

Automated Online Exam Proctoring System Using Computer Vision and Hybrid ML Classifier

by

Zarin Tahia Hossain

18101184

Protyasha Roy

18101116

Rina Nasir

17201008

Sumaiya Nawsheen

17201001

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Zarin Tahia Hossain
18101184



Protyasha Roy
18101116



Rina Nasir
17201008



Sumaiya Nawsheen
17201001

Approval

The thesis titled “Automated Online Exam Proctoring System using Computer Vision and Hybrid ML Classifier” submitted by

1. Zarin Tahia Hossain (ID: 18101184)
2. Protyasha Roy (ID: 18101116)
3. Rina Nasir (ID: 17201008)
4. Sumaiya Nawsheen (ID: 17201001)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Engineering on September 26, 2021.

Examining Committee:

Supervisor:
(Member)



Dr. Muhammad Iqbal Hossain
Assistant Professor
Computer Science & Engineering
Brac University

Co-Supervisor:
(Member)



Mr. Nazmus Sakeef
Lecturer
Computer Science & Engineering
Brac University

Program Coordinator:
(Member)

Name of Program Coordinator
Designation
Department
Brac University

Head of Department:
(Chair)

Name of Head of Department
Designation
Department of Computer Science and Engineering
Brac University

Abstract

The term online proctoring comes hand in hand with online education. There are not many exceptions between traditional education and the online education system except for the fact that online education, as well as online examination, happens over the internet. Due to the Covid-19 situation, when going to schools or colleges and receiving an education is not possible, online education is helping the students to continue their education properly. As much as the importance and necessity of online education can be seen, questions regarding the validity of online exams also emerged. In traditional pen-paper examinations, it is quite difficult for students to cheat under the watchful eyes of the examiner. However, in online examination, the scenario is quite different as there is an absence of an invigilator and thus it is quite difficult to observe examinees and detect cheating attempts during online exams. Even though there are already many systems for online proctoring, not all educational institutes can afford them as the systems are very expensive. In this paper, we have used eye gaze and head pose estimation, along with that voice detection as the main features to design our online proctoring system. Therefore, the purpose of this paper is to use these features to create an online proctoring system using computer vision and machine learning and stop cheating attempts in exams.

Keywords: Online Proctoring; Machine Learning; MFCC; XGBoost; MLP; Hybrid Classifier; Cheating Prediction

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.

Secondly, to our advisor Dr. Muhammad Iqbal Hossain sir for his kind support and advice in our work. He helped us whenever we needed help.

Thirdly, our volunteers whose kind support made it possible to create our data set used in this research.

And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

Declaration	i
Approval	ii
Abstract	iv
Acknowledgment	v
Table of Contents	vi
List of Figures	vii
List of Tables	1
1 Introduction	2
1.1 Research Problem	3
1.2 Research Objectives	4
2 Literature Review	5
2.1 Online Proctoring	5
2.2 Online Proctoring Features	5
2.3 Related Works	6
3 Methodology	9
3.1 System Framework	9
3.1.1 Constraints of the model	10
3.2 Data Collection	11
3.3 Data Analysis	11
3.3.1 Head Pose Estimation	11
3.3.2 Eye Gaze Estimation	14
3.3.3 Audio Detection	17
4 Experimentations	20
4.1 Data Pre-processing	20
5 Result Analysis	27
6 Conclusion and Future Works	30
Bibliography	34

List of Figures

3.1	Proposed Model	10
3.2	Yaw (Pan) and Pitch (Tilt) of human head	12
3.3	System Architecture of RealHePoNet model (taken from [19])	12
3.4	Estimated Head Poses	13
3.5	Example of Head Values in CSV file	13
3.6	Flowchart of the proposed eye algorithm	14
3.7	Estimated eye gaze parameters	15
3.8	Example of eye gaze parameters collection in CSV file	16
3.9	Example of audio parameters collection in CSV file	17
3.10	Extracting method of MFCCs	19
4.1	Correlation of the Features	21
4.2	Small part of the tree	22
4.3	Confusion matrix for hybrid classifier	23
4.4	Confusion matrix for XGBoost and MLP classifier	24
4.5	Confusion matrix for Logistic Regression, SVM and Decision Tree	25
4.6	Comparison of the models	25
4.7	Comparison of the model with ROC curve	26
5.1	Report for Invigilator	28

List of Tables

2.1	Comparison between different features of Online Proctoring software .	8
-----	---	---

Chapter 1

Introduction

None can deny the importance of education in the 21st century. For being a proper human, you need proper education. In that sense, education not only makes a person knowledgeable but also teaches them to decide from right to wrong in life. If it was any normal time, to receive the basic need that is education, the presence of students can be seen in school, college, or any educational institution. However, due to the recent ongoing pandemic named Covid-19, the mentioned situation completely changes and the education system is forced to shift towards online. So that tough times like Covid-19 cannot make the education system paralyzed which can create a big problem for human society in the future, online education became a must.

To illustrate what online education is, it is nothing but the traditional teaching method where teachers give lectures, students attend those lectures, give exams; but the only difference is that it happens via the internet. Situations like this, make people realize how important and necessary this online education is to continue receiving schooling. Although the online education concept is not new and platforms like Coursera, Edx, Udemy, etc, and some universities in the world do provide education online but the countries where online education was not popular faced some problems due to this sudden shifting. Nevertheless, as almost every educational institution in the whole world became highly dependent on online platforms for completing the academic curriculum, some problems also arose.

To survive a crucial time like Covid-19 people came up with the solution which is online education. Yet, some are taking advantage of this situation completely to commit unfair means. By saying unfair means, cheating in online examinations is being highlighted. There are various kinds of cheating, for example, during the exam the examinee can browse in the search engine for answers, can ask the answer to any friend or take help from their books or notes, take help from homework solving websites, etc. In normal exams where invigilators used to be present to stop these cheating attempts, on the contrary, this is not possible in online exams due to the inability to invigilate in person. If the examinees were to be completely honest, not involving themselves in some sort of cheating, questions regarding the validity of online exams would not have arisen. Hence, online examinations will not be as fair and valid as examinations held physically; this opinion is becoming a common thought process. Bilen and Matros remarked in their paper that without proctoring if educational institutions carry out online examinations, there will be a large amount of cheating [1]. It is high time, institutions and authorities take this problem seriously as it is one of the greatest reasons why online learning does not

get similar value to traditional classroom studies. If the educational institutions continue to not be able to take online examinations properly, it will become a great barrier for online studies in the future.

For this purpose, online proctoring is very much needed. To describe online proctoring, it can be a system where invigilator can guard the examinees online in live times or software where many functions are automated to stop cheating attempts, like- monitoring or recording or taking images through web camera for eye gaze and head pose estimation, audio voice monitoring, etc. Estimation of eye gaze and head pose are helpful to catch the examinees who try to look at notes/books or other people during exams. Also, by voice audio recording it is possible to detect if the examinees are communicating with someone else during the exam. As the world is technologically developed so many online proctoring software are also built to detect the unfair means of the students during any examination. However, these software programs are very expensive. Also, for a large number of examinees, the expense keeps on rising and the efficiency of the proctoring ability keeps on decreasing. For this huge sum of money and efficiency-related concerns, educational institutions hesitate to use this type of proctoring.

In the above circumstance, we want to conduct research on this online proctoring system, specifically with eye gaze estimation, head pose estimation and audio detection features. So, the purpose of our research is to find a low-cost solution using computer vision and machine learning to prevent cheating during online examinations so that online education can get the value of normal traditional education. Moreover, we also want to solve the efficiency-related problems so that for a large number of students, the proctoring system works smoothly and catches cheating attempts properly. In the end, it will not only help us during a pandemic but also have a long-term good effect on our education overall. So the main focus of this research is to find an affordable solution that will distinguish cheating in an online exam in an efficient way.

1.1 Research Problem

An examination is one of the important parts of the education system. In the traditional system, examinations are conducted by a proctor in person, still, students conduct unethical tasks. Due to the Covid-19 situation, the online examination has become a must for every educational institution. Similar to traditional pen-paper examination, conducting a proper online examination has also become very difficult as the students are involved in various misconduct like, giving group exams or collecting resources from the internet during examination time. For example, at Harvard University, 125 students were caught in their homes for giving a final exam jointly [2]. In some cases the students have to give exams through video conferences like google meet or zoom and the proctor has to be present all the time to monitor them but for a large number of students, this process becomes very hard for the proctors.

On the other hand, if the system is fully machine-oriented then it creates an accuracy problem. To create a fully automated system with 100% accuracy we need to add so many features like human pose detection, face detection, face recognition, gaze movement, voice detection, phone detection, multiple human detection, and so on. Combining all these features is not an easy task. There is already several proctoring

software but due to inefficiency, sometimes misunderstandings are created. It is said by an examinee that when she was reading the exam question in loud the proctoring software misjudged her by thinking she was talking with someone else [3]. Research by the National Institute of Standards and Technology (NIST) in 2019 claimed that a facial recognition algorithm identifies the wrong face for the colored skin tone people more than the Caucasians' face [3].

There are various solutions for increasing the accuracy rate of online proctoring systems. Some of the systems need extra hardware or devices which increases the expense. Researchers in [4] use Microsoft Kinect- a gesture measuring instrument which tracks the pitch and yaw angle of the head and also the coordinates of hands in examinations to detect cheating attempts of the students. In [5] one 'wear cam' is included in the hardware system to record the view from the user's perspective. For tracking gaze movement external devices like EEG sensors and two web cameras are used in [6]. Using these devices may give accurate results efficiently but this approach is not affordable by all educational institutions.

So, some of the main problems of online proctoring systems can be mentioned as efficiency and cost. To have accurate results external hardware is used which increases the expense. Full automation or the presence of an invigilator during the whole examination can cause inaccurate detection of cheating attempts. In our research, we tried to address these problems.

1.2 Research Objectives

The target of our research is to develop an online proctoring system to prevent cheating attempts with computer vision and machine learning. Moreover, we also want our system to be efficient and have a low cost. To serve the purposes, the research objectives of our paper are-

- Estimate head pose angles by yaw and pitch.
- Estimate gaze to know where the student is looking at.
- Through audio detection, identify human voice and non-human sounds.
- Predict cheating attempts using all the combined features and hybrid classification.
- Find the accuracy of our model.

Chapter 2

Literature Review

The ongoing pandemic Covid-19 has caused a negative effect on education for more than 1.2 billion children in 186 countries in the world as schools and colleges are closed to stop the spreading of the virus [7]. In this situation, for the continuation of education, online education can be the solution. So, to have a valid online education system, a valid online examination is also needed as there are many events, where cheating activities take place with unmonitored online exams. For example, [8] gives an example where a contracted person used to give an exam in place of many students. In this way, online examinations may be considered unfair for some students who sincerely give the exam but fall behind because of those students who perform the restricted rules and regulations during the exam. Therefore, to have a valid and fair online examination, online proctoring is much needed.

2.1 Online Proctoring

The first online proctoring system using human proctors was introduced by Kryterion in 2006 stated in [9]. Since then many proctoring systems were developed like ProctorU, Respondus, Proctorio, AIProctor, etc [10]. Also, many researchers performed and still perform many methods and techniques to come up with the less costly and most efficient online proctoring system.

2.2 Online Proctoring Features

There are different aspects of an online proctoring system as well as different approaches to implement them. There are many features based on which different online proctoring systems are implemented. Some approaches include face detection and recognition [11], eye movement detection [12], head movement detection [13], audio detection [14] etc. Since our research is based on eye gaze estimation, head pose estimation, and audio detection, in section 2.3, we will go through the papers which implemented these features in their proctoring system. Moreover, we mainly focused on the papers that used computer vision and machine learning-related algorithms and techniques to implement online proctoring features.

2.3 Related Works

The purpose of this section is to extensively review the previous works done in this field. That is, we will try to analyze the features of online proctoring like- eye gaze, head pose estimation, audio detection, etc. from the research papers and try to illustrate the algorithms or techniques they used to collect datasets and successfully detect the mentioned features.

Raj, Narayanan, and Bijlani combined PC monitoring techniques, image and audio processing to proctor online examinations automatically [14]. According to their research paper, the students need to have a webcam and from the front view, the face needs to be clearly visible. To ensure continuous presence of the student throughout the examination and no other person is present during the examination with the student, they used face detection technique. Moreover, to detect any human presence inside the room sound was recorded. Finally, their system achieved an accuracy of false-positive rate 0.08 and true negative rate 0.13. Similarly, we have included audio detection in our system which can detect not only human voice but also other non-human sounds as well. So, in the presence of other sounds, a student will not be mistaken as talking with someone. In contrast to their system, we have implemented eye gaze estimation and head pose estimation. Even though in [14] multiple faces can be detected, a student can still perform cheating activities with someone who does not appear in front of the webcam. That is why in our proposed model eye gaze estimation and head pose estimation is included with the audio detection.

Bancud and Palconit researched human pose with machine learning to detect cheating [15]. In their research, there is an external device called Jetson Nano used for image capturing and image preprocessing. A web application was developed using a Python asynchronous framework called Starlette. Inside this framework, Vidgear is used which is a video streaming library. Through wifi, captured video is encoded into images and sent to the computer for analysis to detect cheating. For pose estimation, they used the OpenPose library and for classifying the pose they used the XGBoost library. A prediction of the cheating can be made from the probability of the classification of the pose. Suppose a pose is classified as cheating then one flag will be created and with coordinates of the pose a bounding box will be created in the frame. After detecting all the suspicious poses, the frame will be sent to the database for real-time monitoring. In this proposed model a big drawback is that this model needs an external device for processing the videos.

In most papers, we have read researchers, rather than focusing on one feature, used several ones to accomplish the task of preventing cheating activities. A group of Indian researchers tried to detect cheating by the capture process. The work analysis of Prathish, Narayanan, and Bijlani is that if an examinee opens any search engines, any other window, any software during exam time can be detected by process capture [16]. Also, if a student tries to connect the device with anything like a pen drive or USB drive can be detected by using process capture. For this proposed system writers described that they compared it to a real proctor and the accuracy is 80%.

Moreover, this research paper [16] analyzed the algorithm by X.Zhu for face detection and face landmark localization. They suggested that they can use this algorithm to detect if there are multiple people in the room with the examinee. Also, it was

said that feature points can be used to detect the duration of not being present in front of the device which is clearly an indication of cheating. They showed an experimental result where the experimental output x-axis is the frame number and the y-axis is 1 when a face is present and -1 when there is no face. Furthermore, in this paper, it is said that they can collect audio voice easily from a microphone and with that dataset they can calculate the mean value and peak point of the mean value of the environment. Also, threshold points can be found by calculating the midpoint. So, if at any point the collected data crosses the threshold value then it will be considered malpractice. This audio data is collected at different time points. Finally, an inference system was used to combine all these features. It is said that if any of three is present at any time then for that student it will be said that he is involved in any kind of cheating. The problem of this research is that this system is only for a single student.

To detect malicious activities during the examination, Hu, Jia, and Fu used head posture and mouth state of examinee's which are collected from a webcam [17]. They researched to detect any unusual behavior at the time of online examination. Writers also included abnormal behavior as moving the head frequently while examinee faces the computer screen, all changes of the head pose are considered abnormal. Also, frequent mouth opening and closing were mentioned to be abnormal. At first, to detect a face Adaboost+haar algorithm was used. Then, with the help of a dataset, convolutional neural networks are used to prepare a relation between the collected head pictures and equivalent poses. In this way, the result for head pose estimation was found and that is if yaw angle is greater than 15° among 20 frames consecutively, it was declared as an abnormal behavior[17]. Again, to detect the mouth state, firstly, dlib is used to locate the feature points of the mouth. Then, the help of a Sobel edge detection algorithm was needed to determine if the mouth is open or not. Therefore, using the results from both head pose estimation and mouth state detection whether or not examined behavior is abnormal is detected. In the future, they wish to monitor subtle changes of examinees to create a more efficient abnormal behavior detection model.

Although we have included head pose estimation in our work similar to theirs [17], we also included eye gaze estimation, as a student can face the camera without changing the head pose yet using eye movement can still perform cheating. A student can frequently open and close their mouth without speaking as a habit during the examination, so in contrast to their work [17], we have added audio detection to detect if a student is speaking or not. Also rather than assigning a threshold, we have used hybrid classification where we trained our model to be able to predict cheating attempts.

Similar to our work, in [18] Indi, Pritham, Acharya and Prakasha used eye gaze and head pose to determine a student's attention on the screen. They used a pre-trained FSA-Net model to get three head pose estimates yaw, roll, and pitch. On the condition of whether or not the eye is visible and open, they performed different tasks. They used eye gaze estimation to extract eye gaze estimates for detecting if the eye is invisible or not. Combining these gaze estimates and the head pose angles they pass them as an input to classifier 1-XGBoost. If the eyes are not visible and open, only head pose angles are given to classifier 2 which also works with XGBoost. So, combining dual classifiers, they achieved an accuracy of 96.04%. They also showed an output of their classifier where a student's visual focus of

attention is at the screen or away from the screen can be seen. In addition to eye gaze estimation and head pose estimation, we also included audio detection to detect the presence of voices and other sounds which differs our work from theirs [18]. We also used a hybrid classifier which includes XGBoost and MLP to predict cheating attempts based on eye gaze estimates, head pose angles, and audio parameters.

A comparison among some of the proctoring features is given below using three of the existing software [10], one paper Indi, Pritham, Acharya, and Prakasha’s work in [18] and ours.

Proctoring Features	ProctorU	Respondus	Proctorio	AIProctor	[18]	Ours
Recorded Video Viewing Option	No	Yes	Yes	No	Yes	Yes
Audio Record	No	No	Yes	No	No	Yes
Webcam Required	Yes	Yes	Yes	Yes	Yes	Yes
Voice Recognition	No	No	Yes	No	No	Yes

Table 2.1: Comparison between different features of Online Proctoring software

In summary, there are combinations of various features used in the papers to successfully detect cheating activities during the examination. Some papers like [15] use an external device to detect poses which helps to identify cheating activities. [14] and [16] uses face detection and voice detection to track student’s faces and also checks the presence of multiple faces and voices in the room. Authors in [17] detect cheating attempts of the student by head pose estimation and mouth state estimation. Using eye gaze and head pose estimation, [18] detects the cheating activities of the student. We are going to include the features of eye gaze estimation, head pose estimation, and audio detection to identify cheating attempts of the student.

Chapter 3

Methodology

In this chapter 3 we will be going through all the steps and procedures we followed to conduct our research on online proctoring including data collection. We will describe our research framework which is made of the features head pose estimation, eye gaze estimation, and audio detection step by step. Chapter 4 will describe how we used the resulting parameters found from the estimation and detection to our classification model to predict cheating attempts. Consequently, chapter 5 will discuss our findings found in chapter 4.

As conduction of fair and valid online examination can be achieved by an online proctoring system, the main purpose of our research is to make this system as inexpensive and efficient as possible. Neither it is possible for an invigilator to be present all the time during examination nor a fully automated proctoring software. The first option results in being labor-intensive for the invigilator. Another one that is full automation causes distrust among examinees when incidents like getting penalized for cheating automatically when they claim to be innocent. Also to increase efficiency external hardware is proposed which becomes costly for some people.

Related work done in this field to detect cheating includes using external hardware-like Jetson Nano to capture and process image [15]. Authors in [15] use this external device to estimate pose to detect cheating. Using multiple face detection and audio detection, [14] detects malpractices like-existence of multiple people in the room with the student, the absence of a student's face for a long time, and the presence of the voice in the room. Similarly, [16] uses face detection to continuously track student's faces throughout the exam and also uses voice detection to detect unwanted, talking sounds in the room. Hu, Jia, and Fu use head pose estimation and mouth pose estimation to detect malicious activity [17]. In [18], using eye gaze estimation and head pose estimation Indi, Pritham, Acharya, and Prakasha detect cheating in the exam. In our work, we included eye gaze estimation, head pose estimation, and audio detection as main features to detect cheating attempts.

3.1 System Framework

Our online proctoring system only uses the webcam and microphone which are embedded in laptops or computers. As such, there is no extra cost for external hardware devices. The entire process in order to create our online proctoring system is shown below using the flowchart.

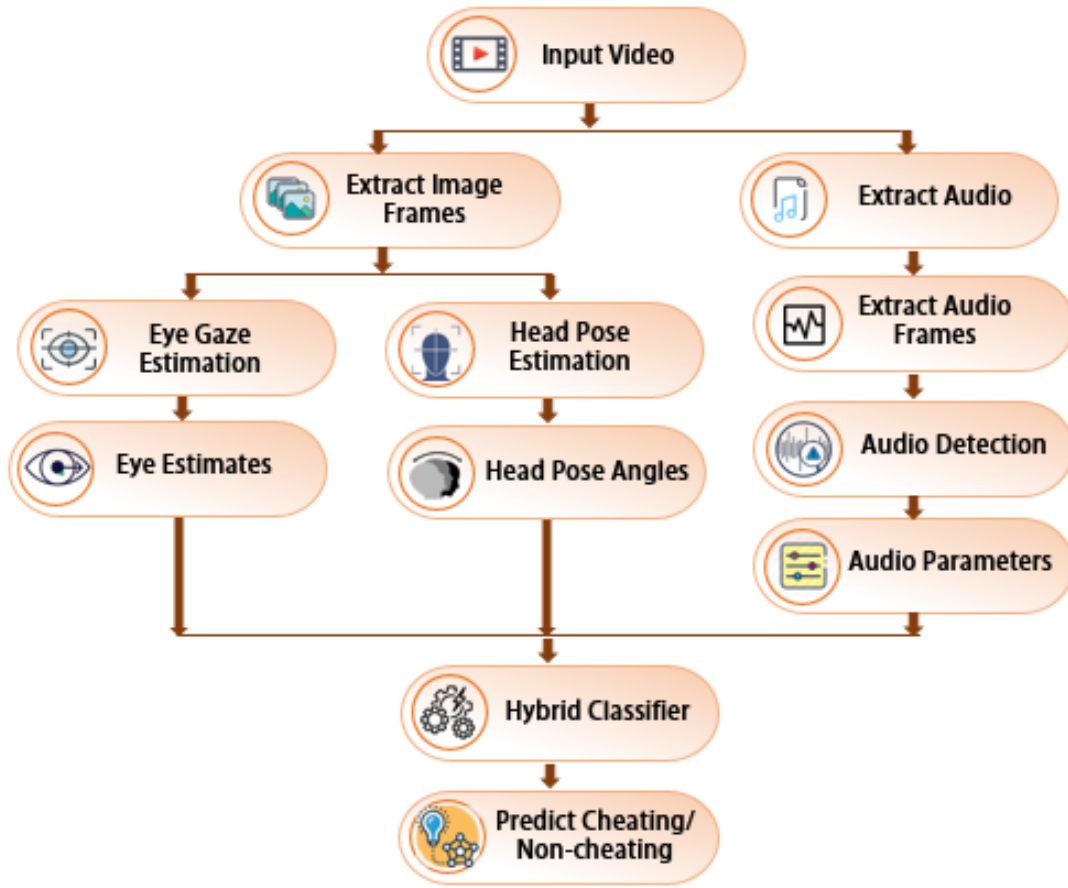


Figure 3.1: Proposed Model

In our proposed system, we will record students' video and audio through their web camera and microphone. From those recorded videos a CSV for each student will be created where all the head and gaze movement's features and audio features are recorded with respect to time. After that, for the prediction of cheating attempts, we are using a hybrid classifier. This classifier will flag the time where the student might be involved in cheating. Finally, a report will be given to the exam invigilator with flagged and unflagged times. As the videos are recorded it is a suggestion to the invigilator to visit the flagged time to make sure of the student's activity.

3.1.1 Constraints of the model

There are some constraints that students need to follow during the examinations. Those are,

1. Students are required to keep their webcam and microphone to be on all the time throughout the examination.
2. Student's front side needs to be visible in front of the camera.
3. The student's are advised to not move their head or eye without necessity.
4. During the exam, no talking is allowed, that is reading questions out loud is not advised.

3.2 Data Collection

Since there is no public dataset on cheating activities of students during examinations, we have created our own dataset for the research purpose. With the help of ten participants, each of them was recorded while performing several online examinations which are available over the internet. They were asked to conduct cheating attempts like- looking at someone, looking at books, talking with someone, etc. On top of that, they also put on some other normal behavior such as thinking about the question and answer while staring anywhere absent-mindedly except the monitor. There were a total of 30 recorded videos; 2-4 videos per participant. The length of the videos which also included audio were 4-5 minutes.

Moreover, to train our system for audio detection different audio was collected from various sources. The audios included a variety of loud noise, animal sound, vehicle sound, etc. Also, male, female voices and whispering sounds were also used in training the audio detection system.

3.3 Data Analysis

Our dataset includes video and audio in each of the recordings which help in our purpose of estimating gaze, head pose, and audio detection. The recordings contain both the cheating attempts and non-cheating attempts making them similar to real-life examinations. The videos helped us in finding gaze estimates and head pose angles. Also, the audio from those videos made it possible to detect human voices and non-human sounds. Frames from those recordings were extracted per second to find the head pose, eye gaze, and audio parameters. Altogether these parameters were used to predict probable cheating attempts of an examinee during the examination. It is to be mentioned that all the detection and classification models we used have been implemented in Python.

To predict cheating attempts, we need the students' head pose angles, eye gaze parameters, and audio parameters to insert into our CSV file. Details on how we found these parameters have been given in the following three sub-section 3.3.1, 3.3.2, and 3.3.3. In 3.3.1, using head pose estimation we have described the method by which we found head pose angles. The 3.3.2 sub-section explains how we found the eye gaze parameters. Using the extracted image frames from the input video, sub-section 3.3.1 and 3.3.2 are used to give further details on finding the eye and head parameters. Lastly, 3.3.3 gives the details on the audio detection techniques. It also describes how the audio parameters are found using the extracted audio frames from the generated audio file from the input video.

3.3.1 Head Pose Estimation

The very first step to implement our proctoring system is estimating the head pose which we were able to do by detecting the head in the extracted images and approximately calculating the head orientation angles. There are three head orientation angles as per Euler's, they are yaw (pan), pitch (tilt), and roll. From figure (3.2), the pitch is the rotation of the head around the horizontal X-axis and the yaw is the rotation of the head around the vertical Y-axis.

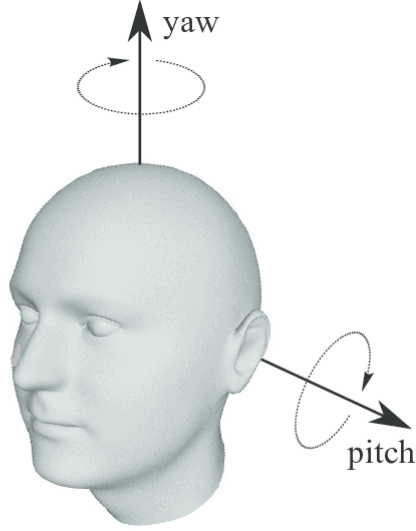


Figure 3.2: Yaw (Pan) and Pitch (Tilt) of human head

In our system, we have applied only these two horizontal and vertical angles to detect and predict the head position in images by following a pre-trained RealHePoNet model, figure (3.3), a single convolutional neural network model [19]. With the help of two datasets, AFLAW and Pointing’04, the researchers of this paper aimed to detect a variety of head pose in real-world conditions. Without the need for various pre-processing steps like facial feature extraction, this model is able to detect head poses in real-world images, not just in laboratory-created images.

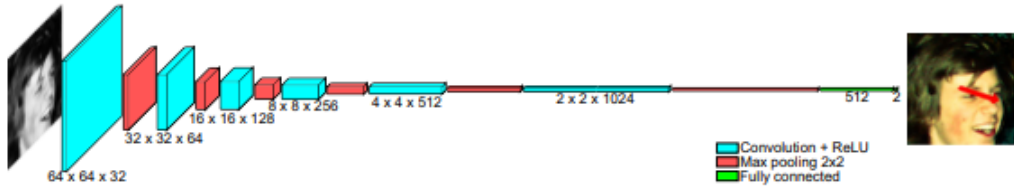


Figure 3.3: System Architecture of RealHePoNet model (taken from [19])

The main reason why we chose this pre-trained model for our cheating attempt detection is that the authors of the paper ensured that the two angles- tilt and pan, are enough to find the focus of a person’s attention. Since our area of research revolves around a student’s focus of attention during the examination, using this model, we obtained the exam giving the student’s head pose angles namely tilt and pan.



Figure 3.4: Estimated Head Poses

The head region of the human face must be located in order to determine the head posture angles. To recognize entire human heads, the head detector model built in the study utilizing neural networks [20] is applied. This trained model is built by using SSD (Single Shot Detector) architecture taken from [21]. The neural network architecture used in the model is constructed like this- six convolutional layers (the first layer starts with 32 layers and the next layer will have a double filter from the previous one) and one fully connected hidden layer that contains 512 neurons. This model has a mean error of 4.37, which the authors believe is low enough to be used in real-world scenarios.

time	tilt	pan
1	-1.214996861	2.898796325
2	0.767066508	2.637812439
3	-6.354745429	8.897277396
4	-3.165942019	5.754368728
5	-0.544341038	5.490780364
6	8.416172175	11.68126657
7	8.670890513	12.25419504
8	-1.452628383	7.146361137
9	1.689636454	7.821932144
10	-7.504458566	8.026369122
11	-8.254367471	12.42624291
12	-14.01543418	12.50147821
13	-4.469436182	3.420153967
14	-13.94415193	12.45692932
15	-2.603215692	9.756490913
16	4.164761706	7.649099304
17	-1.706037363	10.02977243
18	-3.708984101	12.19499432

Figure 3.5: Example of Head Values in CSV file

3.3.2 Eye Gaze Estimation

For the purpose of gaze movement, OpenCV and Dlib (library for machine learning) is used. We have integrated head pose with eye gaze estimation because Wollaston's effect claims that for two different head positions the gaze can remain the same. So, it is seen that people can move their heads without moving their gaze. That is why the head pose is also integrated here [22].

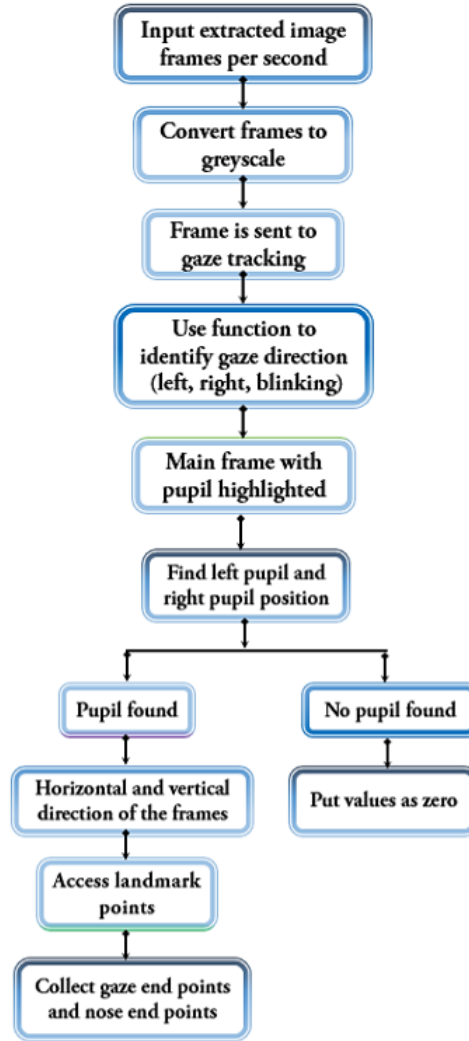


Figure 3.6: Flowchart of the proposed eye algorithm

The process of how OpenCV and Dlib work and serves our purpose for finding eye gaze estimates, figure (3.6) is described below in detail.

For OpenCV to work, at first, the images need to be converted to grayscale. Then the advanced 'Canny Edge Detection' algorithm is used in OpenCV to detect the edge of the pupils [23]. This algorithm removes noise from the image and from the smoothed image detects the compute gradient magnitude and the direction of each pixel. The third step is suppression of the non-maximum pixels that means only

the pixels which have the max value will be kept. Finally, Hysteresis Thresholding will zero out all the pixels beyond the particular threshold. So everything from the picture will be filtered out except the eye regions. Moreover, the Hough transform is used to detect pupil's locations. From that, we need to collect the pupil's movement. In our eye gaze estimation algorithm, for image processing OpenCV and to detect facial landmarks Dlib is used. Dlib can detect 68 facial landmarks but for our research purpose, we have used only six facial landmarks. They are the extreme corner of the lips, nose, chin, and extreme corner of the eyes. After that those landmarks are sent to the OpenCV function to compute the real-world 3D points with the projected 2D images.

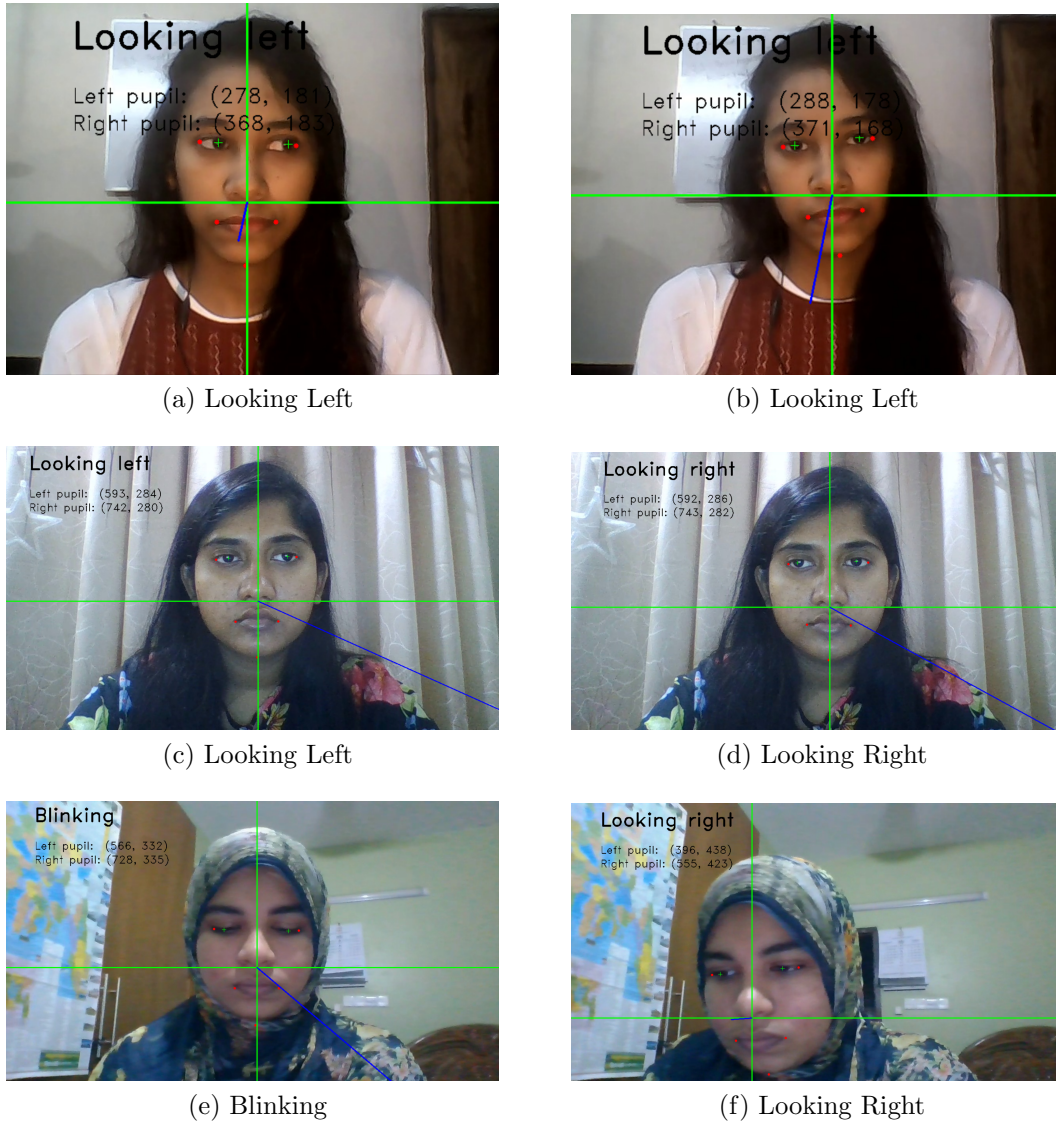


Figure 3.7: Estimated eye gaze parameters

For the calculation of eye gaze estimates, firstly we calculate the eye centers from $leftEyeCenter(x,y)$ and $rightEyeCenter(x,y)$ which are computed with the height and half of the width of the left and right eye landmarks frames which are collected from the Dlib. After that, it is needed to calculate the center of the pupil by finding the centroid $PupilCenterLeft(x,y)$ and $PupilCenterRight(x,y)$ of the image processed and the iris frame. Finally, these will help to calculate the horizontal and vertical ratio of pupil movements.

The equations are:

$$PupilLeft(x) = PupilCenterLeft(x) / (LeftEyeCenter(x) - 2 * 10) \quad (3.1)$$

$$PupilRight(x) = PupilCenterRight(x) / (RightEyeCenter(x) - 2 * 10) \quad (3.2)$$

$$HorizontalRatio = (PupilLeft(x) + PupilRight(x)) / 2 \quad (3.3)$$

$$PupilLeft(y) = PupilCenterLeft(y) / (LeftEyeCenter(y) - 2 * 10) \quad (3.4)$$

$$PupilRight(y) = PupilCenterRight(y) / (RightEyeCenter(y) - 2 * 10) \quad (3.5)$$

$$Verticalratio = (PupilLeft(y) + PupilRight(y)) / 2 \quad (3.6)$$

If the horizontal ratio and the vertical ratio are between 0.0-0.5 then the direction is upper and left and if the ratios are between 0.5-1.0 then the direction is lower and right.

time	blink	right	left	quadrant	left_pupil_x	left_pupil_y	right_pupil_x	right_pupil_y	gaze_center_x	gaze_center_y	r
1	0	0	1	3	559	319	700	323	629	321	
2	0	0	1	3	578	311	720	313	649	312	
3	0	1	0	3	528	313	677	317	602	315	
4	0	1	0	3	497	315	637	315	567	315	
5	0	1	0	3	495	312	635	311	565	311	
6	0	1	0	3	503	309	644	310	573	309	
7	0	0	1	3	528	308	666	309	597	308	
8	0	0	1	3	532	303	671	305	601	304	
9	0	0	1	3	530	297	671	297	600	297	
10	0	0	1	3	535	294	676	294	605	294	
11	0	0	1	3	532	292	673	293	602	292	
12	0	0	1	3	535	293	677	293	606	293	
13	0	0	1	3	578	310	720	312	649	311	
14	0	1	0	3	530	301	675	305	602	303	
15	0	1	0	3	531	305	676	309	603	307	
16	1	0	0	3	536	312	683	315	609	313	
17	1	0	0	3	533	311	683	315	608	313	
18	1	0	0	3	542	311	683	315	612	313	
19	0	0	1	3	544	301	683	305	613	303	
20	0	0	1	3	544	301	684	305	614	303	
21	0	0	1	3	552	302	696	305	624	303	
22	0	1	0	3	535	301	682	306	608	303	
23	0	1	0	3	530	307	677	309	603	308	

Figure 3.8: Example of eye gaze parameters collection in CSV file

It is to be mentioned that this method works in both bright and dark lighting conditions but not in a full dark situation. It also works on any skin color and age group of people while wearing glasses or not wearing glasses.

Therefore, by using this method we get our eye gaze parameters needed for cheating prediction.

3.3.3 Audio Detection

In our proposed model, we have focused on dividing the derived audio into human voice and non-human sound. The human voice is created by the vocal cord, teeth, tongue, etc. The frequency generated by the human voice is different from other sounds around us [24]. So by detecting a number of features from an audio signal we can easily distinguish human sound from other sounds. So, it will help our system to detect if the student is talking with someone during the exam.

time	chroma_s	spectral_c	spectral_b	rolloff	zero_cross	mfcc1	mfcc2	n
1	0.28162	1037.75	1296.08	1666.62	0.04907	-293.036	143.471	
2	0.52066	1580.37	1683.73	3362.61	0.077	-379.229	113.282	
3	0.29036	540.377	709.909	713.777	0.03136	-359.18	154.425	
4	0.32123	1049.97	1190.96	1609.36	0.05342	-191.836	176.717	
5	0.2895	1207.2	850.108	1986.68	0.07756	-197.616	152.934	
6	0.33684	868.866	1441.82	1665.64	0.02836	-512.933	174.17	
7	0.24932	1121.61	781.733	1748.35	0.07138	-216.428	150.812	
8	0.38309	2097.39	1481.75	3259.83	0.14431	-226.32	111.215	
9	0.35141	983.052	1257.07	1561.4	0.04644	-356.423	155.447	
10	0.19872	1292.33	871.398	1854.79	0.08649	-253.105	87.7172	
11	0.1774	1181.25	904.297	1780.89	0.0734	-343.821	77.5462	
12	0.69795	1255.2	1441.92	2099.49	0.04551	-384.472	19.1928	
13	0.44694	1353.91	1220.35	2312.37	0.08104	-346.405	111.723	
14	0.36987	2186.7	1766.39	4132.42	0.13147	-99.5317	114.889	
15	0.27666	1071.78	1182.07	1659.04	0.05621	-281.09	161.878	
16	0.44353	1060.15	1258.97	1916.7	0.03286	-307.502	86.6207	
17	0.31609	1116.86	878.974	1632.85	0.07577	-211.142	193.595	
18	0.26895	1085.18	927.876	1736.11	0.07043	-281.298	173.714	
19	0.21331	906.063	774.723	1325.27	0.05814	-302.174	172.069	
20	0.22881	1012.49	826.43	1417.77	0.06694	-287.455	176.027	
21	0.27585	1001.25	839.748	1506.1	0.06961	-229.582	197.968	
22	0.2784	1013.33	889.875	1385.96	0.07085	-263.102	188.9	
23	0.32135	1342.79	1164.33	1986.19	0.09075	-154.305	150.412	
24	0.25766	1076.61	958.873	1498.03	0.07107	-277.567	174.018	
25	0.22844	990.113	920.415	1489.95	0.06199	-256.91	182.204	
26	0.25822	1012.79	933.146	1522.74	0.06471	-203.347	188.805	
27	0.23145	774.202	654.916	1276.09	0.05121	-240.903	219.078	

Figure 3.9: Example of audio parameters collection in CSV file

In this phase, we extracted the audio which is a “.wav” file from our collection of videos. After that, we divided the audio files into frames per second. From those frames, we collected the sound features which are chroma short time fourier transform, spectral centroid, spectral bandwidth, spectral roll-off, zero-crossing rate and 20 MFCCs. Spectral centroid, spectral bandwidth, spectral roll-off, zero-crossing rate are the low-level features and 20 MFCCs are the mid-level features. To extract the attribute from the provided frames, we utilized Librosa. Librosa is a Python module that allows you to analyse audio signals and extract features from them. This package is very useful for audio analysis. Here the given audio segments are again divided into small chunks. As a result, it returns an array for each feature. Finally, we calculate the mean of all values for each array for the corresponding features. Below, the resulting sound features are explained in more detail.

Chroma based features detects the keys or chords of an audio at a short-time window frame [25]. Chroma_stft where stft stands for Short Time Fourier Transform, extracts chroma spectrogram or chromogram. The equation for the short time fourier transform,

$$F_g f[n, k] = \sum_{m=0}^{M-1} f[n - m]g[m]\varepsilon_k[m] \quad (3.7)$$

where,

$$\varepsilon_k[m] = e^{-2\pi m \frac{k}{N}} \quad (3.8)$$

Here, The uniformly sampled functions are $f[n]$ and $g[n]$. Window function is denoted as g and the window length of g is M . Number of samples are denoted as N .

Some of the most common spectral-domain features are spectral centroid, spectral bandwidth, and spectral roll-off. Spectra centroid [26] finds the center of gravity of each spectrum with the help of Fourier transformation of frequency and weighted mean of the frequency. For the j^{th} frame spectral centroid: In the frequency domain S'_j is the signal of the j^{th} frame after doing proper framing and Fourier transformation. N is the total number of chunks in a frame and F denotes that the features are extracted in the frequency domain.

$$FC_j = \frac{\sum_{i=0}^{N-1} \left(|S'_j(i)|^2 \times i \right)}{\sum_{i=0}^{N-1} \left(|S'_j(i)|^2 \right)} \quad (3.9)$$

Spectral bandwidth is mainly the difference between upper and lower frequencies in a spectrum. The equation which helps to identify spectral bandwidth,

$$FB_j = \sqrt{\frac{\sum_{i=0}^{N-1} \left(|S'_j(i)|^2 \right) \times (i - FC_j)^2}{\sum_{i=0}^{N-1} \left(|S'_j(i)|^2 \right)}} \quad (3.10)$$

The zero-crossing rate feature is a time-domain feature [27]. The X-axis has time and the y-axis has an amplitude that is why it is called time-domain features. It is one of the simplest features. It determines the sign changes of the signal in one second. As a result, it counts how many times the signal's sign shifts from positive to negative and vice versa. Also, it can be said that the number of times the signal crosses point zero in one second. For the j^{th} frame, zero-crossing rate: in the time domain S_j is the signal for the j^{th} frame,

$$ZCR = \sum_{i=1}^{N-1} |\text{sgn}(S_j(i)) - \text{sgn}(S_j(i-1))| \quad (3.11)$$

Spectral roll-off is the frequency that is 85% below the total frequency. The equation for Spectral roll-off frequency,

$$\sum_{i=0}^{FR_j} |S'_j(i)|^2 = 0.85 \times \sum_{i=0}^{N-1} |S'_j(i)|^2 \quad (3.12)$$

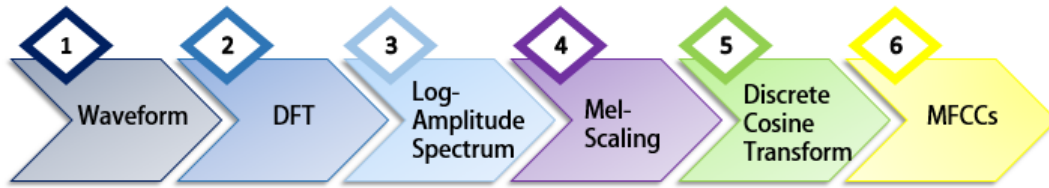


Figure 3.10: Extracting method of MFCCs

In figure (3.10), we are extracting 20 small sets of Mel-frequency cepstral coefficients (MFCC) features, these 20 features all together describe the whole spectrum envelope shape. First of all, the signal is converted from analog to digital. After that Fourier transformation is applied to the given signal. As a result the signal moves from the time-domain (Amplitude vs time) to the frequency domain (Amplitude vs frequency). Then the spectrum is sent to the log function to get the Log-Amplitude spectrum. Now, we will apply the Mel filter bank to the log spectrum for Mel-scaling. As a result, we now have Mel-spectrum. The final step is to apply the discrete cosine transform to the Mel-spectrum and create MFCCs.

Chapter 4

Experimentations

4.1 Data Pre-processing

The collected data are labeled zero and one very carefully with respect to time. Label ‘1’ indicates that the student is involved in any unfair means where label ‘0’ indicates the student gave the exam honestly. If the student’s head movement or eye gaze is suspicious or if there is any human voice at that particular time, then the time is labeled one, otherwise zero.

The dataset was divided into two parts: training and testing. We used 80% of our dataset for training and 20% for testing in this experiment. Later, we combined the records of all the training videos to train our hybrid model. Also, we made sure to remove all the redundant data from our training CSV so that there is no overfitting in our model. Moreover, since we are taking different types of features, we had to scale all the features. We followed the normalization of the min-max scaling technique and kept all our features within the range zero to one. As a result, our features which are measured in different scales do not create any problem in learning the model. The equation used for scaling,

$$x_{\text{scaled}} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (4.1)$$

Along with that, it was seen that some features have a strong correlation with each other. So, for reducing complexity we prefer to remove one of the two features having a correlation greater than 0.9. Then we divided all the properties in the CSV into independent features and dependent variables. Besides ‘label’, all the properties are independent features as the value of ‘label’ depends on the value of eye estimates, head pose angles and audio parameters. So ‘label’ is the dependent variable here.

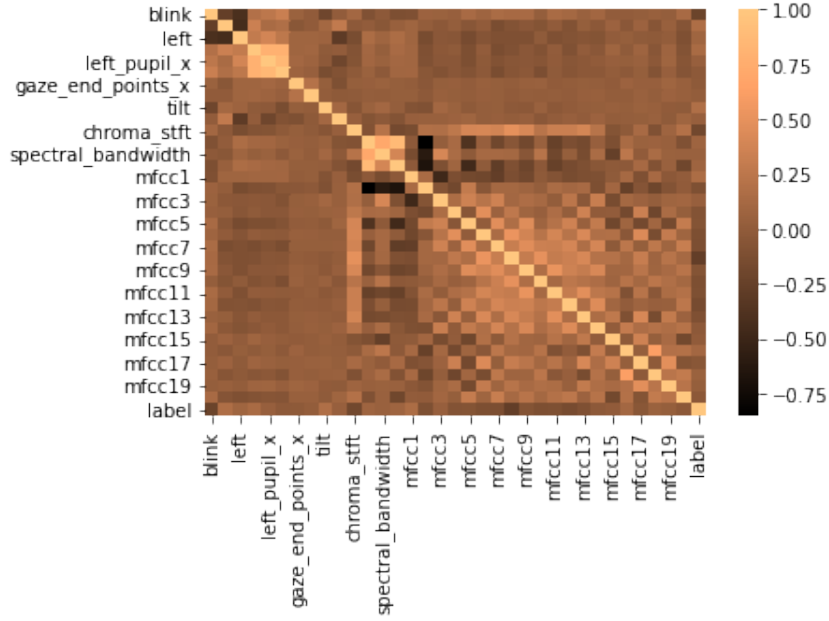


Figure 4.1: Correlation of the Features

So before going into the classification part, in the data pre-processing section, all the eye estimates, head pose angles, and audio parameters are combined in one CSV file. This file also contains a 'label' whose value consists of 0 or 1. If a student has shown abnormal behavior or suspicious head and eye movement like a student's focus of attention is outside of the screen monitor. The presence of the human voice is also regarded as label 1. So label value 1 is given if they show an abnormal gesture and if there is a human voice and label value 0 is given if the student's behavior is normal (Example: student's attention is on the screen) and there is no human voice. After preparing this CSV file, this is given to our classifier to predict the outcome.

A hybrid approach by combining two classifications was followed to predict the outcome of our work. Extreme Gradient Boosting otherwise known as XGBoost was used as the first classifier and Multilayer Perceptron (MLP) was used as the second classifier.

Classifier 1: XGBoost: XGBoost is a gradient boosted decision tree implementation in which trees are layered on top of each other to rectify prior trees' errors. So we have our main model and we also have multiple versions of our model and all of them are chained together. While performing classification, each tree boosts the attributes which caused miss-classification of previous trees. One of the features of XGBoost is that it can automatically figure out missing values in a dataset and it regularizes the features to reduce overfitting. Also, in every iteration, cross-validation can be performed. When the model is training, in every step, performance can be evaluated and its accuracy can be monitored. So, when we can see that doing further iteration is not changing our performance we can stop it and find optimal points quickly. Moreover, XGBoost builds the sequential trees by the parallel process, and also it has the ability to prune unnecessary edges of the tree. As a result, the algorithm has become faster than others [28].

There are some XGBoost hyperparameters like Booster, Eta, Max_depth, etc. whose values eventually determine the learning process of the algorithm. In every step of training, the values are adjusted to achieve maximum accuracy.

The parameters we tuned in Xgboost for better training and testing results are `max_depth`, `subsample`, `n_estimators`, `learning_rate`. We kept all the other parameters' default values. 'max_depth' defines the maximum number of the depth of the trees which helps to reduce the overfitting as a high depth tree will focus on only certain features. Before growing the tree 'subsample' helps to randomly sample the training data. If the subsample is very low then it will under-fit the model or giving a high value will over-fit the model. The parameter 'n_estimators' is mainly the number of trees. We give a proper value of n_estimators so that it does not create unnecessary trees. 'Learning_rate' which is also known as 'eta' shrinks the weights in every step, a low value of 'eta' helps our classifier to get the optimum value.

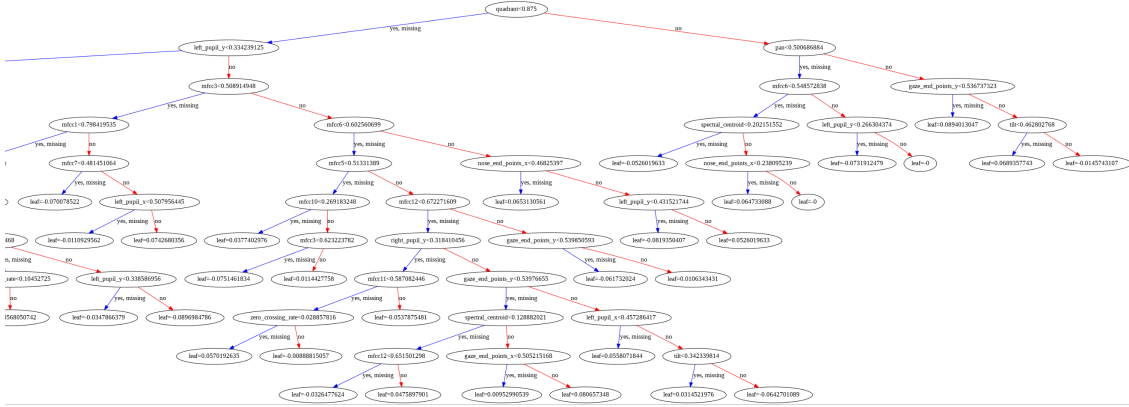


Figure 4.2: Small part of the tree

Now after we train our XGBoost model for classification, we get the leaf nodes of the tree as predicted values. We used those leaf nodes as an extra parameter and combined them with other features of the dataset when we import the CSV file in our second classifier, MLP.

Classifier 2: MLP: The basic unit of the neural network is generally described as a node. A single layer perceptron is defined as a node or neuron which has some weighted inputs and these inputs are multiplied by their weights and with the help of an activation function (Example: Rectified Linear Units (ReLU), Sigmoid function, etc.) output is generated. After data is inputted, the activation function uses a calculation to make a decision regarding the inputs and thus produces that decision or prediction as output. Composed with several perceptrons, a multilayer perceptron is built where data is inserted into the input layers, after completing some processing in hidden layers, the result or prediction against the input is generated from output layers [29].

So, MLP is an Artificial Neural Network (ANN) which means it depends on a neural network for classification. In MLP there are some multiple-input layers and output layers that are fully connected and between these layers there are multiple hidden layers. Usually, MLP is applied to supervised learning problems where dependencies between inputs and output are learned and applied to a model. In the training phase, parameters, weights, biases (constant added to the input to shift activation function to predict better with the input data), etc are adjusted to achieve maximum accuracy from the model. Using the backpropagation method, the weights, biases, etc. are continuously adjusted to reduce the error.

In our model, we use the activation function ReLU for the hidden layers as it helps

to decrease the vanishing gradient problem and gives a better result. For weight optimization across all the nodes, we tune the parameter solver as 'lbfgs'. Also, tune the max_fun for calling the maximum number of the loss functions. Giving an appropriate value of alpha reduces the chance of overfitting and underfitting of our data. We also set the number of epochs by assigning values in max_iter until the convergence. The number of the loss functions is greater than or equal to the max_iter. Moreover, using two hidden layers with the right number of nodes gives the best result for our data. We also set a value for the parameter random_state as it decides the number for reproducing the same results. Finally, we get 100% training accuracy and 97% accuracy for our testing data.

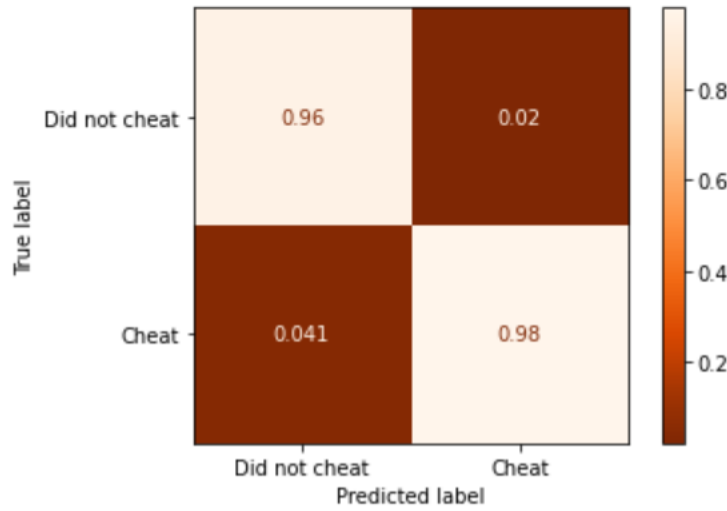


Figure 4.3: Confusion matrix for hybrid classifier

The hybrid model gives 98% true negative result which is cheating and 96% true positive result which is non-cheating. On the other hand, it misclassified 0.041% times for cheating and 0.02% times for non-cheating.

It is also observed that if the data is only used in XGBoost classifiers then the training accuracy is 100% and the testing accuracy is 92%. Moreover, the data is also trained in a multilayer perceptron neural network with ReLU activation function, and keeping two hidden layers gets training accuracy 100% but the testing accuracy is 88%. XGBoost classifies 94% cheating data and 89% non-cheating data correctly. Also, it identifies 11% of cheating data as non-cheating data and 89% of non-cheating data as cheating data. Where MLP is able to predict 0.059% of cheating data correctly and 86% of non-cheating data perfectly. Furthermore, it misclassified 14% of cheating data as non-cheating data and 0.097% of non-cheating data as cheating data. The confusion matrix (figure 4.4) of these two classifiers gives a better understanding.

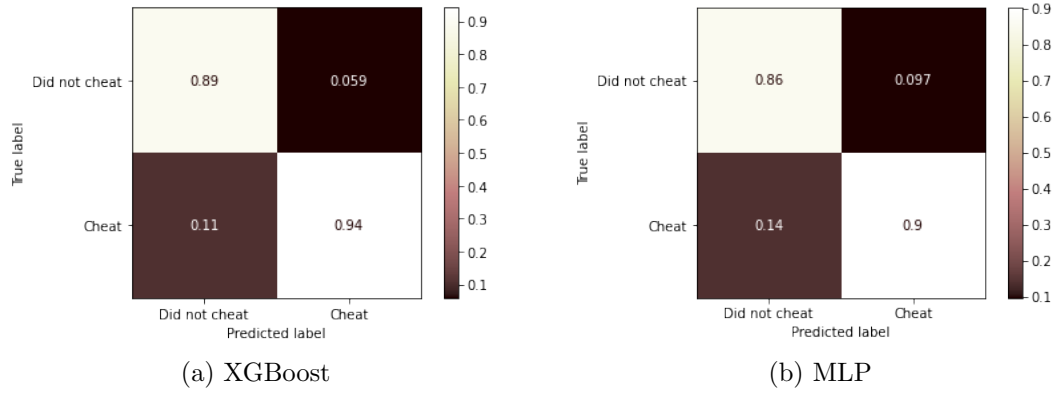


Figure 4.4: Confusion matrix for XGBoost and MLP classifier

Along with that we train our data in some other commonly used machine learning algorithms. One of the well-known binary classification models is Logistic Regression where using the probability concept, logistic regression takes a logistic function to predict 0/1 from the independent variables [30]. It also uses a cost function which minimizes the error. Sigmoid function is the cost function used to create a map from predicted values to probability. This way the logistic regression creates an accurate model. The Logistic Regression model gave a total testing accuracy of 74%.

Another effective algorithm is Support Vector Machine (SVM). If N is the number of features, Support Vector Machine (SVM) creates a decision boundary or hyperplane in N -dimensional space between classes [31]. When we add additional sample points, we can more successfully classify them by ensuring that the distance between the sample points of the classes is as little as possible. This Support Vector Machine (SVM) gave 76% testing accuracy.

Based on different conditions, the decision tree splits a data set. There are decision nodes where the data is split and also there are leaves which are decisions or the prediction of the decision tree [32]. The Decision Tree Model predicted the class by just learning some decisions from prior experience, this classifier gave online exam proctoring data an 80% testing accuracy.

The confusion matrix of Logistic Regression is showing 78% true negative values and 71% true positive values. Where the SVM can recognize 78% cheating values correctly and 71% non-cheating values correctly. On the other hand, Decision Tree can predict 88% cheating values as cheating and 79% non-cheating values as non-cheating. Figure (4.5) also shows these values.

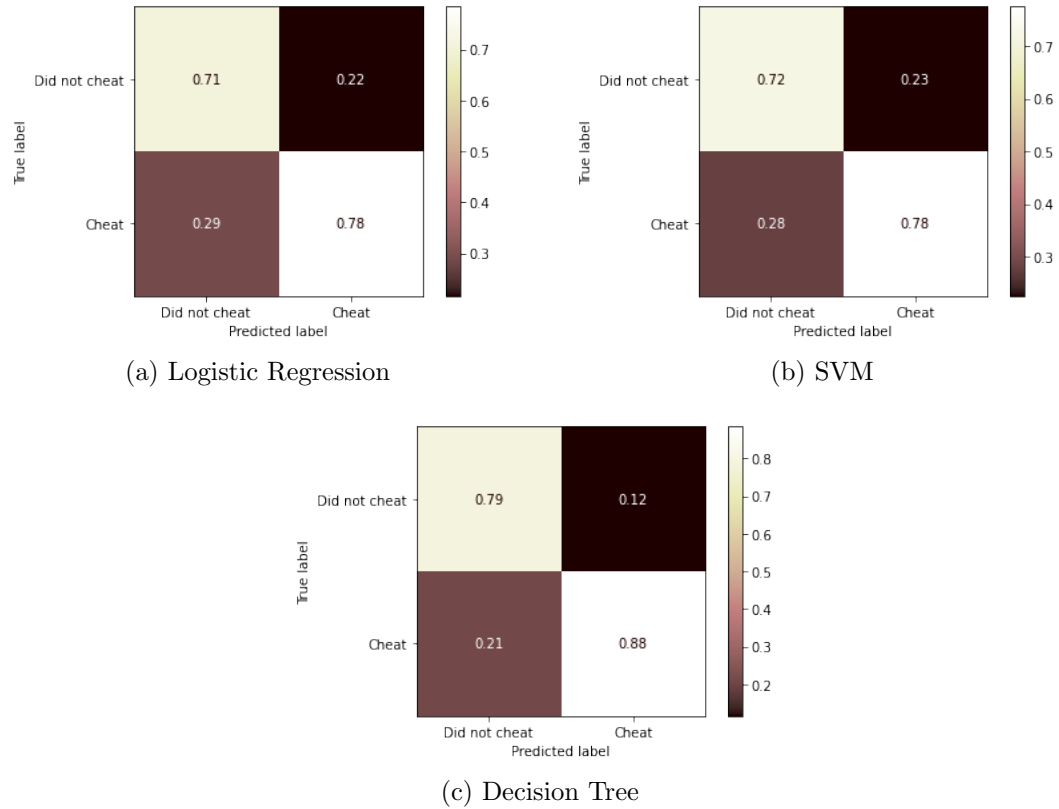


Figure 4.5: Confusion matrix for Logistic Regression, SVM and Decision Tree

Figure(4.6) shows the comparison between accuracy of the classifiers, Logistic Regression, Support Vector Machine (SVM), Decision Tree, XGBoost, Multilayer Perceptron (MLP) and our hybrid model.

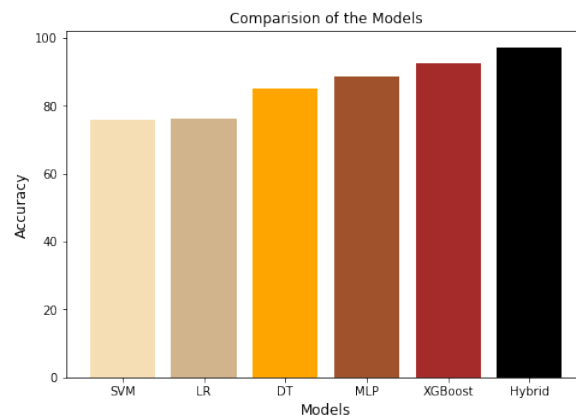


Figure 4.6: Comparison of the models

A Receiver Operating Characteristic Curve(ROC) is also provided (figure 4.7) which clearly demonstrates that the hybrid model has the highest AUC among all the other models.

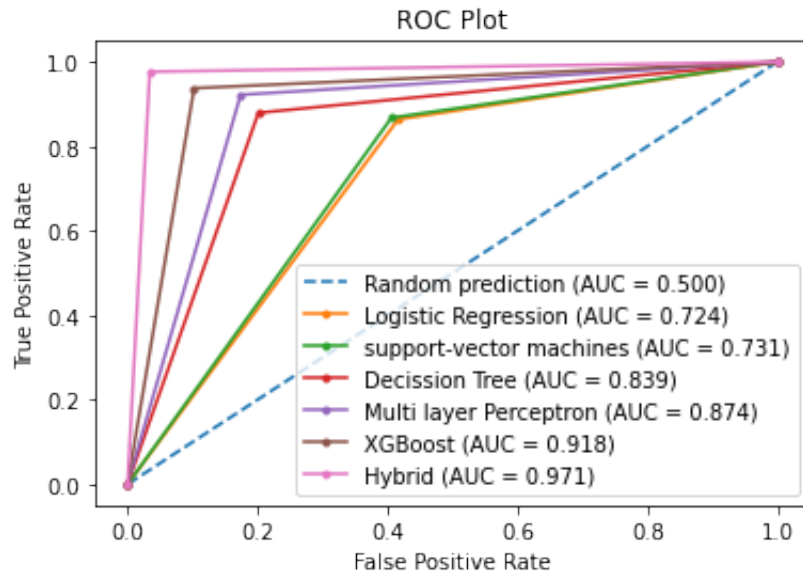


Figure 4.7: Comparison of the model with ROC curve

Chapter 5

Result Analysis

Chapter 5 represents a discussion on the experimental results based on the collected data from chapter 3. This chapter will provide a thorough discussion on the major findings and the last chapter 6 where the conclusion as well as recommendations for future development will be given. This chapter will also include our main contributions in solving the research problems discussed in chapter 1.

The primary objective of this study was to identify a low-cost approach to the most common problem in online examination which is cheating. For our proposed system, we only need a webcam and microphone which are already embedded in laptops. So, there is no extra hardware cost for our proctoring system as opposed to [4] and [15] where external devices like Microsoft Kinect and Jetson Nano were used respectively.

From our collected data, we obtained head pose angles, eye gaze, and audio parameters. Then these values were trained using hybrid classifiers to predict cheating and non-cheating. Our hybrid classifier gives an accuracy of 97% which is better than using only a single classifier. As can be seen from chapter 4, XGBoost which we used as our classifier 1, gives an accuracy of 92%. Also, our classifier 2, MLP gives an accuracy of 88%. Moreover, we have done experiments using several algorithms, Logistic Regression, Support Vector Machine, and Decision Tree and from the experiments, we gained accuracy of 74%, 76%, and 80% respectively. When used individually, the mentioned classifiers do not provide suitable accuracy. However, the combination of XGBoost and MLP makes our hybrid classifier delivers a very high accuracy which is 97%. In online examinations, to predict cheating activities efficiently, this level of accuracy was needed. So, our use of hybrid classifiers serves our purpose to have an accurate and efficient online exam proctoring system.

Moreover, we combined three detection techniques- head pose, eye gaze, and audio. Thus, cheating activities that involve moving the head and eye frequently away from the computer screen and also talking with someone can be detected using our proposed model. From the related works in chapter 2, we can see that researchers applied many techniques to detect cheating. For example, [14] uses face detection and audio detection, [17] uses head pose and mouth state detection, and [18] uses head pose and eye detection to identify cheating activities. The use of head pose, eye gaze, and audio detection to predict cheating activities are very hard to find. So we tried our best to combine these three detection methods to contribute to the online proctoring related research.

One of the main research problems in online proctoring is that automation of the

system does not deliver accurate cheating detection results for every student. Some students may look away from the computer screen for a few seconds without any ill intention, yet the proctoring system flags the student for cheating. In our research, after predicting with hybrid classifiers, we provide a report containing the time period along with the label 0/1 (0 = non-cheating, 1 = cheating) to the invigilator. If a student is flagged for cheating and claims to be wrongly accused, the invigilator can look for that particular time in the recorded video to see whether he is cheating or not. So, even though our system is not fully automated, we propose to solve this efficiency problem in the proctoring systems by cross-checking with the invigilator. In this way, the students will also be less reluctant in using proctoring software in online examinations.

time (sec)	label
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	1
10	1
11	1
12	0
13	1
14	0
15	0
16	1
17	0
18	0
19	0
20	1
21	0

Figure 5.1: Report for Invigilator

Though we managed to contribute in this field of study by giving a low cost and efficient approach of combining head pose, eye gaze, and audio detection and finally use of hybrid classifier to predict cheating activities, we also faced some limitations. As there was no readily available dataset for this particular research, we needed to create our own dataset. Given more data, our classification would be more accurate in predicting cheating activities. Since our proposed model works on recorded videos, we extracted eye gaze, head pose, and audio features serially. However, we plan to move our proposed proctoring system to real-time. There, in the same time duration, multiple students will take place in the examination. So, for that intention, our model needs to extract our parameters for multiple students parallelly. When our proctoring system will proceed towards real-time, for all students, the eye gaze, head pose, and audio features will be extracted and thus predict cheating parallelly. At the same time, the video will also be recorded. So, the invigilator will not need to wait for the report containing cheating and non-cheating label with respect to time,

to crosscheck with the recorded videos for cheating detection.

Chapter 6

Conclusion and Future Works

In un-proctored online exams, the examinees get the opportunity to perform unfair activities and thus authenticity of online examinations as well as online education gets ruined. However, a proctored exam is capable of stopping these cheating activities. So, to have a valid online examination online proctoring system is very much needed. Even though there are many available online proctoring systems out there, academic institutions sometimes cannot afford them. Due to their large expense and sometimes efficiency-related issues, institution and also students do not feel comfortable using them. There are many cases where the system flags cheating attempts wrongly. To solve this, live proctoring with an invigilator present throughout the exam can be an option, but this task is also very labor-intensive. Therefore, the main objective of our research was to address this research problem which lies in this area.

To detect unfair activities during the exam, researchers Raj, Narayanan, and Bijlani [14] and Prathish, Narayanan, and Bijlani [16] used face detection and audio detection as key features for their proctoring system. Bancud and Palconit [15] used external hardware to estimate pose for cheating detection. With head pose estimation and mouth state detection, Hu, Jia, and Fu [17] and Indi, Pritham, Acharya, and Prakasha [18] uses eye gaze and head pose to detect abnormal behavior of students during the exam. As there are not many related works using the combination of eye gaze, head pose, and audio detection, in the work we used these three as main features to detect probable cheating attempts.

The very first we took was to prepare a suitable dataset which was used to extract head pose angles- tilt and pan, eye gaze parameters, and audio parameters. Our proposed model took a hybrid classifier approach by combining XGBoost as the first classifier and MLP as the second classifier. Using the head pose, eye gaze, and audio parameters, our model took help from this hybrid classifier to predict cheating and non-cheating attempts. Finally, we provide a report where cheating and non-cheating activity were flagged and also the time when such an incident occurred. With the help of this report, the invigilator can crosscheck with the video with the mentioned time to detect cheating accurately.

In summary, our proposed model takes a student's head pose and eye gaze estimates, and audio information to successfully detect suspicious activity. The use of hybrid classifiers to train our dataset also results in giving fair predictions in detecting cheating activities. Going forward, our final target is to develop an online proctoring system like the existing ones. As educational institutions face a lot of problems

to conduct online examinations due to student's cheating behaviors, we want to solve this problem by creating an online proctoring software. In the future, we will implement this software to help the institutions to detect cheating attempts of the students and thus create a valid and fair online examination.

Bibliography

- [1] E. Bilen and A. Matros, “Online cheating amid covid-19,” *Journal of Economic Behavior Organization*, vol. 182, pp. 196–211, 2021, ISSN: 0167-2681. DOI: <https://doi.org/10.1016/j.jebo.2020.12.004>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167268120304510>.
- [2] S. S. Chua, J. B. Bondad, Z. R. Lumapas, and J. D. L. Garcia, “Online examination system with cheating prevention using question bank randomization and tab locking,” in *2019 4th International Conference on Information Technology (IncIT)*, 2019, pp. 126–131. DOI: 10.1109/INCIT.2019.8912065.
- [3] C. Watkins, *A student says test proctoring ai flagged her as cheating when she read a question out loud. others say the software could have more dire consequences.* <https://sportsgrindentertainment.com>, 4th October 2020.
- [4] Z. Fan, J. Xu, W. Liu, and W. Cheng, “Gesture based misbehavior detection in online examination,” in *2016 11th International Conference on Computer Science Education (ICCSE)*, 2016, pp. 234–238. DOI: 10.1109/ICCSE.2016.7581586.
- [5] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, “Automated online exam proctoring,” *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609–1624, 2017. DOI: 10.1109/TMM.2017.2656064.
- [6] X. Li, K.-m. Chang, Y. Yuan, and A. Hauptmann, “Massive open online proctor: Protecting the credibility of moocs certificates,” in *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW ’15, Vancouver, BC, Canada: Association for Computing Machinery, 2015, pp. 1129–1137, ISBN: 9781450329224. DOI: 10.1145/2675133.2675245. [Online]. Available: <https://doi.org/10.1145/2675133.2675245>.
- [7] F. L. Cathy Li, *The covid-19 pandemic has changed education forever. this is how*, <https://www.weforum.org>, 29th April 2020.
- [8] D. Newton, *Another problem with shifting education online: Cheating*, <https://hechingerreport.org>, 7 th August 2020.
- [9] D. F. H. Layman, *Another problem with shifting education online: Cheating*, <https://www.caveon.com>, 13th March 2013.
- [10] M. J. Hussein, J. Yusuf, A. S. Deb, L. Fong, and S. Naidu, *Open Praxis*, vol. 12, no. 4, pp. 509–525, 2020, ISSN: 2304-070X. [Online]. Available: <https://search.informit.org/doi/10.3316/informit.620366163696963>.

- [11] H. S. G. Asep and Y. Bandung, “A design of continuous user verification for online exam proctoring on m-learning,” in *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, 2019, pp. 284–289. DOI: 10.1109/ICEEI47359.2019.8988786.
- [12] A. Javed and Z. Aslam, “An intelligent alarm based visual eye tracking algorithm for cheating free examination system,” *International Journal of Intelligent Systems and Applications*, vol. 5, pp. 86–92, Sep. 2013. DOI: 10.5815/ijisa.2013.10.11.
- [13] C. Y. Chuang, S. D. Craig, and J. Femiani, “Detecting probable cheating during online assessments based on time delay and head pose,” *Higher Education Research & Development*, vol. 36, no. 6, pp. 1123–1137, 2017. DOI: 10.1080/07294360.2017.1303456. eprint: <https://doi.org/10.1080/07294360.2017.1303456>. [Online]. Available: <https://doi.org/10.1080/07294360.2017.1303456>.
- [14] R. V. Raj, S. A. Narayanan, and K. Bijlani, “Heuristic-based automatic online proctoring system,” in *2015 IEEE 15th International Conference on Advanced Learning Technologies*, 2015, pp. 458–459. DOI: 10.1109/ICALT.2015.127.
- [15] G. E. Bancud and E. Palconit, *Human pose estimation using machine learning for cheating detection*, Apr. 2021. DOI: 10.13140/RG.2.2.12686.28481.
- [16] 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. DOI: 10.1109/INFOSCI.2016.7845315.
- [17] S. Hu, X. Jia, and Y. Fu, “Research on abnormal behavior detection of online examination based on image information,” in *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 02, 2018, pp. 88–91. DOI: 10.1109/IHMSC.2018.10127.
- [18] C. Indi, V. Pritham, V. Acharya, and K. Prakasha, “Detection of malpractice in e-exams by head pose and gaze estimation,” *International Journal of Emerging Technologies in Learning (iJET)*, vol. 16, no. 08, pp. 47–60, 2021, ISSN: 1863-0383. [Online]. Available: <https://online-journals.org/index.php/i-jet/article/view/15995>.
- [19] R. Berral-Soler, F. Madrid-Cuevas, R. Muñoz-Salinas, and M. Marín-Jiménez, *Realheponet: A robust single-stage convnet for head pose estimation in the wild*, Nov. 2020.
- [20] M. J. Marin-Jimenez, V. Kalogeiton, P. Medina-Suarez, and A. Zisserman, *Laeo-net: Revisiting people looking at each other in videos*, 2019. arXiv: 1906.05261 [cs.CV].
- [21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham: Springer International Publishing, 2016, pp. 21–37, ISBN: 978-3-319-46448-0.
- [22] N. L. Kluttz, B. R. Mayes, R. W. West, and D. S. Kerby, “The effect of head turn on the perception of gaze,” *Vision Research*, vol. 49, no. 15, pp. 1979–1993, 2009, ISSN: 0042-6989. DOI: <https://doi.org/10.1016/j.visres.2009.05.013>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0042698909002430>.

- [23] M. Y. Nam, P. K. Rhee, and C. S. Lee, “A real time pupil location system using multiple illuminations clustering under varying environment,” in *2009 Digest of Technical Papers International Conference on Consumer Electronics*, 2009, pp. 1–2. DOI: 10.1109/ICCE.2009.5012398.
- [24] S. Doshi, *Music feature extraction in python*, <https://towardsdatascience.com>, 30th December, 2018.
- [25] A. Shah, M. Kattel, A. Nepal, and D. Shrestha, “Chroma feature extraction,” Jan. 2019.
- [26] U. Nam, *Special area exam part ii*, <https://ccrma.stanford.edu>, 28th April, 2001.
- [27] T. Giannakopoulos and A. Pikrakis, *Introduction to audio analysis*, <https://www.sciencedirect.com>, 2014.
- [28] J. Brownlee, *A gentle introduction to xgboost for applied machine learning*, <https://machinelearningmastery.com>, 17th February, 2021.
- [29] N. K. Kain, *Understanding of multilayer perceptron (mlp)*, <https://medium.com>, 22th November , 2018.
- [30] A. Pant, *Introduction to logistic regression*, <https://towardsdatascience.com>, 22th January, 2019.
- [31] R. Gandhi, *Support vector machine — introduction to machine learning algorithms*, <https://towardsdatascience.com>, 7th June, 2018.
- [32] *Decision trees for classification: A machine learning algorithm*, <https://www.xoriant.com>, 7th September, 2017.