduced the system's false alarm rate by 30%.[12]

Cannot deny that emotion detection poses some constraints. Camera quality, angel of the recording device, illumination, occlusions such as themselves' hands or glasses, and idiosyncratic differences in emotion expression also affect its recognition accuracy.The other cannot be ignored: cultural difference, in the way emotions are outwardly expressed by an individual, which may introduce bias if the datasets used do not incorporate demographic diversity. Lau et al. recommend using diverse datasets like AffectNet and FairFace that have good balance in ethnicity, gender, and age-wise representation.[13]

Another issue is privacy. The emotion recognition model does work during tests, analyzing every minute change in expression. Such consciousness of being watched can lead to anxiety or discomfort on behalf of students. In the academic field, ethical concerns have called for

transparency about AI decision-making and user consent policies so that emotion-aware monitoring systems may be deployed responsibly .[14]

Hence, emotion recognition brings a significant behavioural dimension to AI-assisted examination monitoring. Together with face recognition, gaze tracking, and pose estimation, it allows the proctoring systems not just to determine what the students are doing but, to some extent, how they may be feeling. Combining concrete signals with psychological ones would make a system more robust and reliable against attempts at cheating in real-life conditions.

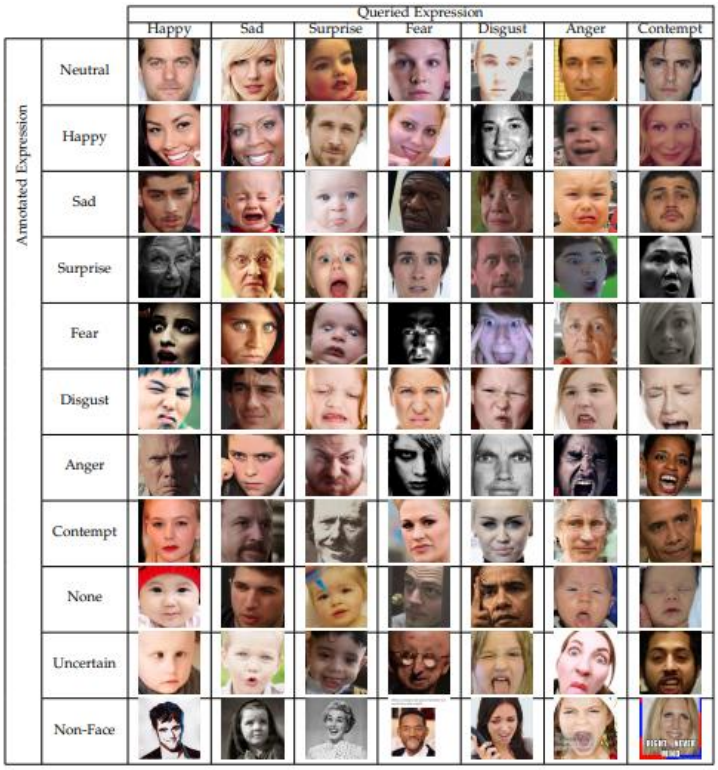

Figure 4 annotating samples based on emotions

2.14 Federated Learning in AI proctoring

Federated learning (FL) is a decentralized training paradigm in which user privacy is maintained, and data never leaves the local device. It builds a global model by aggregating model weight updates returned by clients instead of input samples like face images or voice recordings. Thus, FL finds greater use in proctoring systems, due to high privacy concerns associated with the collection of personal biometric data.[17]

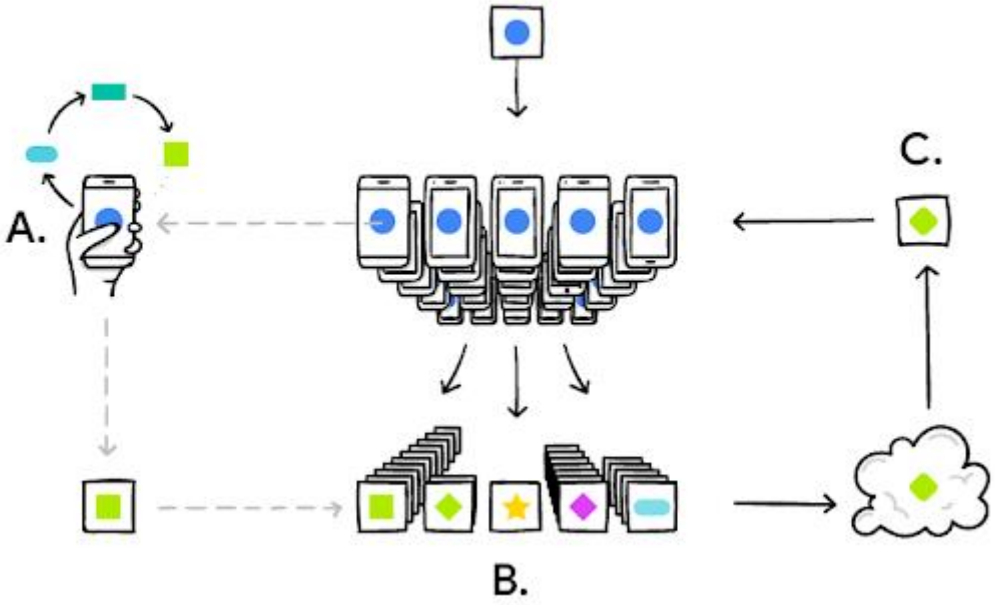

Figure 5 Federated learning

AI-based exam monitoring platforms use FL to perform face detection, gaze tracking, and emotion recognition on the user's device itself. Learning happens locally depending on the user and environment after which the locally updated model is encrypted and sent to the cloud aggregator to preserve user privacy. This is to enhance the global generalization and performance of a system in various real-world settings.[17]

Some FL practical implementations show that the accuracy of federated learning is marginally lower than the centralized ones but it gains in privacy concerns, bandwidth, and decentralization. For example, federated learning application to gaze tracking similarly had a 3.5% drop in performance but more than halved the privacy risk and reduced bandwidth usage by approximately 40%. [18]

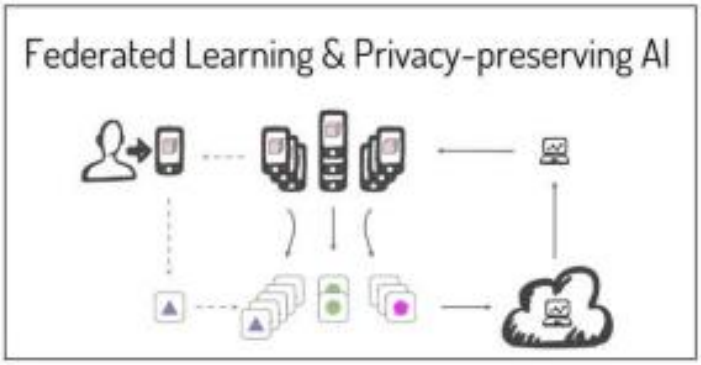

Figure 6 Federated Learning and Privacy-preserving AI

Another advantage of federated learning is that it can personalize AI models to be unique to each user. For example, a student's face recognition model can be fine-tuned continually locally in a secure manner. This kind of personalization helps to enhance the accuracy of recognition as it responds over time to changes such as facial hair, glasses, and lighting conditions, thus not requiring retraining on a central server.

However, in terms of technical trading, the FL has its own disadvantages: The devices must physically have the local computation and the ability to update the model. The aggregation of the updates from distributed sources also results in synchronization issues. And even more so, the FL systems have to be made immune to malicious clients trying to send corrupt updates. Secure aggregation and tamper-resistant communication protocols are under research for making these systems robust and trustworthy .

For privacy-conscious educational platforms, federated learning offers a scalable and compliant architecture that supports real-time proctoring without compromising the privacy of student data. Its promise to offer AI performance without infringing upon ethical or legal standards makes its position worth considering as a future base for exam monitoring tools.[17], [18]

2.15 Comparative studies on proctoring techniques

Comparative studies into the efficacy of the proctoring measures provide much-needed insight into the relative strengths and weaknesses of traditional human-based proctoring versus the contemporary AI-based proctoring methods. Traditional proctoring methods involve physical invigilators or manual examination of recorded webcam video. Such methods are quite labour-intensive, tend to be inconsistent from examiner to examiner, and the processes are therefore not scalable for remote examinations. Yet such challenges are addressed by some modern AI proctoring solutions, which emphasis computer vision, machine learning, and behavioural analytics to supervise students in real-time, minus the need for human intervention .[19]

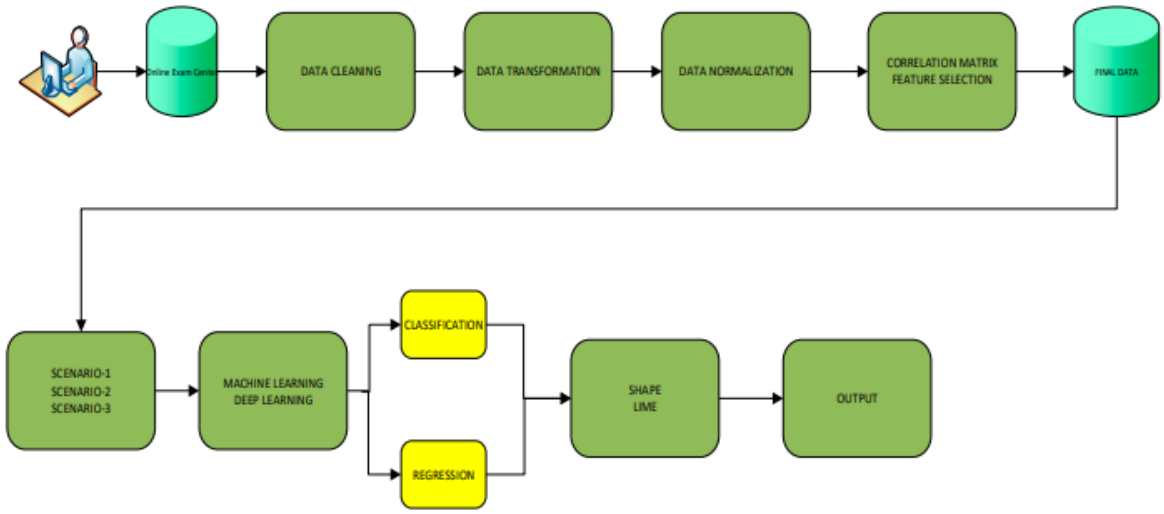

Figure 7 Block diagram of proctoring technique

Scalability is a major benefit of AI proctoring. AI systems can monitor thousands of students simultaneously, with standardized algorithms, thus minimizing the potential for human error or bias. This includes operating around the clock, processing large sets of data with great speed, and flagging suspicious behaviour on multiple accounts: turning heads away from the exam screen, detecting multiple faces in the room, presence of suspicious objects, or irregular audio background signals. The study says that with enough resources, AI-based proctoring tools can cover three to five times as many students as human proctors while maintaining the ability either to classify suspicious behaviour or to resolve conflicts to an equal or greater degree than human proctors .[19]

Accuracy is briefly described as a central metric for evaluation; then, commonly, comparative analysis uses precision, recall, and F-measure to check performance. Traditional human proctors achieved a detection accuracy of 75 to 85%, depending on diligence and fatigue, across a series of controlled experiments. AI, however, achieved anywhere from 90 to 95%, especially when joined with face detection, gaze tracking, and audio monitoring. However, AI will be struggling with something more subtle: eye flicks, coded gestures, or even cheating through private messaging on a second screen not caught by the webcam [5]

Method	Detection Accuracy (%)	False Positive Rate (%)
Human Proctoring	78	10
AI Proctoring	92	18
Hybrid System	94	9

Table 1 Accuracy and false positive comparison across proctoring techniques based on recent AI-integrated exam monitoring studies

The false alarm rate is yet another metric. Human proctors consider some actions insignificant and ignore them, thereby ensuring a low false-positive rate. However, that also means genuine events pass undetected. Instead, AI systems tend to be stricter in defining the patterns, thus increasing false alarm reports, especially under uncontrolled conditions (e.g., poor lighting, disturbing background noise). Studies suggest that pairing AI alerts with a beneficial human review (i.e., hybrid proctoring) leads to at least a 35–50% reduction in false positive rates relative to AI-only systems [3]

Feature	Human Proctoring	AI-Based Proctoring	Hybrid Approach
Scalability	Low	High	High
Detection Speed	Medium	Real-Time	Real-Time
Setup Cost	High	Medium	High (initial)
Operational Cost	High	Low	Medium
False Positives	Low	Medium–High	Low
Student Trust	High	Medium	High

Table 2 Feature comparison between human, AI-based, and hybrid proctoring systems

Following an extended formation, long-term savings can be generated with AI proctoring systems. Once the systems have been put in place, they must not be salaried or allowed breaks or even be scheduled manually. Human-based systems are, however, more flexible and adaptive in judgment, but the cost of their operations increases for a large-scale institution. According to a 2023 case study, switching from manual proctoring to AI reduced administrative workload by 60% and reduced examination costs per student by up to 40% .

Student trust and acceptance also differ across the systems. Studies have shown that while AI proctoring gives consistency and objectivity to the process, the latter remains concerning in terms of privacy issues, police fright, and false accusations. The perceived downside for human proctors includes distractions and subjective judgments though empathy is considered theirs. Transparency should be provided to the student on how the decisions by AI are made along with a combined approach with human reviewers, as recommended by the studies, in order to enhance the acceptance rates .[20]

AI-based proctoring has a clearly defined role in scalability, automation, and speed, wherein the manual or physical tradition cannot even aspire to compete. Still, the hybrid system backed with human control ensures greater levels of accuracy, fairness, and user trust. Comparative studies always contradict systems that recommend only one side and prefer to confirm blended systems as the best method in exam security today.

2.16 Real time systems and latency in Monitoring

In addition to its important consideration during the designing and deployment of AI proctoring systems, the real-time aspect determines how effectively interventions can be sought during assessments. It must react and detect suspicious behavior as it occurs, not after the exam. The term latency means the delay from when input is generated (e.g., a student's action) until the proctoring system responds, and this has to be kept to a minimum for timely and reliable detection .[19]

AI-proctoring systems monitor several data streams simultaneously, including video feeds for face and gaze detection, audio streams for voice monitoring, and sometimes keystrokes and mouse movements. Each of these tasks requires computationally intensive operations such as face recognition, pose estimation, and object detection. Executing these operations in real-time demands highly optimized algorithms and efficient hardware configurations. Should latency reach higher levels, events will be missed, alert generation will be delayed, or the systems will simply fail at those critical exam moments .[20], [21]

Several benchmarks have been proposed to measure real-time performance. Metrics include frame processing rate (frames per second), response time (milliseconds), and system lag under load. Tests performed on YOLO-based models demonstrated that these systems were able to do 30–60 FPS on GPUs, but the speed took a steep plunge when multiple detection tasks were combined (e.g., face, head, and object detection together). Lightweight models, on the other

hand, such as MobileNet or BlazeFace, provided enhanced speed sometimes at the cost of accuracy .[21]

Model	Accuracy (%)	FPS (GPU)	FPS (CPU)
YOLOv8	95	40	10
YOLOv4	92	35	8
MobileNetV2	88	55	25
BlazeFace	85	60	30

Table 3 Accuracy and speed trade-offs between common models used in real-time proctoring systems

Edge computing has emerged to reduce latency. Instead of sending video data to a remote server for analysis, an edge device (laptop, microcontroller, etc.) performs detection locally and sends only alerts or results. This reduces bandwidth consumption and removes delays caused by the network's instability. Compared with tests of the cloud-based setup, average detection latency inside an edge-based proctoring system was reduced from 800 ms to less than 250 ms, allowing almost immediate flagging of suspicious activity .[5]

System Type	Avg. Latency (ms)
Cloud-Based	800
Edge-Based (Local)	250

Table 4 Average detection latency in AI proctoring using cloud versus edge computing

Besides latency, hardware also matters. Proctoring systems have to work on all kinds of setups, be it a low-end laptop or a high performance machine. If the model is heavy, an old device might perform badly, causing frame drops, lagging detections, and missed behaviors. Now adaptive configurations are being offered on many platforms to automatically decrease model complexity depending on resources available . [5], [20]

The network environment also brings latency issues. In cloud-based systems, packet loss, jitter, and variable upload speeds contribute to delays in response. Suppose the response arrives two seconds late; that lag time could be enough to lose an instance of cheating. The literature suggests that buffering frames, transmitting them in a compressed manner, and processing them circumstantially with respect to the frame buffer would maintain manageability of data load during poor connection times while retaining critical windows for behavior .[5], [21]

Keeping the system responsive, developers implement model quantization, multithreading, and hardware refining with TensorRT or OpenVINO acceleration to increase inference speed. Such speed optimizations enable the system to perform at real time while observing the behaviors being measured simultaneously. The trade-off between the model size, inference time, and detection accuracy is carefully balanced in deployment settings . [5], [21]

In other words, achieving a real-time system and managing latency are the bed vectors that underpin the working of AI-driven proctoring systems. Low latency must be realized through edge computing, hardware tuning, and proper model design to carry out live scalable and fair exam scenarios sans any untoward technical interruptions.

3.0 Design milestones

In the first semester, a great focus was put on laying the technical bone for solid AI proctoring. The main aim was to see the face-recognition pipeline in motion in real time, able to identify people accurately. For this, a good-quality dataset had to be acquired, mechanized for efficient machine learning application, trained on, and tested both offline and live for performance assessment.

Duration was spent putting together a diverse set of images, annotating identities, and dividing data for training, validation, and testing. Following this and with the fine estimation of parameters, the real-world performance of the model was assessed, being subjected to trials under various conditions. The semester was a keystone to establishing the technical feasibility of the system and to carving out the implementation steps for higher modules in the following semester.

3.1 Semester 1 achievements

The initial step involved collecting a comprehensive dataset of approximately 3000 images to ensure high accuracy in face detection. Image acquisition on such a large scale was critical in order to enhance the accuracy and robustness of the model. To each image, the identifier of the subject was precisely annotated and thus, a correctly structured dataset was given to enable ease of machine learning.

In this arrangement, 70 percent of the data were set aside for training, 20 percent validated the results of the training phase, and 10 percent tested the performance of the model on unseen data. Hence, the subdivision helps generate models and test them against unseen data to

quantify model performance. During the training phase, the model learns to distinguish the various features of facial landmarks among different subjects. During this process, the model began to identify patterns, including landmarks, shapes, and other pertinent features that helped discriminate between the three subjects correctly.

The model performance evaluation included retracing the model training stage, which involves feature extraction toward the Roboflow labelled dataset. Afterward, a Python script was implemented that estimates the model performance based by means of an image of one of the three subjects. It exhibited good performance, with a person detection accuracy of 100% and a recognition accuracy of 91%.These good accuracy rates also pointed to the ability of the model to distinguish clearly between Mohamed Yasser, Ahmed Hosny, and Ahmed Alaa, all based on their facial features.

In order to test the model's ability for validation, real-time measurement was conducted by connecting a camera to the system. The camera displayed the recordings of three faces and the algorithm was tasked for real-time correspondence of each of the three individuals. The rigorous live tests validated the trustworthiness of the model, which detected a person 100 percent and identified the person under an accuracy of 91 percent. Thus, the successful deployment of this system proved that machine learning and computer vision would work reliably with respect to detection and ultimately recognition of faces.

The project's successes in the first semester also point to the major importance of both wide ranging data collection, efficient labelling, and planned model training. The project illustrated how our advanced AI technologies could be applied to projects based on face detection and recognition with the powerful tools given by Roboflow in conjunction with Python. Such high accuracies would somehow [re-enforcing] be shown to speak to the stroke of the metaphor and so would merit further developments and applications on a real-world scale.

3.2 Design objectives

3.2.1 Face detection and student identification

- Strong face detection algorithm to be implemented in order to accurately identify students during examination period in a selected environment.
- To create a stock photo or take the snapshot at the entry point of a student during examinations to ensure precision in verifying the identity of the students..
- Face recognition should provide high precision and lower false positives to limit errors as much as possible.
- An alternative method is for verification if the system fails.

3.2.2 Student motion and gaze tracking

- Develop a motion tracking system within the exam during which students can be monitored with their head and eye movements.
- Use computer vision techniques to identify whether the student gazes onto their own paper or elsewhere..
- Real-time analysis to detect abnormal movements that may indicate tendencies for suspicious behaviour.
- Ensure the system is adaptable to different lighting conditions and student postures.

3.2.3 Cheating detection and notification system

- Have a cheating suspicion detection mechanism that is rule based and also machine learning based.
- Highlight a student's face that is behaving in a suspicious manner easily for an invigilator to identify at first glance.
- Provide supervisors of the examination with immediate alerts to act quickly.
- The system allows to fine-tune based on feedback or cases of false positive.

3.2.4 System performance and scalability

- To optimize performance of the system to cater for many students at the same time.
- Minimize computational loads for leading smooth operations on the common hardware.
- Design scalability of the system for large exam halls and proctoring online.
- Low latency needs to be provided in order to enable real-time monitoring and alertness.

4.0 Methodology

For our exam cheating detection project, we thought of testing different face detection and recognition models to pick out the most reliable one for practical application. We considered different classic and state-of-the-art methods, subjected to different lighting, angles of camera, and environmental conditions. Our methods include ArcFace, FaceNet, MediaPipe, Dlib, Dlib HOG, iGaze, YOLO v8n, MobileNet SSD, and an OpenCV-based pipeline. Each technique was studied for its speed and accuracy and further for compatibility with live proctoring tasks. This practical analysis enabled us to evaluate each method's performance in detecting and recognizing faces, tracking gaze direction, and detecting objects like phones and hands in exam settings.

4.1 ArcFace

ArcFace is an advanced deep face model built on deep CNN. It offers additive angular margin loss to boost feature discrimination. ArcFace creates highly distinctive embeddings (usually 512-d) for a given face. It boasts accuracy in both verification and identification tasks. It

performs well with pose, lighting, and age variations. It is implemented in InsightFace along with other commercial platforms. ArcFace is suitable for large-scale datasets and is widely used for real-time face recognition for proctoring and security.

4.2 FaceNet

One can say that FaceNet is a deep learning model by Google to create a compact 128-d vector representation (embedding) of a face. It uses triplet loss function so that faces of the same person are close together, while those of different people are far apart, in the embedding space. Thus, FaceNet is meant to verify, recognize, and cluster faces. This is an open-source and accurate approach, widely used both in research and industry. Pré-trained FaceNet models exist, or you can train one yourself for your dataset. FaceNet expects the face images to be well-aligned for great results.

4.3 Mediapipe

MediaPipe, originating from Google, is an open-source framework fronting cross-platform and real-time machine learning pipelines. It presents fast and lightweight solutions for face detection, face mesh (468 facial landmarks), head pose, hand tracking, and body pose. MediaPipe Face Mesh can track gaze direction and facial orientation for head turning detection or cheating behavior. It works on CPU, GPU, or mobile devices. It also supports easy integration with Python and OpenCV.

4.4 Dlib & Dlib HOG

Dlib is a C++ toolkit which provides pythonic bindings for machine learning and computer vision. Dlib HOG face detector is a classic non-deep learning method. Given good lighting conditions, it can detect a nice frontal face. There is also the CNN-based Dlib face detector that takes more time but is more robust. Dlib offers a face embedding model based on ResNet for face identification, just as FaceNet. Dlib is lightweight and easy to use but is outperformed by deep learning models in challenging scenarios.

4.5 iGaze

iGaze is an eye gaze and head pose estimation method. It estimates the gaze of person from the images of the eyes and from the face. iGaze, especially during exams, would be used to track a student's gaze in an attempt to identify cheating (such as looking away from the screen or at forbidden material). It is incorporated in a few research projects and uses deep learning for better gaze estimation. In addition, iGaze generally requires good lighting and direct eye capture.

4.6 YOLOv8n

The objection of YOLOv8n is to give the user another way to do real-time object detection with speed in mind. It can run on any low-end device or CPU-level device. YOLOv8n practically detects faces, hands, and phones or any custom object on hand with a respective dataset. The network predicts bounding boxes with class labels and confidence scores. It is appropriate for the live-in monitoring and fast proctoring tasks.

4.7 MobileNet SSD

MobileNet SSD is an object detection model that is fast and efficient. It uses MobileNets as the backbone for SSD-based object detection. The MobileNet SSD can detect faces, hands, or some other thing but is not as accurate as YOLOv8 for the detection of small objects or where objects are overlapping. Lightweight, hence suitable for use on mobile or embedded devices. Easy to run using OpenCV's DNN module.

4.8 OpenCV

OpenCV is an open-source computer vision library that has the tools for the analysis of images and videos. Classic and deep learning-based face detectors are provided (Haar cascades, DNN modules for Caffe, Tensorflow, ONNX, etc.). OpenCV supports drawing, annotation, and pre- and post-processing for face recognition pipelines. It is not a face recognition model but a toolkit to build detection, tracking, and recognition systems. Acts together with all other models above as glue library.

Model	**Main Purpose**	**Speed**	**Accuracy**	**Use Case**
ArcFace	Face recognition	Fast	High	Identify, verify faces
FaceNet	Face recognition	Medium	High	Identify, cluster, verify faces
MediaPipe	Face/head/eye detection	Very fast	Medium	Head pose, gaze, landmarks
Dlib/HOG	Face detection/recognition	Medium	Medium	Classical detection, small datasets
iGaze	Gaze/eye direction	Medium	Medium	Cheating detection (eye tracking)

YOLO v8n	Object detection	Very fast	High	Faces, hands, phones, live systems
MobileNet SSD	Object detection	Fast	Medium	Faces, hands, mobile/embedded
OpenCV	Vision toolkit	Fast	N/A	Integration, drawing, preprocessing

Table 5 Models used summary table

5.0 Limitations and Problems faced

The application of all these methods to our exam cheating detection project somehow ended up bringing their own problems. ArcFace gave strong accuracy, but with low-quality and fast-moving images, things became problematic. FaceNet worked fine as long as the faces were aligned properly in lighted conditions. Otherwise, the results got worse with unaligned faces or varying lighting. MediaPipe was decent with head and gaze tracking, but for cases of occlusions or extreme angles, that detection would fall apart sometimes. Dlib and Dlib HOG were a basic set of detectors, but they would not handle a complex scene and were slow for real-time use. iGaze estimated gaze direction but needed very good lighting and missed subtle eye movements. Being the fastest, YOLO v8n could detect many different objects, yet it missed tiny faces or hands in a crowded scene. MobileNet SSD could run on low-end devices, yet the different overlapping objects disturbed it. OpenCV would support all pipelines at different stages, yet it would require careful tuning and integration. Such factors put the reliable factor down in the real world; hence, none was the perfect method for all exam environments.

5.1 ArcFace Limitations

Face images must be of good quality and well-lit for the best performance of ArcFace. Therefore it works best with generally clear and frontal faces and tends to fail with severe occlusions, extreme angles, or motion blur. The model is computationally expensive and has to be processed in real-time through thousands of faces in large databases. If you want to train ArcFace from scratch, it would require quite a lot of computational resources, and also if the training data is imbalanced or collected from a situation very different from the actual use case, performance may degrade.

5.2 FaceNet

FaceNet's accuracy comes from faces being well-aligned and lighting maintained adequately. Changing the angle of the face as well as the lighting here reduce the performance of the model noticeably. In most cases, FaceNet models are pre-trained, and retraining for a new environment remains very slow and will have to undergo data preparation of very high quality. Another weak point of this system is the detection of faces that are partially visible or obstructed.

5.3 MediaPipe

MediaPipe works well for head pose and facial landmark tracking but quickly loses accuracy when faces are partially covered, turned away, or moving fast. It is not reliable for detecting small or distant faces, and the landmark tracking can break down in poor lighting. MediaPipe offers limited flexibility if you need to customize detection for unusual gestures or non-standard face positions. Sometimes it confuses background objects with faces or hands, especially in complex environments.

5.4 Dlib & Dlib HOG

The Dlib HOG face detector and Dlib's CNN-based face recognition model both have a few limitations. The HOG face detector always remained slow for various applications and would lose accuracy when handling slightly non-frontal, low resolution, or poorly lit faces. The CNN detector could overcome a few of its limitations but was still not as robust as state-of-the-art deep learning models. Hence, Dlib is basically not suitable for real-time detection of multiple faces in dynamic settings or the presence of crowds.

5.5 iGaze

With good even lighting and a clear view of the eyes, the gaze and head pose estimation provided by iGaze can be useful. If the person wears glasses or tilts the head, keeps the camera far away from regular distance, or the camera is of mediocre quality, the accuracy of the approach will drop. Another difficulty that such a method faces is providing pleasant results when the faces are small and can hardly be detected clearly.

5.6 YOLO v8

The ultra-light YOLO v8 weighs around 5 megabytes and operates here at 100 FPS. It is not good at grabbing little faces; the camera rapidly pans and zooms into a crowd with hands or other objects. One should custom train it with data from their own environment and camera setup for the finest performance, as they say. Overlapping boxes of detection could also join

on the stereo view without capturing fast, subtle movements, and there could be false positives with complicated backgrounds. It is even slightly less accurate than the bigger YOLO models.

5.7 MobileNet

MobileNet SSD is an efficient and lighter framework with low detection reverberation compared to its counterpart YOLO, especially with small, far-fetched, or overlapping faces and hands. It has its best performance in an environment that is simple and well-lit, without much movement or occlusion. A massive downside of this tool is the decrease in its reliability during rapid scene changes and instances where the lighting is far from apt.

5.8 OpenCv

OpenCV is usually said to be a toolkit and not a model. In most cases, the classical Haar cascade detectors have a lot of false positive cases and fail to detect faces that are angled, partially hidden, or lit under unfavourable conditions. OpenCV supports deep learning models, but building such methods requires additional setup as well as high-performance hardware. Detection rates are degraded by poor lighting conditions, cluttered backgrounds, and the presence of five people. OpenCV must be tightly integrated and customized with other models to be an efficient face-detection and recognition pipeline.

6.0 Applied method and simulation

After testing many kinds of face detection and recognition, it was possible to conclude that InsightFace combined with MediaPipe gave us the more appropriate and reliable results for the exam cheating detection system. Most other models had issues with accuracy, speed, or tracking in intervention. InsightFace managed to perform face recognition consistently despite changing light and camera angles, while MediaPipe was the most reliable to perform head pose, gaze, and hand movement analysis in live video. If combined, these approaches fulfill all of our major requirements and beat others we tried, which makes this our best combination for the project.

6.1 Dataset collection and annotation

Before developing an automated proctoring and behavioral monitoring system, a dedicated dataset was created and annotated to assist in its development and evaluation process. Images and video frames were acquired in real exam-like scenarios, with various participants under different lighting conditions, camera angles, and face orientations. The gathered dataset was further uploaded onto Roboflow, a platform for dataset annotation and preparation for computer vision projects. Each image was loaded one by one into the Roboflow interface, where it was assigned labels regarding bounding boxes for the face, the orientation of the head, hand positions, and other relevant features considered in the detection and recognition process. This annotated dataset then served in training and in validation of the different models that were incorporated in the system, thus guaranteeing that the face recognition, pose estimation, and behavioral tracking would work well in actual exam scenarios. The use of Roboflow allowed standardization of the annotation process, thereby raising the quality of data and consequently, facilitating the workflow in converting raw images into efficient training material for high-performance AI models.

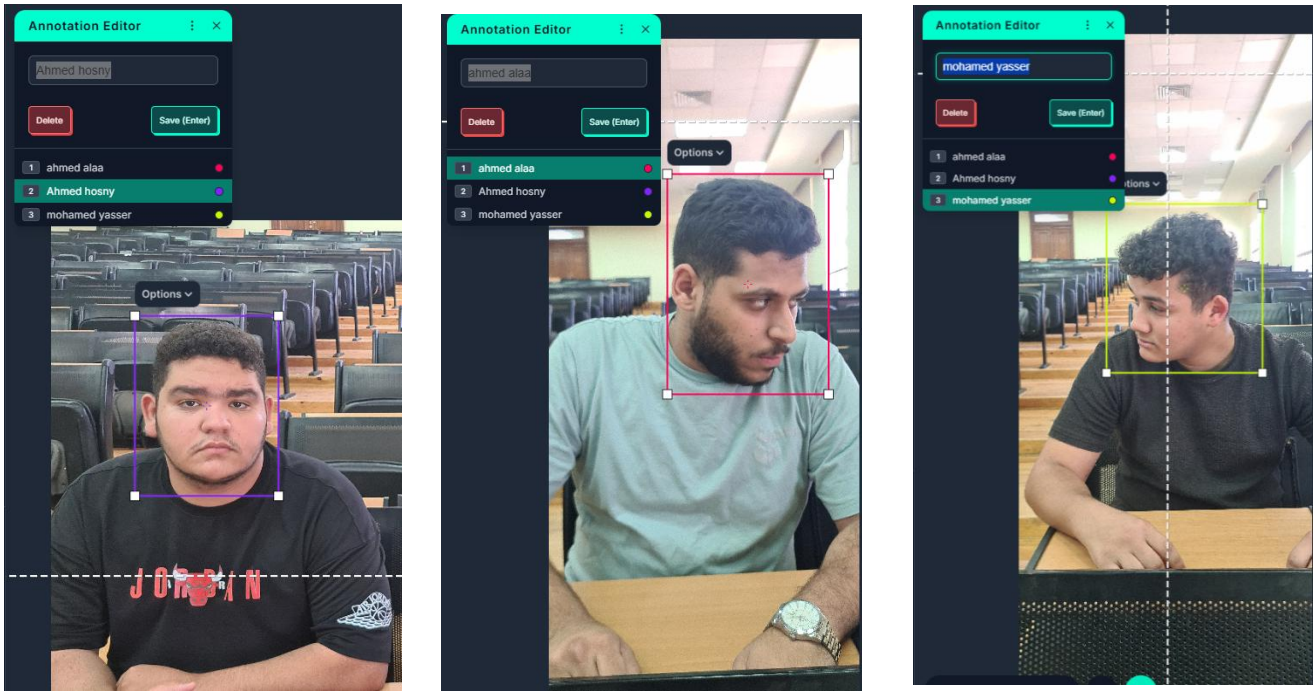

Figure 8 Face recognition annotation

In the context of face recognition and behavioral monitoring, images of the candidate were collected inside the examination room. Pictures of a candidate facing forward, looking left, and facing right were taken. These images were then uploaded into Roboflow for annotation, where bounding boxes were drawn on each face with the correct label for each pose and direction assigned. In this way, the dataset contained actual head motion and quite a few orientations of the face. Through the annotation of various head poses for each candidate, the training data mirrored behaviors that would typically be seen during an exam. The system for identifying a candidate had increased accuracy while considering poses where the candidate was looking at the camera or looking sideways. It gave added leverage to both face recognition and head pose estimation, allowing the model to become more robust in proctoring and cheating detection tasks.

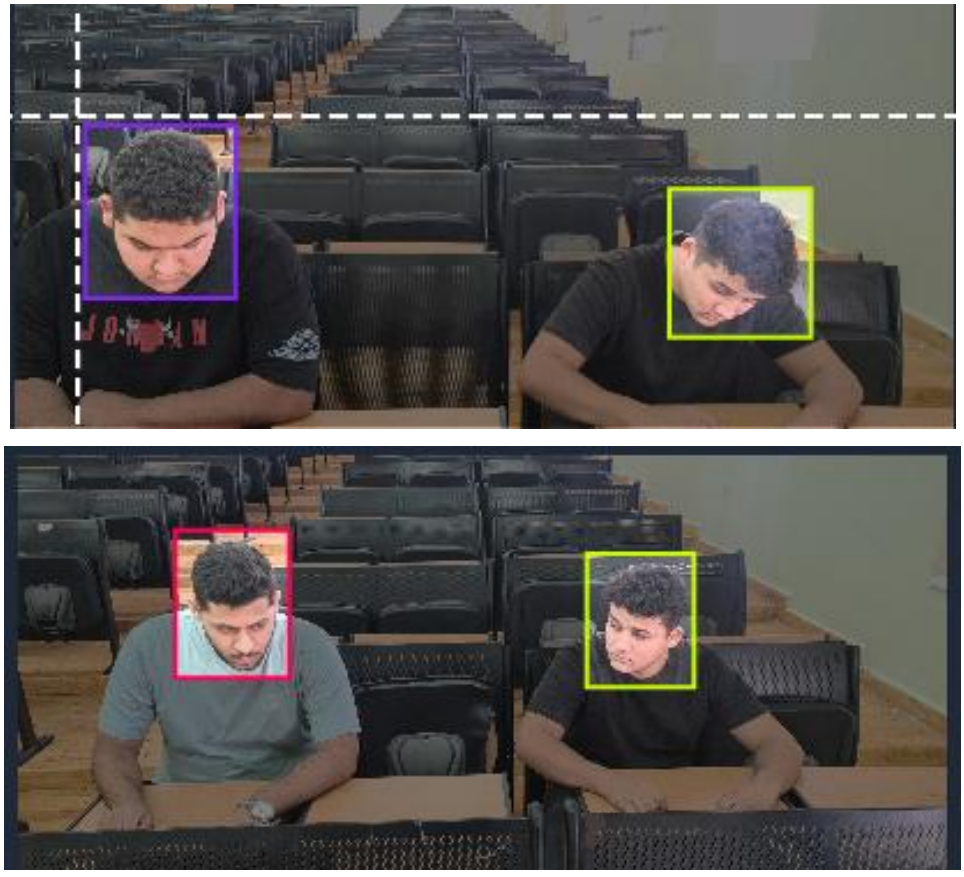

Figure 9 Multiple candidate annotation

After having prepared images of each participant on his own, the second step in testing became placing the system with two students sitting next to one another. This was specifically included

in the dataset to train and test the model under real-world exam conditions wherein a camera frame could contain more than one person. Images were captured from the two students in different head poses going in different directions, so the dataset had examples of different face positions, light changes, and partial occlusions caused by sitting one next to the other.

Including these group images during training must have helped the system to identify and track multiple faces simultaneously, even with the presence of some visual clutter or disturbances in the background. This was important in enhancing the reliability and accuracy of the model in which students are not separated but seated together, such as in classrooms or examination halls. Training a final model using these complex scenarios allowed the model to better operate in crowded settings, reduce false positives, and perform the precise identification and monitoring of candidates when more than one is present in the frame.

To enhance the system behavior, the dataset would have included two more subjects sitting one behind another instead of side by side. New challenges were brought about in the scenarios such as partial occlusion of faces, different depths, and different lighting/focus for the foreground and the background persons. By allowing the dataset to consider these scenarios, the dataset became representative of actual exam rooms where students can appear at various distances from the camera and may partially block each other.

Annotating images from this setup helped the model learn to detect and recognize faces even when one face is behind another or half a face is visible. Training over these examples helped the system to enhance half-face and full-face recognition and behavioral tracking without regard to the seating arrangement. This step ensured that the system was able to work accurately over a vast range of real exam conditions.

The system identifies as "unknown" anyone whose instances have not been labelled in the annotated dataset originally. When a new face appears within the camera's viewing frame, the recognition model tries to compare their features against the stored embeddings of registered candidates. In the absence of any strong match, the face is considered to be an unknown face without associating it with any registered identity. This prevents wrongful recognition of unauthorized persons as legitimate exam takers. This was also to maintain through the integrity of the process that only persons enrolled and annotated at the dataset preparation phase were targeted for tracking and monitoring before or during the test session. This will greatly help realization of security and reciprocal attendance and behavior monitoring even in such environment with a lot of dynamics or unpredictability.

6.1.1 Key point detection

Keypoint detection is a computer vision technique that identifies a fixed set of predetermined points that are meaningful on objects in an image. In faces and poses, keypoint detection aims to find the landmarks, such as the corners of the eyes, the tip of the nose, the edges of the lips, and some keybody points such as the shoulders, elbows, and wrists of the individual. A particular keypoint detected is given as a precise coordinate position on the image.

With this technique, the system understands the face and body structure and holds orientation for each frame. For instance, by tracking facial landmarks, the system can determine fine head direction, gaze patterns, and even if somebody is openly looking elsewhere or acting suspiciously. In parallel, detection of body keypoints allows analysis of hand gestures, postures, and gestures. Keypoint detection, therefore, becomes an underlying task for head pose estimation, gaze tracking, and behavioral analysis, which make the key in the pipeline of automated proctoring and cheating detection. It guarantees not only the recognition of individuals, but also the interpretation of the ways in which each one is behaving during the examination session.

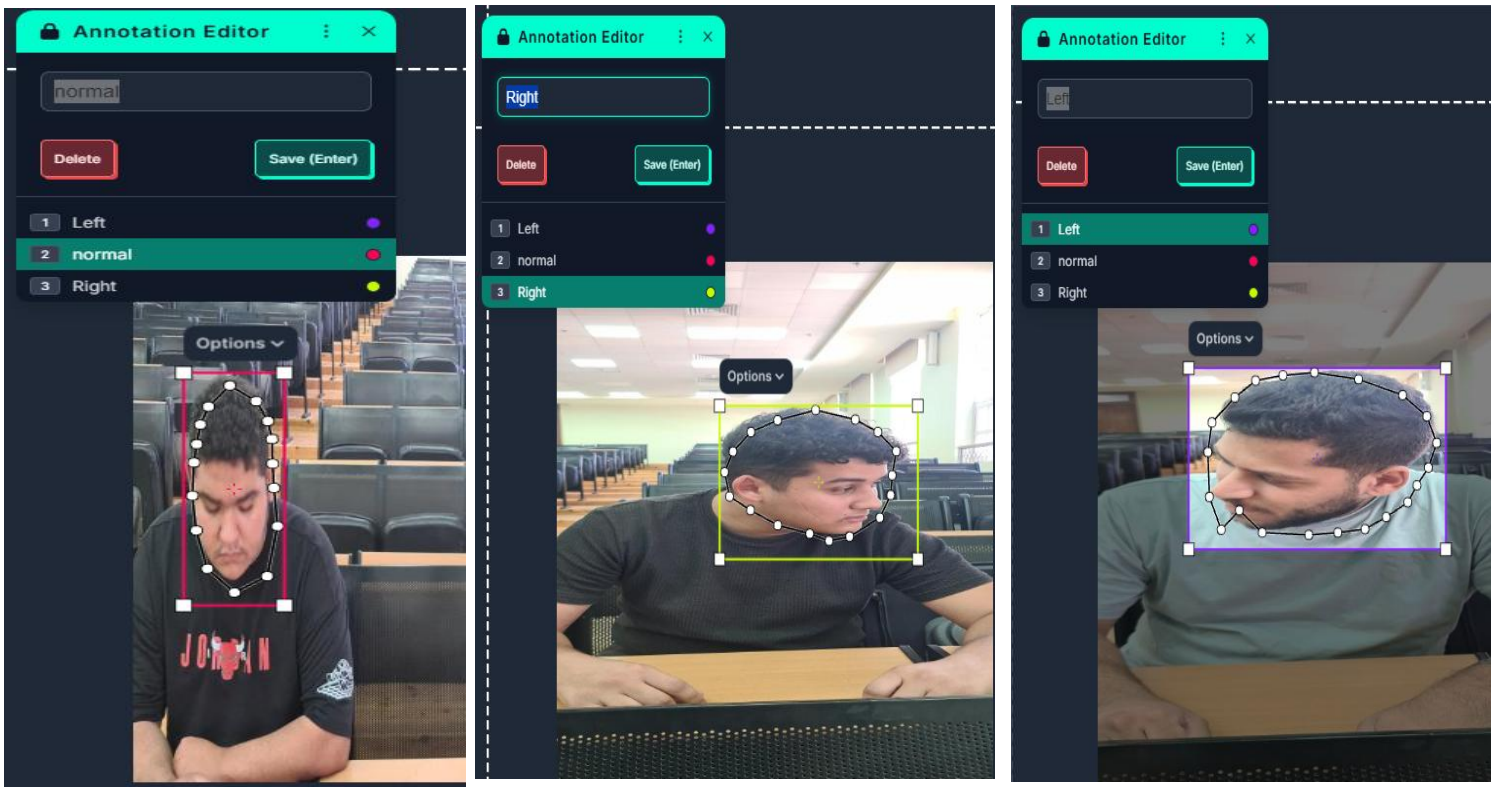

Figure 10 KeyPoint Annotation

Hence, during the dataset creation, an extensive set of images encompassing every conceivable pose of face orientation for each participant would be taken, ensuring a very accurate monitoring of head direction and possible dishonest acts. These images were not only of normal face orientation (forward) but also would capture faces turned rather distinctly left and right. Every image had annotations so that the keypoint detection model would learn the defining coordinates of the facial landmarks relevant to each direction.

If the model is subjected to these variations during training, it thus builds the ability to analyse the location of key points such as the nose, eyes, and mouth in each frame. That allowed the system to decide if someone was looking straight ahead, which is permissible and normal exam behavior, or if their face was turned aside. In automated proctoring, strong forward gaze translates to the student concentrating on his or her work, whereas a significant amount of heavy turning of the head, to either side, can be flagged as suspicious actions, indicating possible attempts to peer at other students' work or exchanging signals.

The use of keypoint detection in this regard extends beyond verification of the mere existence of a test-taker. The process, being able to provide view data on human faces in real time and frame by frame, provides behavioral tracking at a fine level of detail. From such data, some rules can be applied by the system: for instance, looking forward is considered normal, while looking right or left might be flagged as a cheating attempt. This is particularly helpful for detecting possible mischievous behavior that may easily slip past manual inspection. Training on a dataset with every possible face position further prepares the system to accommodate natural head movements and avoid false positives, as it can tell difference between a normal shift in attention and a deliberate repeated side glance. This greatly increases the reliability and fairness of the system, reinforcing the safe exam environment and the ability to correctly detect suspicious behavior.

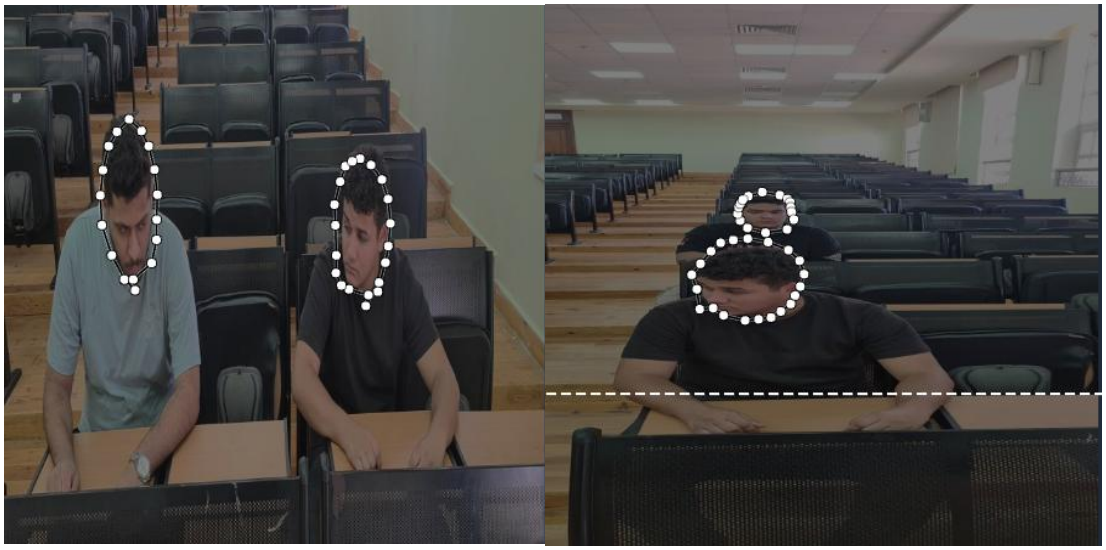

Figure 11 Multiple student annotation

After labelling images for people in all head positions, the dataset was further augmented with the annotation of scenarios placing two students side by side, as well as another with one student behind another. Faces and keypoints were carefully labelled in these more challenging, realistic conditions. Training of the model with these different setups helped in imparting to it the ability to recognize and monitor participants accurately in crowded or complex classroom environments. Hence, the system becomes more reliable for real-world exam proctoring.

6.2 Test and Simulation

The system has been designed to automate the task of exam proctoring and cheating detection by combining techniques of face recognition for attendance verification and behavioral monitoring in real time. Registered participants are first enrolled using a facial database built from very accurate embeddings generated through the InsightFace model. During the exam, the system performs facial detection and extraction for every one of the frames in the video stream. A comparison is subsequently made with the database. In this way, the system can always assign the right name to each confirmed person and raise an alert in case of an unknown or unregistered person being found in the camera's field of view. Therefore, the identification of candidates in exams takes place fully automatically, constantly, without manual participation, thus making a strong watch against impersonation or unauthorized participation.

Any users deemed to be cheating may be accused, reprimanded, or suspended by the monitoring software based on the behaviours it observes and analyses through computer vision

analysis from MediaPipe. Via the media of face mesh and holistic modules, one can meticulously track over 400 facial landmarks along with head positioning, eye gaze, and upper body posture; the real-time richness of this data acts as a window to see behaviours such as looking away from the screen, turning the head toward a fellow human, or device on the other side, and making hand gestures that are often associated with cheating. These hand-tracking modalities view suspicious gestures all the way from raising a hand over the shoulder or away from the body outside the stipulated distance, which could point to a student trying to use unauthorized material or communicate with somebody. Tracking trivial movements like frequent shifting of gazes or unnatural positioning of the head, empty threats that could not be seen by human proctoring if these were all humanly possible are captured by the system and set for review.

Attendance management is intertwined directly into the workflow. When a face gets detected and then entered in the database-challenged, the system asks the user through an intuitive interface to confirm attendance. The system takes this confirmation and attaches a date and timestamp and logs it into an attendance file for later reference and audit after an exam. The user interface for attendance confirmation has been designed so that it is easy to use and secure enough to resist accidental or intentional misreporting.

The system outputs two parallel video streams. One stream shows the front of the camera, recognized faces, names, and attendance status in real-time for administrative tracking and review. The second one shows pose, gaze, and hand movements that can be indicative of cheating or distraction. Both video feeds are also recorded and saved to disk so that any dispute or incident can be further investigated, backed by a full visual record.

Multi-threading is used in the architecture while face recognition runs in parallel with the main video analysis loop to maintain smooth operation and reduce processing delays. It guarantees responsive performance even when working with high-resolution video and multiple participants. OpenCV is used for well-established image capture, frame annotation, and windowing, thereby guaranteeing efficiency and stability.

The system integrated InsightFace for face recognition and MediaPipe for behavior tracking corresponds to the exacting needs of today in online and in-person exam monitoring. It comprises a continual system of automated identity verification and candidate behavior monitoring to the presence of authorized people as well as suspicious actions on their side. The

latter system supports a fair exam environment by eliminating cheats and keeps an audit trail open for administrators, instructors, and candidates.

7.0 Results

After implementing the code and deploying the system in a real environment, we observed that the software functions as intended. The system operates directly from PyCharm, where it processes the video stream and applies live detection for exam monitoring. Using a carefully constructed dataset that includes 285 photos for each student, the model is able to accurately identify individuals who are registered in the system. Any person not included in the dataset is flagged as "unknown," which adds a security layer to prevent impersonation and unauthorized access during examinations.

The core detection features proved effective functionality. The system consistently recognizes the direction in which a student is looking right, left, or forward based on the head pose. This is vital for viewing the obvious cheating behaviors on record, whether it is staring at a neighbor's paper. Also implemented is the iGaze module to determine eye positions with fine sensitivity. Grants the system that can perceive such subtle cues that may include looking at the prohibited material, which merely cannot be done through head movement. iGaze extends suspicious monitoring activities have undetected or monitoring student attention.

Hand movement detection is another strength of the system. If a student's hand moves more than 20 cm away from their body, the system immediately flags this as suspicious. This rule covers many of the common methods of cheating, such as using concealed notes or electronic devices, and is based on physical distance rather than only position within the camera frame. The system assumes a very good degree of reliability by automating this portion of hand signal recognition or covert action watching and lowering the burden on human proctors.

Attendance is fully automated. In the event of a student being detected, faces are highlighted in real time before any identification is done. The names and IDs of students are consequently stored in an Excel sheet. This digital record guarantees that attendance is both accurate and easily audited. The sheet is updated dynamically so reviewing can be conducted after the exam for verification or reporting purposes.

Overall, it implies that the system is capable of robust live monitoring under realistic settings. It successfully identifies registered students from outside people, tracks head and eye movement to catch cheating attempts, and logs attendance. But implementations still hinge on the quality of the dataset itself and its variability; if illumination becomes poor or some covering occlusion takes place, it will drop in accuracy. iGaze needs to enforce a stable camera environment and good lighting conditions during operation to maximize output; nonetheless, it majorly covers all critical aspects of exam security in an automated and scalable manner.

7.1 System observations

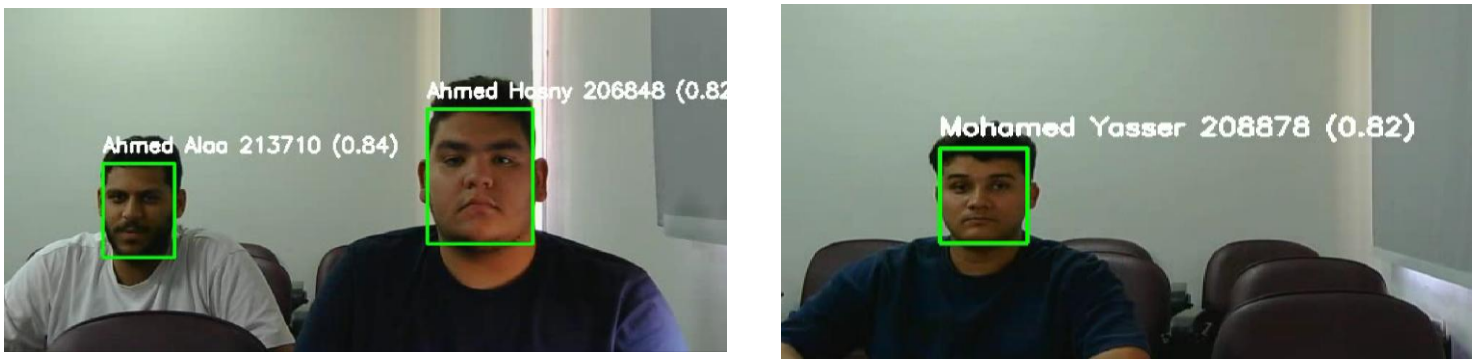

Figure 12 Student attendance system

7.1.1 Attendance system

During exams, this attendance system identifies and verifies students through real-time face recognition. It highlights the student's face with a green box and labels it with their name, ID, and confidence score on the screen. The system functions reliably whether there is only one student in the frame or multiple, and it logs the attendance data instantaneously in an Excel sheet, thus speeding up, securing, and ensuring accuracy in the process. This automated process reduces manual errors and helps support efficient, tamper-free examination monitoring.

7.1.2 Cheating and movement detector

The system detects and displays the exact direction a student is facing, either forward, left, or right, along with the status of their eye gaze. On the screen will be the labels for students to know if the student is looking forward, left, or right, and if their eyes are centered or elsewhere. Such accurate tracking would allow suspicious head or eye movements to be quickly seen during testing, thereby making it almost impossible to cheat undetected. Again, this function works perfectly when there's one or many students present.

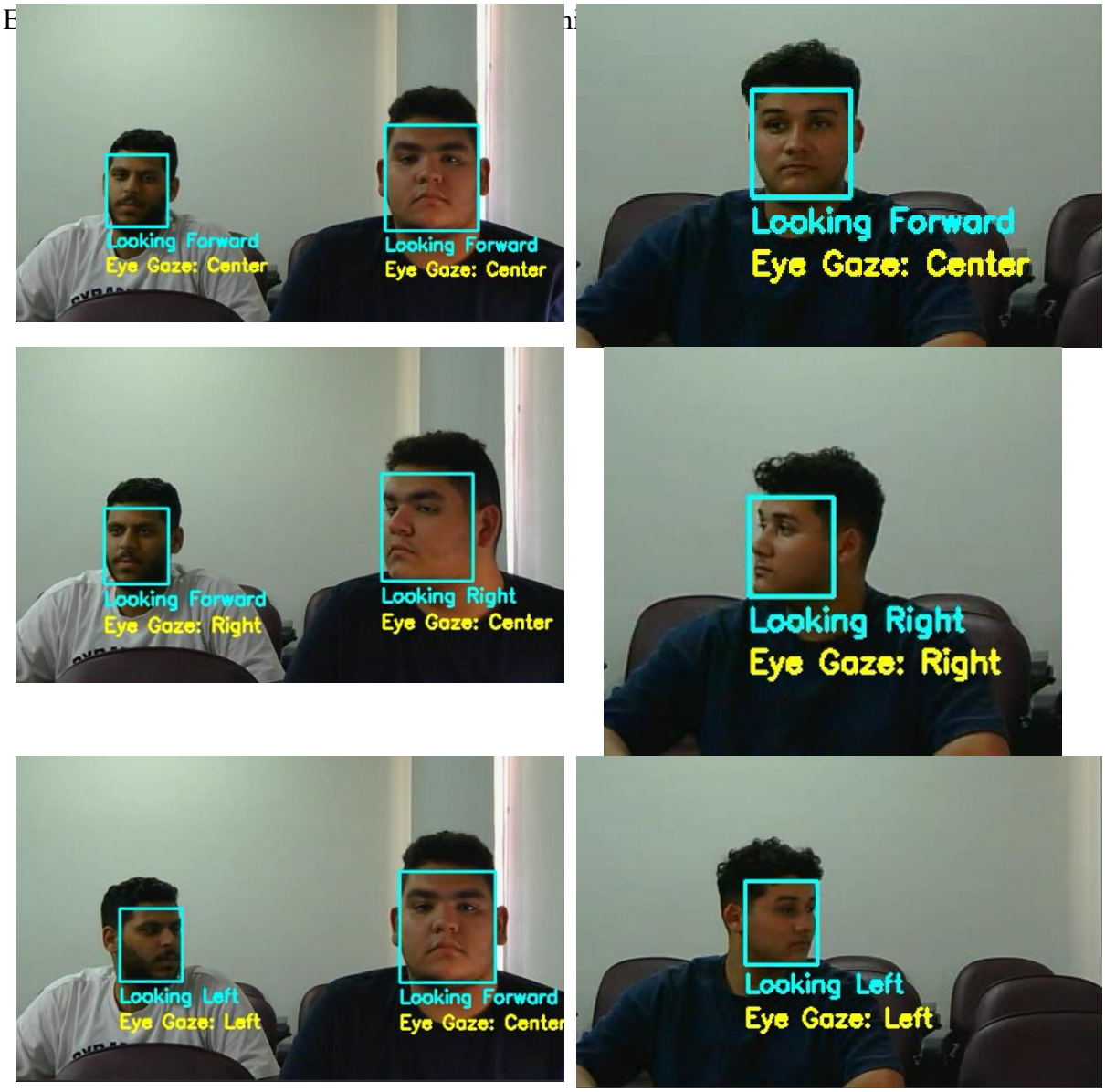

Figure13 Head Movement detection

7.1.3 Hand gestures detection

Tracking hand movements considers the position of each hand in real-time. If the hand is kept in a normal, static position near the body, the box will be green, indicating nothing suspicious. However, if the hand moves away or exhibits strange behavior, this will render the box red, indicating that the movement tends to be considered an instance of potentially cheating. This feature might affect the system by locating the actions of a candidate in reaching for unauthorized materials or signaling to others, hence, it intercepts students cheating under the eyes of the system. This color indication gives fast and instantaneous clues to exam supervisors to respond accordingly.

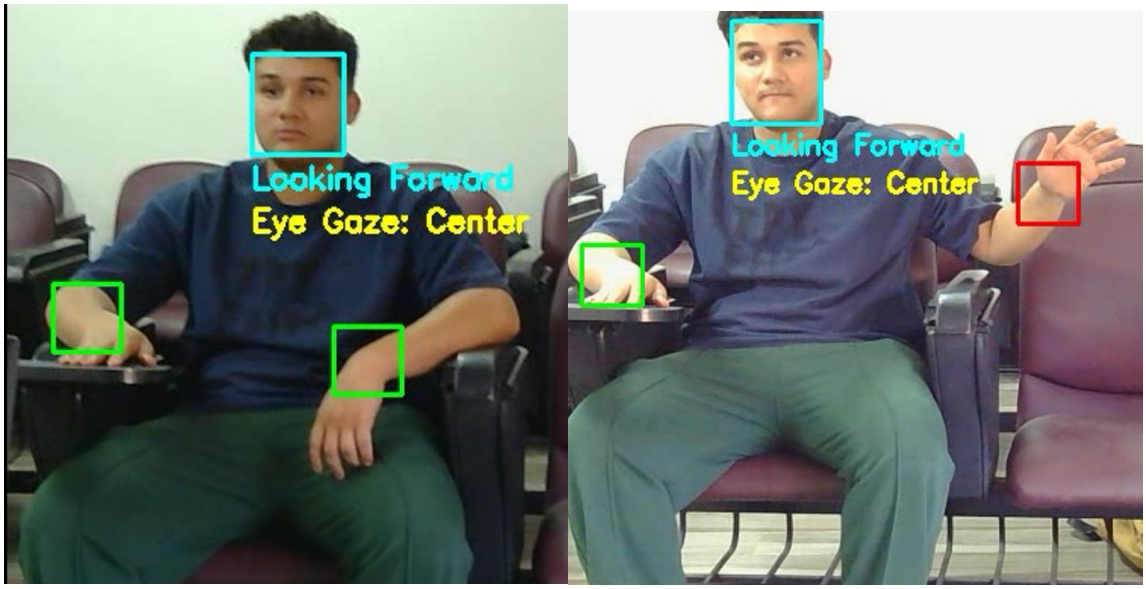

Figure 14 Hand gestures detection

From these observations, you can see that the system reliably identifies students, monitors head and eye direction, and detects suspicious hand movements in real time. It adapts to both single and multiple students, works under classroom conditions, and provides clear visual feedback for every action. The model highlights normal behavior and flags suspicious activity instantly, supporting supervisors with objective, automated monitoring. These features make the system effective for practical use in exam environments, increasing both security and accuracy while reducing human error and bias.

8.0 Conclusion and Future work

Machine learning and computer vision techniques have been implemented for the reliability of real-time monitoring in the online and non-remote setting of the examination. The system successfully identifies students and their faces, monitors head movement and eye direction, and sounds alerts about hand movement considered suspicious. Automated attendance check and behavior analysis lessen human errors and enhance security, while supervisors get supported with definitive and actionable details. The system either works on a one-student or several-student basis and, therefore, adapts to realistic classroom settings while still maintaining considerable detection speed and accuracy. These results demonstrate that using AI to maintain exam integrity-related outcomes can be genuinely feasible and effective.

In future developments, attention will be paid toward extending the system's detection range, enabling coverages of larger rooms with a wider variety of layouts. The presence of multiple cameras will enhance the field of view while minimizing blind spots. Shifting the dataset to the cloud will enable scalability and remote updates. Optimizing the recognition pipeline shall allow for accurate identification of each student with two or three reference photos, thus reducing dataset size and registration effort. All these are made toward making the system flexible, efficient, and easy to deploy in an extended set of exam environments.

References

[1] G. Moukhliss, R. F. Hilali, and H. Belhadaoui, “Intelligent solution for automatic online exam monitoring,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5333–5341, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5333-5341.

[2] *2016 International Conference on Information Science (ICIS)*. IEEE, 2016.

[3] K. Gopalakrishnan, N. Dhiyaneshwaran, and P. Yugesh, “Online proctoring system using image processing and machine learning,” *Int J Health Sci (Qassim)*, Jun. 2022, doi: 10.53730/ijhs.v6ns5.8777.

[4] B. Erdem and M. Karabatak, “Cheating Detection in Online Exams Using Deep Learning and Machine Learning,” *Applied Sciences (Switzerland)*, vol. 15, no. 1, Jan. 2025, doi: 10.3390/app15010400.

[5] A. Adhikari, “YOLO-based face recognition for automatic cheating detection in examination environments,” *~ 29 ~ International Journal of Electrical and Data Communication*, vol. 5, no. 2, 2024, [Online]. Available: https://www.datacomjournal.com

[6] “Deep Learning-Based Multimodal Cheating Detection in Online Proctored Exams,” 2024.

[7] Y. Zuo, S. S. Chai, and K. L. Goh, “Cheating Detection in Examinations Using Improved YOLOv8 with Attention Mechanism,” *Journal of Computer Science*, vol. 20, no. 12, pp. 1668–1680, 2024, doi: 10.3844/jcssp.2024.1668.1680.

[8] A. Singh and S. Das, “A Cheating Detection System in Online Examinations Based on the Analysis of Eye-Gaze and Head-Pose,” European Alliance for Innovation n.o., Jun. 2022. doi: 10.4108/eai.16-4-2022.2318165.

[9] “Deep Learning-Based Multimodal Cheating Detection in Online Proctored Exams,” 2024.

[10] F. Ozdamli, A. Aljarrah, D. Karagozlu, and M. Ababneh, “Facial Recognition System to Detect Student Emotions and Cheating in Distance Learning,” *Sustainability (Switzerland)*, vol. 14, no. 20, Oct. 2022, doi: 10.3390/su142013230.

[11] A. Mollahosseini, B. Hasani, and M. H. Mahoor, “AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild,” *IEEE Trans Affect Comput*, vol. 10, no. 1, pp. 18–31, Jan. 2019, doi: 10.1109/TAFFC.2017.2740923.

[12] H. Tang, W. Liu, W. L. Zheng, and B. L. Lu, “Multimodal Emotion Recognition Using Deep Neural Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2017, pp. 811–819. doi: 10.1007/978-3-319-70093-9_86.

[13] R. Kosti, J. M. Alvarez, A. Recasens, and A. Lapedriza, “Emotion recognition in context,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., Nov. 2017, pp. 1960–1968. doi: 10.1109/CVPR.2017.212.

[14] A. Dhall, R. Goecke, T. Gedeon, and N. Sebe, “Emotion recognition in the wild,” Jun. 01, 2016, *Springer Verlag*. doi: 10.1007/s12193-016-0213-z.

[15] A. Tweissi, W. Al Etaiwi, and D. Al Eisawi, “The Accuracy of AI-Based Automatic Proctoring in Online Exams,” *Electronic Journal of e-Learning*, vol. 20, no. 4, pp. 419–435, Jun. 2022, doi: 10.34190/ejel.20.4.2600.

[16] A. Bensakhria and P. Fergus, “Leveraging Real-time Edge AI-Video Analytics to Detect and Prevent Threats in Sensitive Environments,” 2023.

[17] S. Patil, “Federated Learning for Privacy-Preserving AI: Revolutionizing Data Sharing Across Industries.”

[18] K. Bonawitz *et al.*, “Practical Secure Aggregation for Privacy-Preserving Machine Learning.”

[19] B. Erdem and M. Karabatak, “Cheating Detection in Online Exams Using Deep Learning and Machine Learning,” *Applied Sciences (Switzerland)*, vol. 15, no. 1, Jan. 2025, doi: 10.3390/app15010400.

[20] F. Ozdamli, A. Aljarrah, D. Karagozlu, and M. Ababneh, “Facial Recognition System to Detect Student Emotions and Cheating in Distance Learning,” *Sustainability (Switzerland)*, vol. 14, no. 20, Oct. 2022, doi: 10.3390/su142013230.

[21] Y. , C. S. S. , & G. K. L. (2024). C. D. in E. U. I. Yolo. with A. Mechanism. J. of C. S. 20(12), 1668–1680. Zuo, “Zuo, Y., Chai, S. S., & Goh, K. L. (2024). Cheating Detection in Examinations Using Improved YOLOv8 with Attention Mechanism. Journal of Computer Science, 20(12), 1668–1680.”.