

Exam Proctoring System Using Face Detection

Rajini Chittimalla, Sujung Choi, Madhu Sai Vineel Reka

Department of Computer Science

Missouri State University

Springfield, USA

rc2523s@MissouriState.edu, sj0998@MissouriState.edu,

mr53s@missouristate.edu

I. ABSTRACT

Online exams present a vulnerability where students can employ various cheating methods. To address this challenge, significant efforts have been made to enhance exam proctoring systems using multi-modal approaches. In this paper, we propose an exam proctoring system utilizing the Mediapipe face detection technique. By leveraging this technique, our system detects a student's face and extracts six facial landmarks, which serve as crucial data for the cheating classification process. To achieve optimal prediction results, we trained a neural network model using collected dataset. Through extensive testing, our system successfully identifies instances of cheating by monitoring the number of times a student looks left or right and quantifying the duration of each gaze. Notably, our proposed system achieves an impressive classification accuracy of 100%, thus addressing the identified problem effectively. This research contributes to the advancement of exam proctoring systems, enhancing their ability to detect and deter cheating behaviors in online exams.

II. INTRODUCTION

As the E-learning courses have increased rapidly, advancing the technology for exam proctoring has been important for students to take exams anywhere and anytime but still reasonably

under monitoring. However, it is difficult for a proctor to be physically present and monitor the students at all settings. Therefore, it is essential to automate the monitoring system in place of the physical presence of a proctor. It has been challenging to develop proctoring techniques due to the multiple ways students try to cheat. For example, a student should pay attention to his system while taking an exam, but if the student's head is turned in any other direction, then it is assumed that the student is likely to cheat. To prevent such malpractices, many proctoring methods have been developed by integrating multi-modals such as estimating head pose, object detection, face recognition, eye-tracking, and hand gestures. In this paper, we propose an exam proctoring system using face detection that extracts certain facial feature points. By simplifying the approach, we only consider three conditions which are looking front and either side (i.e., left or right). In this way, the system can detect if a student is trying to look somewhere outside of the screen and report suspicious activities.

III. LITERATURE REVIEW

Since Covid-19, there has been a change in terms of assessments and has shifted towards online exams. The absence of a physical invigilator is one of the drawbacks of online exams for proctoring. Hence, online, and AI-powered proctoring solutions are becoming popular. To replace the physical presence of a proctor during the exams, Malhotra et al. [1] proposed a strategy by developing a multi-modal system. The system detects the emotion of the examinee, the presence of a cell phone, book, or any other person that can be used for malpractice, and the head pose of the examinee. A framework is presented for the real-time monitoring of examinees by combining existing proctoring platforms with facial expression recognition models using the architecture of convolutional neural networks (CNN). The proposed CNN model was trained on the FER2013 (Facial Expression Recognition 2013) dataset and the dataset was divided into a training set (consisting of 28709 images belonging to 7 classes) and a test set (consisting of 7178 images belonging to 7 classes). To estimate the head pose, head pose estimation, first faces were detected in the input image and then facial landmarks were extracted. Six facial landmarks were used: nose tip, chin, extreme left and right points of lips, and the left corner of the left eye and right corner of the right eye. Measured head motions using yaw angle. The angle can

range from 0 to 90 degrees on one side and 0 to 90 degrees on the other. The yaw angle will be zero for the student in a frontal stable state and the change in the yaw angle for a long period of time is considered cheating. To detect the presence of malicious objects like cell phones, a book, and multiple people, using the pre-trained weights of the YOLOv3 model was obtained by training the model on the COCO dataset. If the count of the person class is not equal to one and when numerous people are detected, then a flag is created. The book index is 61 and the cell phone index is 67 in the COCO dataset. The presence of a phone and a book can be tracked if the detected class index matches the defined index in the dataset. The integration of these models creates an Intelligent rule-based inference system that can determine if any malpractice took place during the exam. The proposed system can be used with a secure exam browser to prevent cheating in online exams.

Monteiro et al. [2] proposed a new AI-based online proctoring website using CNN/RNN algorithms. The three components they considered were the active video frame capture, frame input to the CNN algorithm, and its processing. It includes features such as viewing students' screens anytime during the exam, notifying them via warning messages when suspicious activities were detected, terminating a student's exam, eyeball tracking, and lip movement tracking. They set a threshold value and let the supervisor get notified when it detects the threshold value goes beyond the set limit. They utilized the framework Flask and Twilio to implement webRTC. For eye tracking detection, they considered extreme left, extreme right, top, and bottom that go beyond a certain range from the initial location of the eyes. For that, they used libraries, opencv and dlib in Python. To locate the eyeballs, not the entire face, they applied black mask using numpy. To detect if the student is talking, mouth-opening detection was used. For that, they also deployed Dlibs facial key points to locate the lips of the user, and once the system measures the initial distance between the closed lips, it could detect when the distance suddenly goes beyond a certain limit. For head pose estimation, they used the Caffe model which is a DNN module in OpenCV. It takes an input size of 128x128 that contains the candidate's face. For accurate results, they used focal length, optical center, and radial distortion parameters, and by projecting the 3D points on a 2D surface, they were able to detect whether the head is up, down, left, or right. Also, they used 6 points of the face, including extreme left and right points of a lip, left

and right corners of each eye, nose tip, and chin—to detect the head pose. By calculating the distance and angle, they found when the head is down or toward the right, it would be equal to or greater than 48 degrees while when the head is up or toward the left, it would be equal to or less than -48 degree. As a result, they successfully developed an AI-based proctoring system based on the CNN algorithm using real-time updates via dlib.

Prathish et al. [3] proposed a comprehensive multi-modal system for proctoring exams. The authors aimed to develop a system that is fully automated, multi-modal, inexpensive, and user-friendly with basic hardware. The proposed system is based on capturing audio and video through a webcam and active window capture. By combining such features, the rule-based inference system decides whether suspicious practices occurred during the exams. The system extracts feature points of the person's face and estimates a head pose. Malpractices are detected based on the face's angle, audio presence, and system usage. For the system usage, they detected if the examinee opens any web browsers or connects any other devices through USB. For the video analysis, they considered different features such as the duration of face disappearance, head pose estimation, multi-face detection, and landmark localization. To ensure the presence of the examinee throughout the exam, face tracking continuously detects face feature points. To detect the head movements of the student, the yaw angle, which indicates the left and right turns of the head, can be measured. If the student is looking in front of them, the yaw angle in general will be nearly zero. The system detected if the yaw angle goes beyond 15 degrees to the other side. Also, it considered the time period and the number of movements because it should not consider just a single short movement as malpractice. Likewise, they captured audio that goes over the standard threshold values. The inference system combined all of the outputs and classify them to detect the chances of misconduct. For the experiment, they created a dataset including 39 videos with two minutes per video on average. Among the dataset, they used some videos to conduct experiments considering different scenarios such as the existence of multiple faces, having different yaw angle variations, and face disappearance. For another experiment, they compared the proposed system with a real proctor. As a result, out of nine total cases, it correctly classified malpractices in seven cases, a false positive in one case, and a false negative in one case. A similar experiment was conducted for active window capture. The results showed

an 80% accuracy rate for the proposed system. The results proved that the system they developed gave a better performance rate than the existing systems.

Even though e-learning and online exam proctoring techniques have been developed, it needs to be more accurate while not requiring much manual force. Therefore, Susithra et al. [4] proposed a framework that returns high precision with less manual force using head movement and hand orientation. They proposed a framework using OpenCV. The first step is pre-processing to eliminate the anomalies and noise from the input video by using Gaussian Filter. After that, the system extracts Harr-like features which allows finding face and mouth areas quickly. It resizes the image to a grayscale picture and, in turn, to a matrix. For classification, they used Neural Network and multiple layers in that to analyze particular face images from other objects. For the next step, they used the Haar Cascade technique, which is an object detection algorithm that detects the human face through video. And they used ADA boosting technique to make a strong classifier from weak classifiers. The system was trained to detect human faces in the dataset, and if it detected suspicious face images, it warns the student by giving the alarm sound and sending the capture to the exam proctor. While in the previous system, it was more complicated to figure out who is conducting malpractice and also needed one proctor to watch 20 students, this proposed system advanced in a way that it requires only one proctor per 50 students which is much less. It also adds some features, for example, when the person keeps looking straight, it sends a message saying "Thanks for concentrating" while when the user turns to the other side, it gives a warning via an alarming sound with the message "Don't turn that side." As a result, they got an accuracy of 90%.

Hossain et al. [5] proposed an online proctoring system based on eye gaze estimation and head poses estimation features. They collected 30 recorded videos from 10 participants and each video's length was around 4-5 minutes. For head pose estimation, they detected the head from the extracted images and calculated the head orientation angles. They used two head orientation angles, which are yaw and pitch. To recognize the entire human head, they used the built-in head detector model based on neural networks. The neural network architecture consists of six convolutional layers and one hidden layer that is fully connected and contains 512 neurons. For eye gaze estimation, they used OpenCV and Dlib which are machine-learning libraries. The

data was classified as one or zero with respect to time (0=non-cheating, 1=cheating). If the student's head movement or eye gaze were detected as not normal, then the time was labeled one. If they seemed to be normal, then were labeled as zero. They used two classifiers, which are XGBoost and Multilayer Perceptron (MLP). Using the hybrid classifier (i.e., combination of the two classifiers) gave an accuracy rate of 96.5%. By comparison, with Logistic Regression model gave an accuracy of 73%, SVM gave 72%, and the DT model gave 79%, which proved the hybrid model of XGBoost and MLP they selected gave the best result.

Any educational program, including online education, have exams. Fraud is a possibility in any exam; hence detection and prevention are crucial. Exams fall into one of three categories: conventional, online, and distant Online human supervision, semi-automated supervision, and completely automatic supervision are the three categories into which online supervision techniques can be divided. Online human proctoring refers to the practice of having a remote proctor keep an eye on students while they take an online test. The behavior of the student's interactions with the computer during the exam are recorded. Analysis might be done directly or after an inspection. Anomaly detection, matching real inputs with known and unknown a priori profiles of students, feature extraction based on behavioral models at the molecular level (which also consists of a combination of numerous phases), and anomaly detection are some of the steps that make up the study. In the study of Rabiha et al. [6] used the systematic literature review (SLR) method recommended by Kitchenham and Charters to demonstrate the functionality and applicability of facial recognition models in online testing. The information is utilized to train the CNN model, which analyzes facial expressions to identify the emotions present in each image of a human face in the dataset. A grayscale 48x48 pixel image of a face makes up the dataset. In each frame, the face is basically in the same place and roughly centered. In order to discover and examine the facial recognition-based online test monitoring model approach utilized to catch cheaters, the authors conducted a literature analysis. A designed set of inclusion and exclusion criteria led to the selection of thirteen studies. The authors further analyzed the Face Detection Method, Face Recognition Method, Initial Feature, Behavior Analysis, and Evaluation Metrics utilized in each study using the data from these studies to address the research issues.

The difficulty of an online exam is using a proctoring approach to detect any potential for

cheating while maintaining its integrity. There are instances when there is no standard for supervision or when there are interventions during the supervision process, which is one of the vulnerabilities in the monitoring process with human proctors both offline and online. In most cases, the automated proctoring approach created to confirm an examinee's authenticity involves reviewing the video that was taken during the exam. Video data typically consumes a sizable amount of data bandwidth, which makes data processing and storage more difficult. Additionally, this can make it difficult for the test-taker to complete the exam. Yusuf et al. [7] proposed a technique to automate the exam proctoring. Under different circumstances, such as poses and image quality in face photographs, face augmentation is utilized to create key-face images. Face verification compares the examinee's image from the exam with the image from registration that has been enhanced through the augmentation process. A sizable computing procedure and a variety of training data are needed for the facial recognition method's verification phase. If additional test takers are added, the recognition model needs to be retrained. The transformation of data into several kinds of data is known as a data augmentation approach, and it is typically used with training models. Geometric and photometric transformation are examples of generic transformation. Rotation, reflection, flipping, zooming, scaling, cropping, padding, perspective transformation, elastic distortion, lens distortion, and mirroring are a few examples of geometric changes in photographs. Color jittering, gray scaling, filtering, illumination disruption, noise addition, vignetting, contrast modification, and random erasing are all examples of photometric alteration. The implementation and validation of the suggested fraud detection design are the upcoming tasks. The process accuracy and speed of this design will be assessed. Face verification, as well as precision, recall, F-measure, and accuracy in the cheating detection procedure generally, were all evaluated using TAR-FAR. Each process' execution time is also tracked, and this information is utilized to gauge how quickly the automated exam proctoring operates.

The main goal of the exams is to evaluate a student's status of being progress both personally and professionally. Since its inception, it has been accompanied by numerous strategies, such as cheating, that enable it to be defeated with the least amount of effort. There are many different modalities, and the goal is obvious. Many educational institutions want to put in place a system that will let them watch their students take online exams and spot cheating. The study of A. H. S.

Ganidisastra and Y. Bandung [8] aims to find any instances of plagiarism that may have occurred when creating an online test. Since there was a dearth of literature outlining the architecture of the under-investigation surveillance system, it was suggested that the study team look for published information on projects related to those of intelligent surveillance. Due to the development of facial recognition based on ARM architecture, it was decided to investigate similar architectures of other types of applications. The authors proposed a solution to improve a facial recognition by implementing a facial recognition system based on architecture and software under ARM where the recognition algorithm can be continuously optimized. A solution is suggested to make an expert system's learning process easier and boost the accuracy of its behavior pattern analysis findings. Experiments on behavior in a military environment were conducted in order to assess this concept. Where the experiment-based monitoring method was put into practice in a scenario created to analyze the movement of specific objects that would act as a detection target. In this study, they proposed using a method of incremental face recognition training to user faces. According to the incremental method, user faces are trained gradually using an image of their face that was obtained during a lecture session. The previous dataset saved in the server storage that has already been trained is erased and replaced by the new dataset each time a fresh collection of user's images is gathered. The prototype of the system is created in order to assess the effectiveness of the suggested approach. Before being input into the Face net model, the incremental training is applied to the face detection and face embedding stages. Since the image shot by the user's smartphone will be shrunk to 320x244 pixels, while the face extracted from the image is resized to 160x160 pixels based on the input size utilized by the Face net model, larger input image size is not linearly correlated to the accuracy of face detection. Python and TensorFlow environments are used to construct the face detection, face verification, and training processes. However, the accuracy of face recognition may be improved by using the face detection approach. The deep learning approach (MTCNN and YOLO-face) performs better than the conventional ones (Viola-Jones and LBP). Therefore, the deep learning approach would be worth considering for facial recognition. While the comparison between MTCNN and YOLO-face reveals a slightly different evaluation outcome. Based on the used datasets, YOLO-face has surpassed MTCNN in terms of both detection speed and accuracy rate. The technique

with the best performance will be able to meet the requirements for creating an accurate and reliable proctoring system.

As traditional exam security got outdated, several cheating methods have been developed by Essahraoui et al. [9]. It becomes essential to use cutting-edge technologies for automating cheating case identification. Numerous educational institutions have observed that the rate of exam fraud has increased. According to a 2020 investigation by McCabe, 64% of high school students admitted to cheating on a test, 58% admitted to plagiarism, and 95% said they engaged in some type of cheating, whether it was on a test, through plagiarism, or by copying homework. Since the supervisor cannot always keep an eye on the students, an intelligent anti-cheating solution based on machine learning is required to assist in student monitoring. One of the fundamental components of computer vision is face detection. Additionally, it serves as a starting point for additional research to pinpoint specific people and mark important features on their faces. Authors employ a pre-trained model known as the Caffe model in the situation. The Caffe model of OpenCV's DNN module was chosen because it performs effectively with occlusions, quick head motions, and can recognize side faces. Additionally, it offers frames with a size of 300×300 and provides the fastest frequency-per-second (fps) of any other face detector. The high-level feature extraction step is applied to each video frame after the ingestion step of video frames is begun for each video stream. This creates the input dataset from the video streams before the training phase. The proposed model achieves 94% accuracy on the training set and 75% accuracy on the test set, according to the implementation results, but its performance evaluation was quite above the average due to system restrictions like a lack of data and a significant time-consuming due to the use of a nonparallel processing design.

Since students must take tests remotely, conducting exams online proves to be rather difficult. Certain activities hamper the integrity of an online exam like students switching between tabs of exam and browser window, deviating from webcam, use of mobile, and multiple persons. Using a secure browser with face detection, object detection, and tab locking technologies all integrated into one can help to lower the likelihood of such malpractices. Sasikala et al. [10] states in their paper that in the initial stage of online exam proctoring, a face detection algorithm is executed to detect any face objects in the cam feed. When the algorithm recognizes the test-takers face, it

stores the features that were extracted from it and stores them in the database against a unique ID that will be used for further validation. Once the test had begun, the predefined Viola-Jones facial detection algorithm was activated and run continuously to track the student's activity. One of the key activities to check for misconduct was students getting distracted or deviating from the screen. This usually happens when the student tries to overlook the computer screen or peek away from the screen in the lookout for answers. These actions were tracked using the face's coordinates from the screen, and a warning message was sent if the values rise above a threshold value. The authors discussed following four important and efficient face detection algorithms for online proctoring:

- 1) Face Detection Algorithm using Viola-Jones: most effective methods for object detection in image datasets.
- 2) Successive Mean Quantization Transform (SMQT) Features and Sparse Network of Windows (SNoW) Classifier Method: SnoW was used to speed up the original classifier while SMQT was suggested for lighting of an image and insensitive operation in detection.
- 3) Face Detection using Neural Networks: The face detection strategy used by this algorithm was to inspect each window and determine if a face is present or not.
- 4) Support Vector Machines (SVM): It uses a supervised learning model that draws a hyper-plane to categorize the data points into two classes. Certain data points were considered as Support vectors that exactly sit on the boundary of the hyper plane's margin. Margin refers to the difference in distance between the closest data point and the hyper-plane.

The efficiency of these face detection algorithms was compared using common parameters called precision and recall. The values were obtained by plotting a precision vs recall graph that produces value when it recognizes faces in the input image provided to the model. The result of the comparison presents that Viola-Jones face detection is the best and most accurate method for face detection applications.

One of the main challenges of online exams is the possibility of fraud and cheating during the exam. To address this issue, Nguyen et al. [11] proposed a model of Proctoring of Online Test (POT) using an embedded system with artificial intelligence algorithms to authenticate applicants and predict cheating behavior. The advantage of the system is its ability to monitor the online

exam automatically and continually in real time. Two cameras are attached to the embedded system to record images and conduct behavioral analysis. The Techniques used to build the POT model are determining the state of the eyes and estimating the head posture. The aspect ratio of the eye and the given set of horizontal and vertical coordinates can be used to identify the eye's state of vision. The aspect ratio of the eyes is calculated using waypoints referred from a 2D image to generate a 3D mask of the face. The eyes are then surrounded by six markers, starting in the bottom left corner, and moving clockwise to the right. The position of the eye can be calculated using an aspect ratio equation from the relationship between width and height. To determine whether cheating or not, it is necessary to calculate the percentage of the iris and eye borders. To estimate the head posture, 6 facial landmark parameters (3 reference points for horizontal rotation, 3 reference points for vertical displacement) are used. A face landmark mask in 3D form made up of the projection of the appropriate landmarks in the 2D image will be used to pick the landmarks of the head position with the coordinates (x,y) of the face. To represent the head posture, OpenCV synthesized and created the Perspective-n-Point (PNP) equation. POT used video clips with uninterrupted frames over time to identify and evaluate human behavioral patterns. The MediaPipe Pose toolkit identifies the body with 33 landmarks applied in areas such as face detection and human behavior recognition. To determine candidate behavior, selected 22 body markers from the upper body. User interface (UI) will be used by proctors, applicants, and administrators to access the POT system. When logging into the monitoring system, the proctor will choose the appropriate exam from the list. The proctor will watch the candidate's live video on the screen and send reminder text messages, which will be recorded in the exam log. The proctor will determine whether to suspend or resume the test if the candidate breaks one of the exam's regulations after receiving a message from the system. After the exam is finished, the POT will upload the test and the results of the artificial system's monitoring and supervision to a cloud server. If cheating is not recorded on camera, the contestant's image will be removed after five minutes to remove load on the cloud storage system. According to experimental results, the POT system can identify fraud at a rate of above 85%. This serves as the foundation to assess the effectiveness of POT system for monitoring online exams and promote honesty of students.

Due to the COVID-19 epidemic, universities and institutions adopted online exams as the

norm. Online exam proctoring is one of the challenges to be addressed to maintain the integrity of online exams. Existing proctoring methods are time consuming and labor-intensive and need a small number of proctors to monitor a large number of students in order to identify cheating students. To address this issue, A. Abozaid and A. Atia [12] implemented multi-modalities using a webcam to monitor the student's activity during the online exam and sends a report to the proctor for the suspected student. The modalities include head-pose, and eye-gaze estimation. The head position is thought to be a crucial modality to detect cheating in an online exam. In the implementation of head pose, five models VGG16, VGG19, RESNET50, Xception, and InceptionV3 models were tested and compared them for best result to work with head pose. Combined two datasets, cropped version of pandora dataset and from Gourier et al.'s paper to train the model. Cropped pandora dataset, has 15,679 photos of 10 males and 12 females from various head angles, while the second dataset, contains 2790 images of 15 people from different head angles. Merged the two datasets and categorized them into the focus, left, and right classes. Using cvlib, detected and cropped the images on the head only so that there is no background in the image and then converted them into grayscale. Eye gazing is other crucial modality to identify abnormal exam-taking behavior from the student as they can direct or gaze to an object that does not appear to the camera, such as another monitor, paper on the wall, or a variety of other cheating methods, without moving his or her head or using any object that could be seen by the camera. The Eye gaze is implemented using a model introduced by A.Abdelrahman. L2CS-Net. Using multi-loss, the model takes an RGB image and calculates the 3D gaze angles. Using two loss functions each gaze is given an angle, either the yaw or the pitch. At last, the head pose modality and eye gaze modality are combined into one system. The system input is a camera stream or recorded footage of a student taking an exam. Every 10 seconds, the system captures a picture frame, which is then given to the two modality functions. First, the head pose which crops the face and converts it to grayscale, then the image is sent to the model to determine the direction of the head and is attached to a report if it contains abnormal behavior. Second is the eye gazing model that determines the pitch and yaw angles, detects the direction of the gaze and is included in the report if it contains abnormal behavior. To evaluate the system, three experiments were done with the help of 29 students. The results show that in experiment

2, the accuracy of detecting the events is 96.66% and 95.69% in experiment 3.

IV. BACKGROUND

Neural networks is a type of machine-learning model that is inspired by the human brain [13]. The idea behind the neural networks is to process data in a way that the biological brain works by creating and simulating a network of artificial neurons in a layered structure that can learn to recognize patterns in data. The input layer receives the data, and the data is processed by one or more hidden layers. Each hidden layer is made up of a group of neurons, which alter the input data in a nonlinear way. Based on the data that has been analyzed, the output layer generates the final prediction or classification result. In this work, neural network model for multi-label classification is implemented to classify if a student is cheating or not by detecting them looking left or right during the exam. Therefore, the model is trained on a labeled dataset, which is called supervised learning. Supervised learning is one of the machine learning approaches which requires labeled data in order to train a model that can make predictions based on that. Labeling data is important in supervised machine learning because it allows the model to understand the patterns between input features and the output variable in order to identify the new data and label them to particular classes. In our method, the dataset is classified into three classes which are normal, right, and left and are labeled as 0, 1, and 2, respectively.

MediaPipe is an open-source Machine Learning based solution for live and media streaming. We use Python programming code to detect face of a participant from the webcam. In particular, we use the face detection method. The face detection method returns the six points for the bounding box around the face. Each of six points contains x and y coordinates of LEFT_EYE, RIGHT_EYE, NOSE_TIP, MOUTH_CENTER, RIGHT_EAR_TRAGION, LEFT_EAR_TRAGION. All the coordinates are normalized between 0 and 1.

TensorFlow is an open-source software library developed by Google for numerical computation and machine learning. It provides a flexible and efficient way to build and train different types of neural networks, including deep learning models.

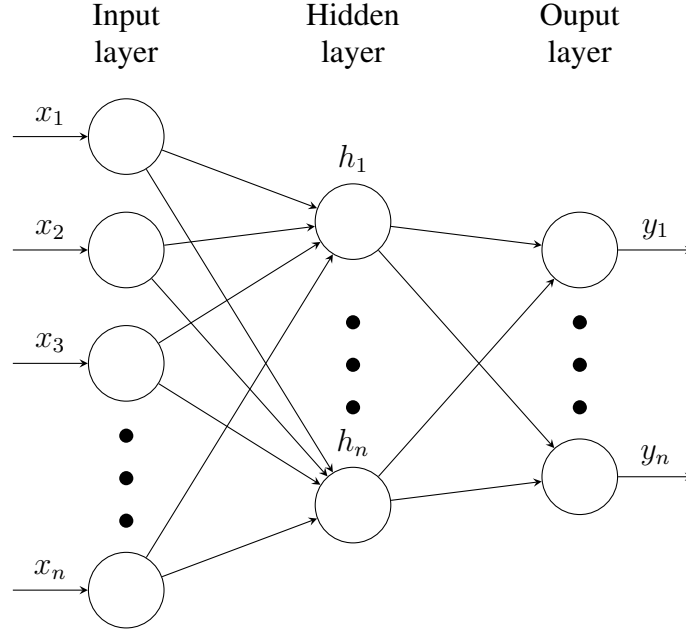


Fig. 1. Neural Network Architecture

Keras provides a high-level API built on top of TensorFlow for training the neural networks. With its easy-to-use framework, Keras simplifies the implementation of complex neural networks. Keras framework is used to design the deep neural network as it requires minimal codes to build and test the model.

V. METHODS

We considered three cases, whether a person is looking front, left, or right as shown in the figures below. The objective was to classify the behavior as either normal (front) or cheating (left

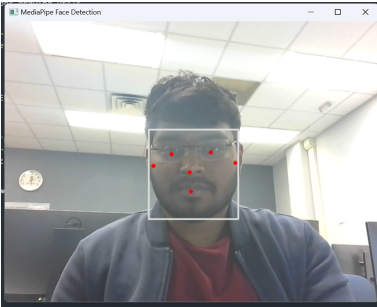


Fig. 2. Front

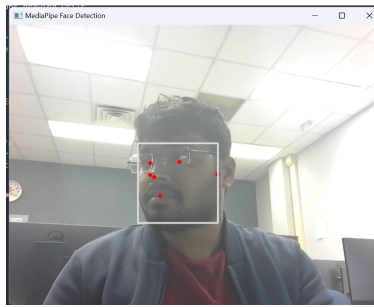


Fig. 3. Left

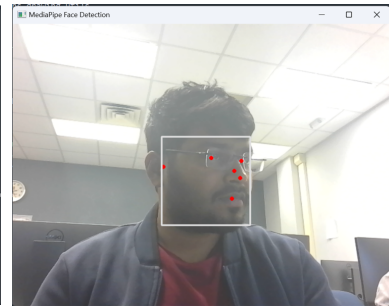


Fig. 4. Right

or right). We gathered a dataset consisting of 500 samples for each class, representing instances

of looking right, front, and left. The dataset encompassed the following features: “right_eye_x”, “right_eye_y”, “left_eye_x”, “left_eye_y”, “nose_tip_x”, “nose_tip_y”, “mouth_center_x”, “mouth_center_y”, “right_ear_tragion_x”, “right_ear_tragion_y”, “left_ear_tragion_x”, “left_ear_tragion_y”. To facilitate the classification task, we assigned labels of 0, 1, and 2 to the classes normal, right, and left, respectively. For training and testing the models, we employed a supervised classification approach using a Neural Network model implemented in the TensorFlow library. The model was constructed with the following hyperparameters: a learning rate of 0.0005, a batch size of 32, and a total of 350 epochs for training. We utilized the Adam optimizer to fine-tune these parameters, aiming to maximize the model’s accuracy. During training, the process was automatically halted upon reaching a threshold accuracy of 99.98% to prevent overfitting. The trained model was subsequently saved for further usage.

```
Model: "model"
```

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 12)]	0
dense (Dense)	(None, 400)	5200
dense_1 (Dense)	(None, 200)	80200
dense_2 (Dense)	(None, 100)	20100
dense_3 (Dense)	(None, 50)	5050
dense_4 (Dense)	(None, 32)	1632
dense_5 (Dense)	(None, 16)	528
dense_6 (Dense)	(None, 3)	51

```

=====
Total params: 112,761
Trainable params: 112,761
Non-trainable params: 0
=====

```

Fig. 5. Summary of the model describing the layers, activation functions and parameters

In the monitoring phase, we loaded the pre-trained model and employed it to track cheating behavior in real-time. We captured frames from a camera feed and passed each frame through the model to obtain classification outputs. Whenever the model detected a left or right direction, it initiated a recording process by setting a “start_time” variable. Subsequently, when the person returned to the normal direction, the recording was concluded, and the “stop_time” variable was set. The duration between “start_time” and “stop_time” was considered as a timestamp,

```

1/47 [.....] - ETA: 0s - loss: 0.0145 - accuracy: 1.0000
24/47 [=====>.....] - ETA: 0s - loss: 0.0054 - accuracy: 0.9987
47/47 [=====] - ETA: 0s - loss: 0.0052 - accuracy: 0.9993
47/47 [=====] - 0s 2ms/step - loss: 0.0052 - accuracy: 0.9993
Epoch 13/350

1/47 [.....] - ETA: 0s - loss: 0.0022 - accuracy: 1.0000
26/47 [=====>.....] - ETA: 0s - loss: 0.0039 - accuracy: 1.0000
Reached 99.98% accuracy, so stopping training!!

47/47 [=====] - 0s 2ms/step - loss: 0.0041 - accuracy: 1.0000
Classification error: 0.0010234194342046976
Classification accuracy: 100.0

```

Fig. 6. Accuracy and loss of the training model

representing the period during which the person exhibited cheating behavior. The number of occurrences and the respective durations were logged in a file for further analysis.

By employing this approach, we obtained valuable insights into the frequency and duration of cheating attempts, enabling us to assess and understand the extent of cheating behavior among individuals.

VI. EVALUATION

We conducted testing by ourselves to assess the performance of the trained model in predicting and classifying our own behavior. To evaluate the model, we first loaded the trained model and then passed the mediapipe-detected six face key points for each look as input to the predict function and checked if the model classifies correctly. As shown in Figure 6, when the student looked left, the landmarks on the face are differentiated using blue color and if turned right, the features are indicated using red color. During the testing phase, the primary focus was on determining whether the model could correctly identify instances when we looked right or left. We monitored our behavior using the trained model. Each time we shifted our gaze to the right or left, the model classified and logged the behavior with a timestamp indicating the starting and ending points of the look. Additionally, the duration of each look was recorded. To facilitate analysis and organization, we saved the recorded data into a CSV file, which included the timestamp, duration, and labeled behavior (right or left). Throughout the evaluation, we carefully

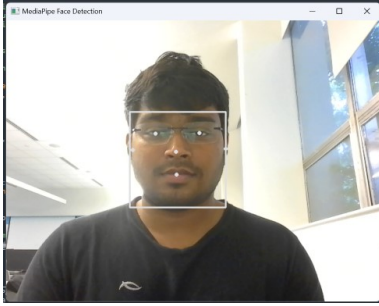


Fig. 7. Front

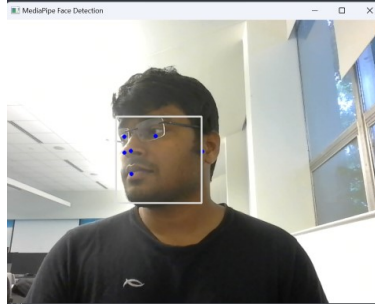


Fig. 8. Left

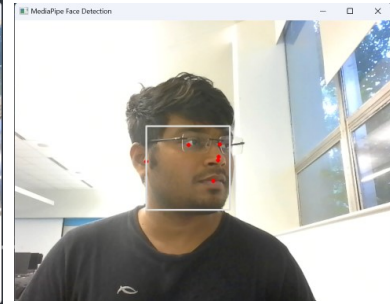


Fig. 9. Right

analyzed the recorded data, taking into account both the predictions made by the model and the ground truth behavior exhibited. We observed that the model correctly classified our behavior in most of the instances of looking left or right. This analysis allowed us to assess the accuracy of the model in differentiating between normal behavior (looking front) and cheating behavior (looking right or left).

	0	1	2	3
0	21:01:25	21:01:29	3.784005	Left
1	21:01:36	21:01:39	3.577206	Right
2	21:01:40	21:01:45	4.386283	Left
3	21:02:00	21:02:05	5.237038	Left
4	21:02:13	21:02:16	3.560598	Left

Fig. 10. Log of the results saved in record.csv

Following is the descriptions of the recorded results in the Figure 10:

- 1) Column 0 is the starting time (HH:MM:SS) of look.
- 2) Column 1 is the ending time (HH:MM:SS) of look.
- 3) Column 2 is the duration in seconds the person looked left or right.
- 4) Column 3 is the predicted label of the facial direction.

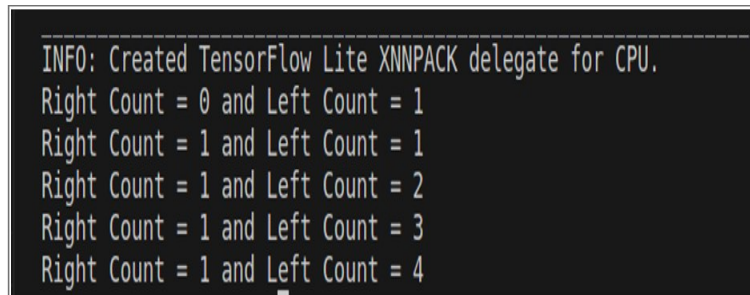
By executing this testing methodology, we aimed to assess the model's ability to accurately detect and classify cheating behavior. Furthermore, we sought to determine the effectiveness of the model in quantifying the number of looks (both right and left) exhibited by individuals.

These metrics provided valuable insights into the frequency and duration of cheating attempts, enabling a comprehensive evaluation of the model's performance.

VII. RESULTS

Firstly, we observed that the model had a minimum duration threshold for counting instances of looking left or right. Specifically, if the gaze was not sustained for at least 1 second in the left or right direction, the system did not register it as a valid action. Consequently, instances where the gaze was briefly shifted to the left or right, but for a duration less than 1 second, were not captured by the model.

From the start of invigilation, we keep a RIGHT and LEFT count, that is incremented and printed when the person moves his face left or right as shown in Figure 11. It is observed that when the face is at the center of the camera, the model correctly classified our behavior and achieved the 100% accuracy.



```
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
Right Count = 0 and Left Count = 1
Right Count = 1 and Left Count = 1
Right Count = 1 and Left Count = 2
Right Count = 1 and Left Count = 3
Right Count = 1 and Left Count = 4
```

Fig. 11. Display of model prediction

While the face detection system demonstrated promising results, it is important to acknowledge its limitations and areas for improvement. The major limitation we encountered was that the model's detection of left and right gaze directions exhibited a certain rigidity in its classification boundaries. It was observed that the model had difficulty accurately detecting subtle or slight deviations in gaze direction. This limitation may be attributed to the nature of the training dataset. During dataset generation, the team members performed deliberate and clear left or right gazes. However, in real-world scenarios, individuals may exhibit varying degrees of gaze deviation, which the model may not be able to detect with the same precision. This limitation indicates

the need for a more diverse and representative training dataset to enhance the model's ability to detect a broader range of gaze variations.

To address these limitations and improve the system, future work could focus on:

- collecting a larger and more diverse dataset to enhance the model's generalization capabilities.
- exploring advanced techniques such as facial expression analysis and gaze tracking to provide additional insights into students' behavior.

VIII. CONCLUSION

In this work, we proposed an exam proctoring system using the Mediapipe face detection technique to address the vulnerability in online exams where students can employ various cheating methods. The system extracts six facial key points for each look of the student (i.e., normal, right, and left). These points serve as dataset for the cheating classification process. We classified the dataset into three labels 0, 1, and 2 to the classes normal, right and left respectively. A neural network is used to train and test the model using training and testing datasets. Then, pre-trained model was loaded to predict the cheating behavior of the student in real-time. We captured frames from camera and passed each frame to the model to predict the classification output. We achieved 100% accuracy of the model in predicting cheating behavior. Additionally, the system records the duration and timestamp of each time the student turned their face.

While our trained model shows promising performance, it is essential to address the limitations to enhance its robustness and to be able to generalize in real-world applications. By collecting larger dataset and including a wider range of gaze variations, we expect to build a much more robust system to detect students' suspicious behavior.

REFERENCES

- [1] N. Malhotra, R. Suri, P. Verma, and R. Kumar, "Smart artificial intelligence based online proctoring system," in *2022 IEEE Delhi Section Conference (DELCON)*, 2022, pp. 1–5.
- [2] S. Monteiro, R. Bhate, L. Sharma, and P. Shaikh, "Proct-xam – AI based proctoring," in *2022 2nd Asian Conference on Innovation in Technology (ASIANCON)*, 2022, pp. 1–6.
- [3] S. Prathish, A. N. S., and K. Bijlani, "An intelligent system for online exam monitoring," in *2016 International Conference on Information Science (ICIS)*, 2016, pp. 138–143.

- [4] S. V. R. A. B. Gope, and S. S, “Detection of anomalous behaviour in online exam towards automated proctoring,” in *2021 International Conference on System, Computation, Automation and Networking (ICSCAN)*, 2021, pp. 1–5.
- [5] Z. T. Hossain, P. Roy, R. Nasir, S. Nawsheen, and M. I. Hossain, “Automated online exam proctoring system using computer vision and hybrid ml classifier,” in *2021 IEEE International Conference on Robotics, Automation, Artificial-Intelligence and Internet-of-Things (RAAICON)*, 2021, pp. 14–17.
- [6] S. G. Rabiha, I. H. Kartowisastro, R. Setiawan, and W. Budiharto, “Survey of online exam proctoring model to detect cheating behavior based on face recognition,” in *2022 8th International Conference on Systems and Informatics (ICSAI)*, 2022, pp. 1–7.
- [7] A. Yusuf, N. Suciati, and A. Saikhu, “Design of automated exam proctoring for user authentication through face augmentation and verification,” in *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, 2022, pp. 357–361.
- [8] A. H. S. Ganidisastra and Y. Bandung, “An incremental training on deep learning face recognition for m-learning online exam proctoring,” in *2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, 2021, pp. 213–219.
- [9] S. Essahraui, M. A. El Mrabet, M. F. Bouami, K. E. Makkaoui, and A. Faize, “An intelligent anti-cheating model in education exams,” in *2022 5th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2022, pp. 1–6.
- [10] S. N, B. M. Sundaram, V. N. Kumar, S. J, and H. S, “Face recognition based automated remote proctoring platform,” in *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 2022, pp. 1753–1760.
- [11] X. H. Nguyen, V. M. Le-Pham, T. T. Than, and M. S. Nguyen, “Proctoring online exam using iot technology,” in *2022 9th NAFOSTED Conference on Information and Computer Science (NICS)*, 2022, pp. 7–12.
- [12] A. Abozaid and A. Atia, “Multi-modal online exam cheating detection,” in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 2022, pp. 1–6.
- [13] S. Tchynetskyi, R. Peleshchak, I. Peleshchak, and V. Vysotska, “A neural network development for multispectral images recognition,” in *2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)*, vol. 2, 2021, pp. 278–284.