Thesis Project Report
on

# A Comprehensive Multimodal Approach for Detecting Malpractice in Online Exams

*submitted by*
YETURI VENKATESH
Roll No.224102324

*For the partial fulfillment of requirements for the degree of*
***Master of Technology***
*in*
***Signal Processing and Machine Learning***

*under the supervision of*
Dr. Anirban Dasgupta
Assistant Professor


Dept. of Electronics and Electrical Engineering
IIT Guwahati

May, 2024

# Acknowledgement

I would like to extend my sincere gratitude to my supervisor Dr. Anirban Dasgupta for his invaluable guidance and continuous supervision. His unwavering commitment and extensive insights have been instrumental in shaping my thinking abilities, ultimately contributing to the successful completion of the project and this report. His mentorship has truly been a cornerstone of my academic journey. I am grateful for the contributions of all those students who generously provided data for training various machine learning models in my project. Additionally, I extend my appreciation to PhD scholar Mr.Rahimul Mazumdar for his assistance in acquiring necessary hardware requirements, such as GPUs, whenever they were needed. Special thanks are extended to my sister Sai Santhoshi Yeturi for her invaluable contribution in data collection which made my intermittent presentations/reports more professional.

Also, I would like to acknowledge the continuous mental support of my parents and friends throughout this journey of my MTech at IIT Guwahati.

**Abstract**

This research work addresses the issue of identifying malpractice in online examinations using a novel multimodal approach utilizing several machine learning techniques. While existing online proctoring systems monitor factors like browsing, they often overlook behavioral attributes. The work utilizes FaceNet for face verification, addressing authentication concerns at intermittent intervals. Additionally, we also integrate face spoof detection. The use of external assistance is identified during the exam through the temporal variations of head pose, eye gaze, and mouth states. We also integrate plagiarism checks in answer sheets, multiple device logins, and screen sharing. The model shows reasonable accuracies in potential misconduct detection through the temporal patterns of head pose and eye gaze variations, paving the way for a new introspection into malpractice detection.
**Keywords** — Online proctoring system, Malpractice detection, Face recognition, Head pose estimation, Eye gaze estimation.

# Contents

# List of Figures

# List of Tables

# 1 Introduction

The rise of online learning marks a great shift in the landscape of education, with increased accessibility and flexibility for learners worldwide [1]. Yet, amidst this digital revolution lies tough challenges, none more pressing than the issue of malpractice in online examinations [2]. Cheating, plagiarism, and other forms of academic dishonesty seriously threaten the integrity of educational assessment thus leading to the erosion of trust in academic qualifications and undermining the credibility of institutions.

## 1.1 Motivation

Continuous monitoring by human proctors is the common method for maintaining exam integrity, yet it's time-consuming and challenging. Alternatively, recording and offline analysis of individual videos are also cumbersome. Hence, automated proctoring solutions emerged to address these limitations. However, the existing commercial systems only look at a few aspects, such as screen sharing, browsing other tabs, and logins through multiple devices. As such, candidates still bypass such systems. This motivates the creation of a robust system for comprehensive identification of different malpractice types. To enhance exam integrity, a comprehensive automated proctoring system could integrate advanced AI algorithms capable of detecting subtle cues indicative of malpractice, such as eye movement patterns, facial expressions, and voice analysis for signs of stress or deception. Additionally, incorporating machine learning models trained on a diverse range of cheating behaviors could further bolster the system's effectiveness in identifying and deterring unethical practices during exams. The development of such a robust proctoring system requires an understanding of the different malpractices.

## 1.2 Types of Online Malpractices

Based on the literature [3], the most common types of malpractices can be categorized into three broader categories, as shown in Fig. 1. The first category is impersonation, implying someone else is appearing on the candidate's behalf in the examination. The second category involves taking external help such as using external devices, collaborating through screen sharing, or asking members present in the room but outside the camera's field of view. The

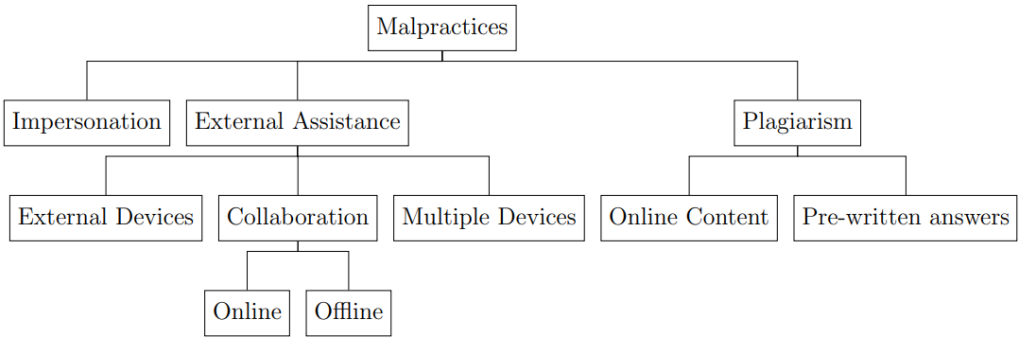



Figure 1: Types of malpractices in online examination

final category is plagiarism, which includes copying content from the internet or some pre-written notes.

## 1.3 Prior Art on Detection Strategies

Various researchers have attempted to detect each of these misconducts through different strategies.

### 1.3.1 Impersonation

Impersonation is an infamous practice that involves having someone else take the exam on behalf of the actual student. This is achieved by using a proxy or hiring someone to take the exam using the student's login credentials [4]. The most common strategy to detect impersonation is biometric authentication, such as fingerprint [5] or face verification [6].
Ullah et al. [7] compared both face and fingerprint authentications in online examination and reported that the main issue using the fingerprint method is the requirement of exclusive fingerprint-scanning devices. Moreover, multiple fingerprint verification intermittently could disrupt the examination flow. Hence, the exam can be manipulated by impersonation once the fingerprint is given by the actual candidate. This issue of fingerprint biometrics makes face authentication systems superior because of the non-requirement of such exclusive hardware as well as intermittent multiple verifications [8] throughout the exam duration.

### 1.3.2 Plagiarism

Plagiarism indicates direct copying of content from different sources during the exam [9]. This may include content from available websites, e-books, and e-notes, as well as from pre-written answers. The most common technique to detect plagiarism is to check for similarities by comparing students' answers against a large database of academic materials, including websites, journals, and previously submitted papers. Such anti-plagiarism software is integrated with the online exam platform. Examinator is one such tool proposed by Approv et al. [10], compares pairs of exam responses and generates a report with evidence for these cases using a similarity metric. Edward et al. [11] have introduced a weirdness vector, which measures how unusual a student's answers are compared to all other students. Most of the detections are based on offline analysis, which opens up a scope for the development of a real-time plagiarism detection algorithm.

### 1.3.3 External Assistance

Candidates use external help either from external devices, collaborating with other examinees, or logging in using multiple devices. Unauthorized devices such as smartphones, smartwatches, or hidden earpieces are employed to access information during the exam [12]. Collaboration with other candidates or members present in the room and offline media can be detected primarily through head pose, talking, or having other people in the room. The state-of-the-art method for malpractice detection is by Indi et al. [13], which proposed a detection scheme using the visual focus of attention obtained through head pose and eye gaze from successive frames. Their method looks for the presence of a face using the Haar Cascade classifier. The system computes eye gaze estimates and combines them with head pose angles to serve as input features for the XGBoost Algorithm.
Sharing screens with others to display exam questions or seek help constitutes another form of malpractice [12]. One way to detect the usage of external devices in online exams is to track the students' head pose and eye gaze and detect any suspicious behavior, such as looking away from the screen. The detection of malpractice in this form is challenging, and several issues can be addressed through substantial research.

## 1.4 Research Issues

The following challenges pop up while developing such a robust online examination proctoring system.

- No data comparing malpractice and authentic cases.
- No literature that considers temporal variations of facial features during the exam.
- Distinguishing live faces and spoofed faces for detecting impersonation during the exam is difficult.

## 1.5 Objectives

The main objective of this work is to develop a robust and comprehensive online examination proctoring system addressing most of these aforementioned conditions, preferably implemented as software.

## 1.6 Contributions

The significant contributions of our work are as follows:

- Creation of a dataset involving malpractice and sincere cases,
- Developing a neural network that considers the temporal variations in head pose, eye gaze, and mouth states,
- Improving impersonation by incorporating face liveliness detection,
- Integrating plagiarism, multiple device detection, screen sharing, and the proposed methods into a single framework.

# 2 Dataset Preparation

Three forms of data are required for this research. The first kind demands live and spoof images for face spoofing-based impersonation detection. The second requirement is different types of head poses, eye gazes, and mouth states. The final category requires temporal variations of these states for two categories, *viz.* authentic and malpractice cases. Accordingly, we have described our database preparation techniques.

## 2.1 Face Liveliness Data

The initial phase of our proposed methodology involves authenticating the examinee, where face verification is an authentication technique. We leverage two established datasets *viz.* the ROSE-Youtu Face Liveness Detection Dataset and the CelebA Spoof For Face AntiSpoofing dataset [14] to accomplish this.

The ROSE-Youtu Face Liveness Detection Dataset comprises 4225 videos featuring 25 subjects with 3350 videos from 20 subjects and has been collected using five mobile phones at resolutions of $640 \times 480$ and $1280 \times 720$, with a typical distance of 30-50 cm between the subject's face and the camera. The dataset includes three distinct types of spoofing attacks *viz.* printed paper attack, video replay attack, and masking attack, thus encompassing a broad range of potential threats to facial authentication systems.

The CelebA Spoof For Face AntiSpoofing dataset contains over 200,000 celebrity images to discern between genuine and counterfeit facial images, with counterfeit images generated using Generative AI, mimicking the challenges posed by sophisticated spoofing attempts.

We construct a sample dataset comprising 50 images per subject from each of these datasets, encompassing both real and fake facial representations across 30 subjects.

## 2.2 Face Data Collection

Several factors were considered when conducting the experiment to record face images, ensuring consistent sitting conditions with comfortable seating arrangements, maintaining adequate lighting conditions, selecting a camera with sufficient resolution and quality positioned at an appropriate angle and height, and determining the optimal distance between the camera and participants. A balanced representation of genders among participants was used to avoid bias, defining age groups reflective of the target population with clear participant instructions and informed consent. These captured images are cropped using the SSD ResNet face detector, as specified in Algorithm 1. The experiment summary is provided in Table 1.

Table 1: Experiment Details

| **Factor** | **Details** |
|---|---|
| Sitting Conditions | Comfortable seating arrangements, with no occlusion |
| Lighting | 500 to 1000 lumens |
| Camera Setup | webcam with 1920 × 1080 positioned at eye level at a distance of 30-50 cm |
| Gender Distribution | 18 males, 12 females |
| Age Group | 18-34 years |
| Recording Protocol | nine head poses, five eye gaze, closed eyes, mouth closed, slightly open and then wide open |

**Algorithm 1** SSD ResNet Face Detection

**Require:** Input image $I$
**Ensure:** Detected faces $D$
1: Preprocess $I$ (e.g., resize, normalize)
2: Pass $I$ through the ResNet backbone to obtain feature maps $F$
3: For each position on feature maps, apply convolutional kernels to predict bounding boxes, class scores, and offsets:
4: $B(x, y, i) = f_{\text{box}}(F(x, y))$
5: $C(x, y, k) = f_{\text{class}}(F(x, y))$
6: $O(x, y, k) = f_{\text{offset}}(F(x, y))$
7: Generate default anchor boxes $A$
8: Compute predicted bounding boxes $P_B$:
9: $P_B = \text{decode}(B, A, O)$
10: Apply non-maximum suppression to remove overlapping boxes
11: Threshold confidence scores $C$ to select face candidates
12: $D = \{B_i : C_i > \text{threshold}\}$

## 2.3 Head Pose Extraction

Nine head poses are considered for malpractice detection, *viz.* frontal (HP0), left (HP1), right (HP2), up (HP3), down (HP4), up left diagonal (HP5), up right diagonal (HP6), down left diagonal (HP7), and downright diagonal (HP8) directions. For each head pose, we have around 300-400 face images. The link for accessing this dataset is available here.

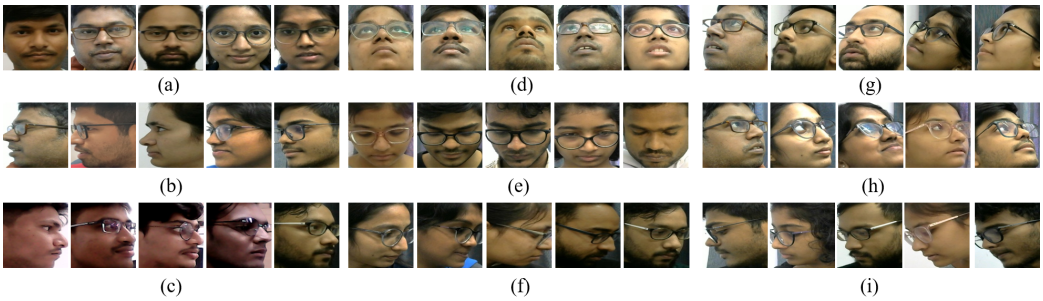



Figure 2: Sample face images of different subjects with the head poses of (a) frontal (b) side left (c) side right (d) up (e) down (f) diagonal down right (g) diagonal up left (h) diagonal up right (I) diagonal down left head poses

## 2.4 Eye Gaze

We extracted the eye images using the Haar cascade classifier and divided them into six eye gaze classes *viz.* frontal, left, right, up, and down, along with a category for closed eyes where gaze information is unavailable. We have 500-700 eye images for each class, with some samples shown in Fig. 3.

## 2.5 Mouth States

We used a pre-trained Haar cascade for mouth detection and refined it manually to enhance its accuracy in identifying mouth regions within images. We have considered three distinct mouth states, *viz.* lips closed, mouth slightly opened, and mouth wide opened, with samples shown in Fig. 4.

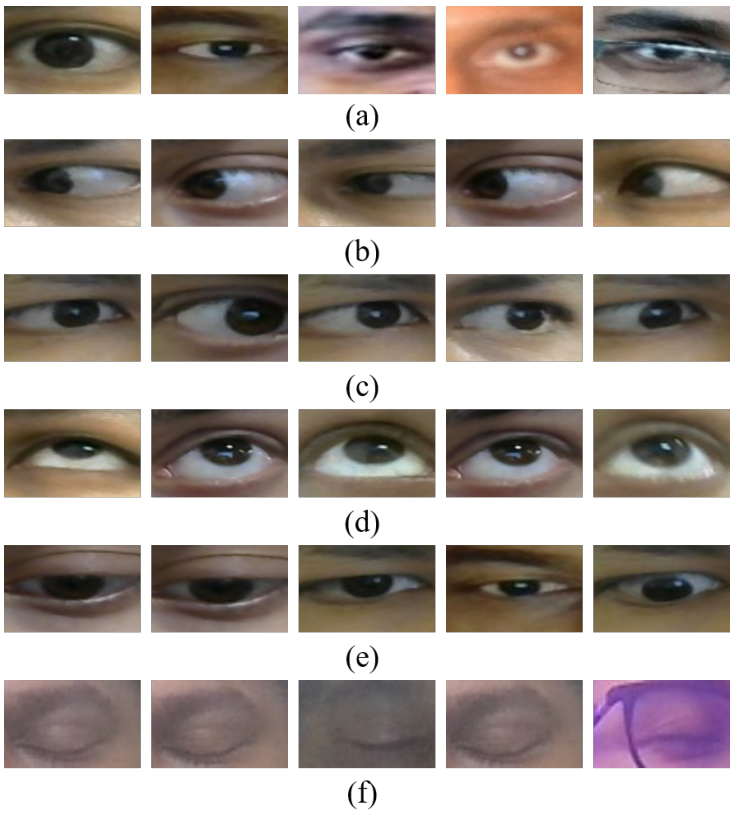



Figure 3: Sample eye images of subjects with (a) frontal gaze (b) left gaze (c) right gaze (d) up gaze (e) down gaze (f) closed eyes

## 2.6 Malpractice Dataset

This dataset captures video recordings depicting both conducting and not conducting malpractice scenarios during online tests. In the Control Group, participants undertake online tests without engaging in any malpractice. Measures are implemented to maintain test integrity, including the absence of electronic gadgets near the testing area, the presence of invigilators in the room, and disabled screen-sharing options. Conversely, the Experimental Group comprises participants who engage in malpractice during online tests. For Group B, facilitative conditions for malpractice are established, including having answer sheets beside the table, the presence of a friend to communicate answers, placement of cell phones on the exam table, and enabling screen sharing options while lacking invigilators' oversight. The online

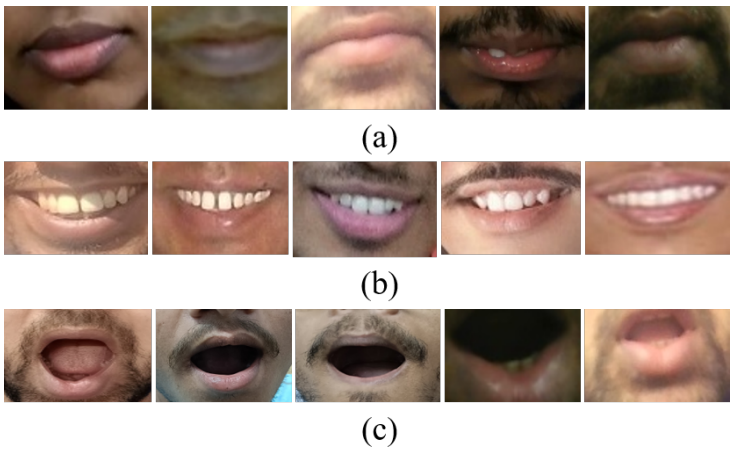



Figure 4: Sample mouth images of subjects with (a) closed lips (b) mouth slightly open (c) mouth wide open

tests encompass questions of varying difficulty levels, comprising two types *viz.* four multiple-choice questions and two short notes, each with a duration of two minutes. The experiment maintains strict participant confidentiality and informed consent adherence, ensuring alignment with ethical guidelines and regulations governing research practices.

# 3 Impersonation Detection

After preparing the data, the first task is impersonation detection through face verification. For this, we have employed embeddings of a Siamese network called the FaceNet. FaceNet [15] is a pre-trained face embedding network useful for face recognition tasks. We employ the embeddings in our framework, as explained.

## 3.1 Face Verification

The students will register five frontal face images $I_{g,k}(x, y), k \in [0, 4]$ during the registration phase. These images are termed as the gallery images. Deploying the FaceNet model $\mathcal{F}$, we generated 128-dimensional face embeddings for the gallery images $F_k$ as shown in Fig. 5. The FaceNet generates

the face embeddings using the Triplet loss $\mathcal{L}$, defined as

$$\mathcal{L} = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0\right) \tag{1}$$

Here, $A$ represents the anchor image, $P$ represents the positive (same person)

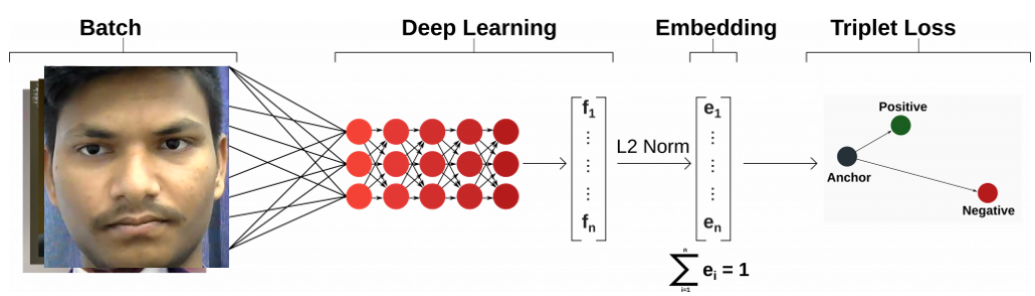



Figure 5: Flow of FaceNet

image, $N$ represents the negative (different person) image, $f(.)$ denotes the embedding function, and $\alpha$ is the margin that is enforced between the positive and negative pairs. During the training process, triplet loss is employed to evaluate the relationships between anchor, positive, and negative images. Following training, embedding vectors are generated to quantify face similarity within a multi-dimensional space. This process involves the utilization of inception modules, normalization techniques, fully connected layers, and L2 normalization before producing the final output.
Mathematically, let's denote the input image as $I(x, y)$, and the output embedding vector as $f(I(x, y))$. The FaceNet model can be represented as a function $f : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{128}$ Here, $H$ is the height, $W$ is the width, and $C = 3$ is the number of channels of the input image.

$$f(I(x, y)) = \mathcal{F}(I(x, y)) \tag{2}$$

The generated embeddings of the test face image $I_t(x, y)$ are now compared using two different metrics. The first approach is finding the Euclidian distance between two image embeddings in 128-dimensional vector space. A threshold is obtained empirically to verify the student.

$$d(I_t(x, y), I_g(x, y)) = ||f(I_t(x, y)) - f(I_g(x, y)||^2 \tag{3}$$

The second approach is similarity computation using cosine similarity $\mathcal{C}$, which can be defined as

$$\mathcal{C}(I_t(x,y), I_g(x,y)) = \frac{I_t(x,y) \cdot I_g(x,y)}{\|I_t(x,y)\| \|I_g(x,y)\|} \tag{4}$$

The mean of these Euclidean distance $d_k(I_t(x,y), I_g(x,y))$ and $\mathcal{C}_k(I_t(x,y), I_g(x,y))$, $\forall k \in \mathcal{G}$. Here, $\mathcal{G}$ is the set of all five gallery images collected during the registration phase through a GUI implemented by us, as shown in Fig. 8. In this manner, we utilized FaceNet to authenticate the identity of the examinee by comparing the intermittent face images with the gallery ones obtained during the registration process. The intermittent test images are obtained after each minute interval.

One major challenge in this process is if the impersonated candidate gives the exam while keeping a photo print of the original candidate in the camera's field of view. This issue is addressed by the detection of face spoofing, as mentioned in the following section.

## 3.2 Face Spoofing

Face spoofing is detected using face liveness detection techniques that involve prompting users to perform specific actions like blinking or smiling to verify their liveness. Spoofed images or videos fail to accurately respond to these prompts, aiding in the detection of spoofing attempts. Face spoofing detection is the process of classifying images into two categories, i.e., Real and Spoof. Hence, this problem is taken up as a binary classification problem. For this work, we consider images of size $224 \times 224 \times 3$. The spoof images belong to screenshots of the faces from electronic devices, face cutouts, and face printouts. First, the face ROI is extracted from these images using a pre-trained model, as explained earlier. This ROI is then fed to our network, which comprises 3 convolutional blocks, each block comprising two conv2D, one MaxPooling2D, and a Dropout layer, along with Normalizations in every layer. This is followed by two dense layers and finally to a sigmoid unit to output a probability of an image being real. This method is specified in Algorithm 2.

**Algorithm 2** Face Liveness Detection using Our Model

**Require:** Captured image $I$
**Ensure:** Liveness label $L$ (either *live* or *spoof*)
Detect faces in image $I$ using a face detection algorithm
**if** No face detected **then**
  $L \leftarrow$ *spoof*
**else**
  Extract ROI from the image using pre-trained caffe model
  Preprocess the extracted face
  Feed the processed image to the model
  **if** Prediction indicates live face **then**
    $L \leftarrow$ *live*
  **else**
    $L \leftarrow$ *spoof*
  **end if**
**end if**
**return** $L$

# 4 Collaboration Detection

Once the candidate passes the impersonation test, the next task is to monitor through the exam process, for potential collaboration. As evident from the literature, during collaboration in online exams, there will be frequent and prolonged shifts in head pose and eye gaze. Talking during exams can also be observed through lip movements. Hence, our proposed method involves observing the temporal variations of these parameters to classify the two cases.

## 4.1 Head Pose Classification

Classifiers were trained on the prepared dataset using transfer learning of pre-trained networks trained on the 'Imagenet' dataset. We compared with a standard CNN based on AlexNet, VGG-16, DenseNet121, ResNet51, and GoogleNet Inception V3 to select the best-performing model for head pose classification. The feature extraction module is fixed, while the classification blocks are trained on our dataset with nine classes, as explained in Table 3.

**Algorithm 3** Transfer Learning on Networks Trained with ImageNet

**Input:** Pre-trained ImageNet model $M_{\text{ImageNet}}$, New dataset $D_{\text{new}}$ with $C$ output classes

**Output:** Fine-tuned model $M_{\text{fine-tuned}}$ on $D_{\text{new}}$

Initialize model $M_{\text{fine-tuned}}$ by loading weights from $M_{\text{ImageNet}}$ Freeze all layers of $M_{\text{fine-tuned}}$ Replace the final fully connected layer of $M_{\text{fine-tuned}}$ with a new fully connected layer with $C$ output neurons **while** *not converged* **do**

    Sample a mini-batch of data $(X, Y)$ from $D_{\text{new}}$ Compute the gradients of the loss concerning the parameters of $M_{\text{fine-tuned}}$ using $(X, Y)$ Update the parameters of $M_{\text{fine-tuned}}$ using Adam optimization

**end**

## 4.2 Eye Gaze Classification

The eye region is localized using the Haar cascade classifier, as explained in Section 2.4. Two eye images are obtained per frame as the left and right eyes. These eye images are trained via transfer learning in the same manner given in Algorithm 3, but with 6 classes for the fully-connected layer.

Table 2: Face Verification Performance Comparison

| **Algorithm** | **Accuracy** | **F1 Score** |
|---|---|---|
| KNN | 95.62% | 0.95 |
| Random Forest | 95.04% | 0.93 |
| Logistic Regression | 94.7% | 0.87 |

## 4.3 Mouth State Classification

Following the successful localization of the mouth region using the appropriate Haar cascade classifier, we proceeded to classify its state into three categories. Similar to our approach in head pose and eye gaze classifications, we utilized transfer learning, but this time with three distinct classes.

## 4.4 Temporal Fusion

We use the variations in head pose, eye gazes, and mouth states across a sequence of frames as indicators of potential malpractice. This work uses

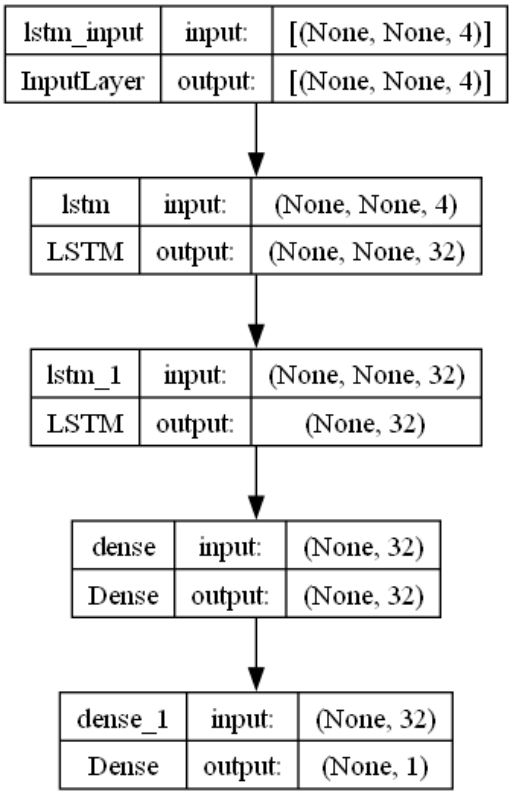



Figure 6: Architecture of the LSTM for temporal classification

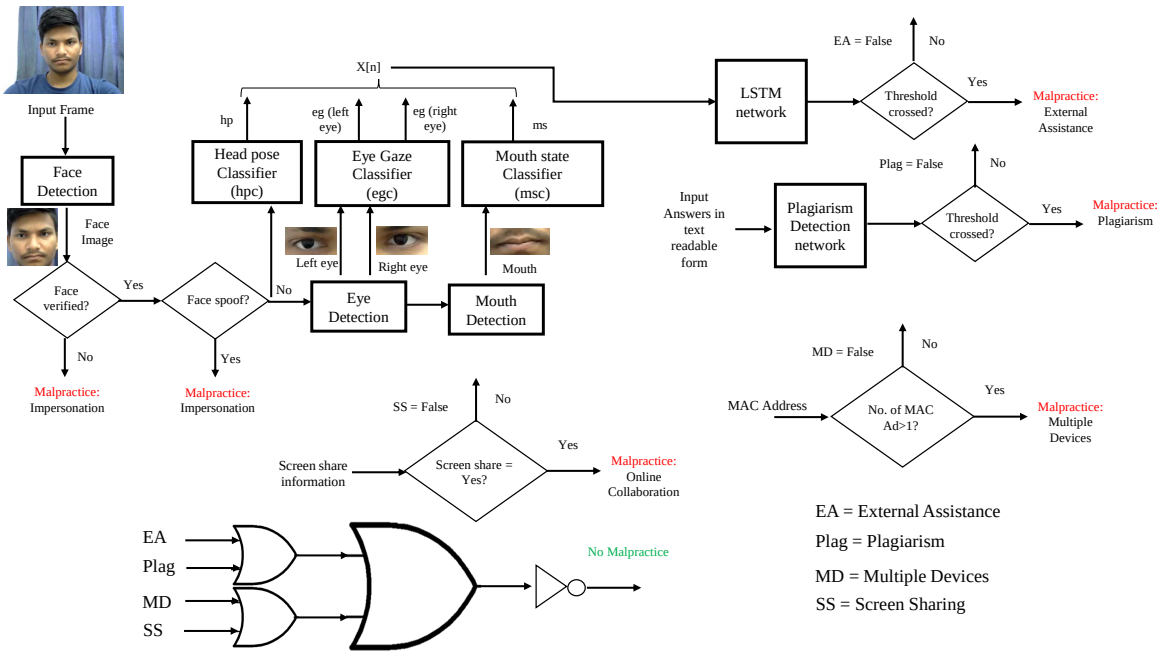
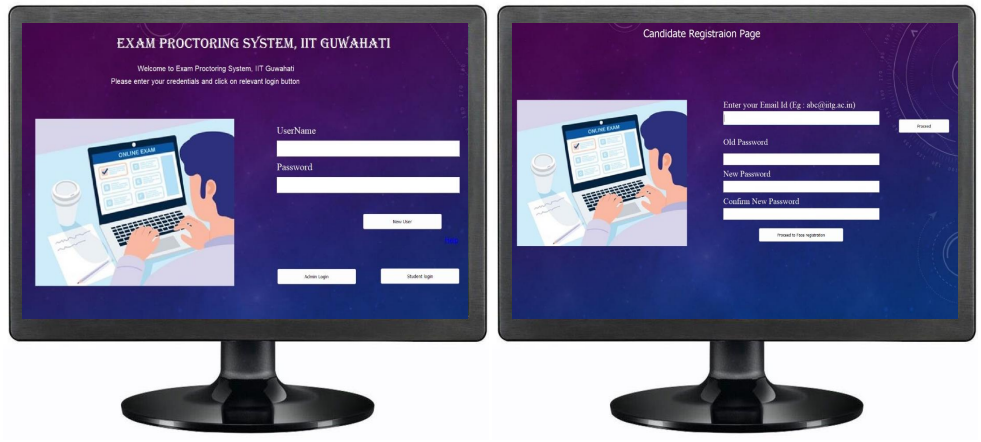



Figure 7: Flowchart representing the entire process

LSTM networks to discern whether a sequence of head poses, eye gazes, and mouth states signifies malpractice. This selection of LSTM networks is due to their effectiveness in capturing short-term memory information, as in malpractices, there are short-term instances of eye gaze shifts or head pose shifts. Also, during talking during the exam, there are short-term lip movements, thereby altering mouth states. The architecture of the LSTM network used for the task is depicted in Fig. 6.

# 5 Plagiarism Detection

Plagiarism's meaning comes from the Latin word 'plagiarius,' which means to kidnap. When someone uses the work of another writer or artist without properly citing the source or giving credit, that's plagiarism. Plagiarism detection is the process of finding plagiarism between two works using a manual or automated process. This section aims at working on one such machine learning based approach to detection plagiarism between two texts.




(a) (b)

Figure 8: Developed GUI using PyQt5 showing (a) user log-in and (b) registration pages

| | Head Pose | | | Eye State | | | Mouth State | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Accuracy (%)** | **F1-score (%)** | **frame per sec(ms)** | **Accuracy (%)** | **F1-score (%)** | **frame per sec(ms)** | **Accuracy (%)** | **F1-score (%)** | **frame per sec(ms)** |
| **CNN** | 98.01 | 98.00 | 164.35 | 96.96 | 96.98 | 89.83 | 92.33 | 92.35 | 85.20 |
| **VGG-16** | 97.31 | 97.28 | 110.60 | 86.98 | 86.87 | 55.54 | 91.06 | 90.95 | 134.93 |
| **DenseNet121** | 97.59 | 97.57 | 112.91 | 98.02 | 98.02 | 115.81 | 94.90 | 94.19 | 126.56 |
| **ResNet51** | 95.96 | 95.94 | 85.64 | 89.23 | 90.45 | 126.82 | 70.62 | 70.53 | 130.45 |
| **GoogleNet** | 98.30 | 98.28 | 128.94 | 96.38 | 97.30 | 189.09 | 95.01 | 95.02 | 158.52 |

Figure 9: Comparision of different headpose classification models concerning accuracy and runtime

Out dataset is taken from QQP (Quora Question Pairs) Dataset. It comprises of two columns containing two different questions, which acts as two input texts for our model, and a label which tells if the two questions are similar or not, that acts as the label for our dataset.

## 5.1 Model Architecture

For our plagiarism detection model, we take two text strings as input. For each text string, we tokenize the text into words and embeddings for each of the words are generated which are then combined to give the embedding for that text. While generating their embeddings, we check for unwanted stop

| Actual Class \ Predicted Class | HP0 | HP1 | HP2 | HP3 | HP4 | HP5 | HP6 | HP7 | HP8 |
|---|---|---|---|---|---|---|---|---|---|
| **HP0** | 0.88 | 0.00 | 0.00 | 0.00 | 0.05 | 0.00 | 0.07 | 0.00 | 0.00 |
| **HP1** | 0.00 | 0.96 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.03 | 0.00 |
| **HP2** | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **HP3** | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **HP4** | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| **HP5** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | 0.00 |
| **HP6** | 0.01 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.00 | 0.00 |
| **HP7** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 |
| **HP8** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

Figure 10: Confusion Matrix for Head Pose Classification (nine classes)

| Actual Class \ Predicted Class | EG0 | EG1 | EG2 | EG3 | EG4 | EG5 |
|---|---|---|---|---|---|---|
| **EG0** | 0.97 | 0.02 | 0.00 | 0.00 | 0.00 | 0.00 |
| **EG1** | 0.00 | 0.98 | 0.00 | 0.00 | 0.00 | 0.00 |
| **EG2** | 0.01 | 0.01 | 0.95 | 0.00 | 0.02 | 0.01 |
| **EG3** | 0.00 | 0.00 | 0.00 | 0.99 | 0.01 | 0.00 |
| **EG4** | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | 0.01 |
| **EG5** | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |

Figure 11: Confusion Matrix for Eye Gaze Classification (six classes)

Table 3: Comparison of Face Verification Results with and without Liveness Detection

| **Verification Results** | | **Liveness Detection** |
|---|---|---|
| **With Liveness Model** | **Without Liveness Model** | **Accuracy (%)** |
| TP = 470 | TP = 473 | 98% |
| FP = 6 | FP = 27 | |
| FN = 2 | FN = 11 | |
| TN = 46 | TN = 489 | |

Table 4: Comparison of Deep Learning Models for Face Liveness Detection

| **Method** | **Metrics** | | | |
|---|---|---|---|---|
| | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| VGG16 | 0.92 | 0.92 | 0.92 | 0.92 |
| VGG19 | 0.89 | 0.89 | 0.89 | 0.89 |
| ResNet150 | 0.79 | 0.80 | 0.79 | 0.78 |
| DenseNet121 | 0.95 | 0.95 | 0.95 | 0.95 |
| InceptionV3 | 0.91 | 0.91 | 0.91 | 0.91 |
| Our Model | 0.96 | 0.96 | 0.96 | 0.96 |

words and omit them while finding the word embeddings. To generate word embeddings, we used a pre-trained word2vec model from google. We then split the training, test, and validation dataset into dictionaries with keys "left" and "right" corresponding to the two text strings in each pair. The maximum sequence length was calculated using the training and validation dataset, which is used to apply zero-padding in our dataset to ensure uniform sequence lengths. Next, we define and train a Siamese network, which consists of a shared embedding layer followed by a shared LSTM. The previously mentioned left and right inputs are passed separately to the network. Then the Manhattan distance between the two output vectors of the LSTM is calculated to get the similarity between the input sequences. On every epoch, the model calculates the 'binary crossentropy' loss between the true and predicted output and tries to minimize it to train the network. The algorithm is specified in Algorithm 4.

Table 5: Comparison of Deep Learning Models for Plagiarism Detection

| **Final Layer** | **Metrics** | | | |
|---|---|---|---|---|
| | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
| Exponent Negative Manhattan Distance | 0.83 | 0.83 | 0.83 | 0.83 |
| Cosine Similarity | 0.80 | 0.82 | 0.80 | 0.80 |
| Sigmoid Layer | 0.78 | 0.79 | 0.78 | 0.79 |

Table 6: Confusion Matrix for Mouth state classification

| | | **Predicted Class** | | |
|---|---|---|---|---|
| | | **Closed** | **Slight Open** | **Wide Open** |
| **Actual Class** | **Closed** | 0.95 | 0.02 | 0.02 |
| | **Slight Open** | 0.02 | 0.97 | 0.01 |
| | **Wide Open** | 0.04 | 0.01 | 0.95 |

# 6 Framework Integration

The overall proposed framework is shown in Fig. 7. The framework will take the images from a camera along with other information *viz.* the MAC address of the logged-in device, screen sharing status, and input answers.

## 6.1 Graphical User Interface

We have developed a basic Graphical User Interface (GUI) tailored for seamless user interaction during both login and registration processes, facilitating the smooth enrollment of candidates into the portal, as depicted in Fig. 8. Following data collection, stringent security measures are implemented to ensure the confidentiality and privacy of stored information within our database.

The login interface features fields for user identification (userID) and password entry. Upon submission, the software prompts candidates to capture their facial images for further verification. Additionally, functionalities for new user registration and password retrieval are conveniently provided.

Users are prompted to input their email address within the registration GUI, from which pertinent details are retrieved from the server.

Table 7: Comparison of RNN Variants in Face Verification

| **Model** | **Accuracy** | **Precision** | **Recall** | **F1 Score** |
|---|---|---|---|---|
| RNN | 57.00 | 0.60 | 0.57 | 0.51 |
| LSTM | 90.11 | 0.91 | 0.9 | 0.9 |

Table 8: Confusion Matrix

| | | **Predicted** | |
|---|---|---|---|
| | | Malpractice | Sincere |
| **Actual** | Malpractice | 0.98 | 0.02 |
| | Sincere | 0.18 | 0.82 |

## 6.2 Impersonation Framework

The initial phase involves face detection within the captured frame. Given that each login attempt should involve only one candidate, the face detection algorithm is expected to yield a single detected face. Consequently, the face detection process includes a face count mechanism and flags any instance of multiple faces detected, ensuring compliance with the single candidate per login requirement. Following successful face detection, the next step is to verify the candidate's identity. The system flags the occurrence as a potential impersonation attempt if more than one face is detected. The verification process proceeds for a single detected face, where the facial features are compared against stored data for authentication. Any failure in this verification process results in the flagging of the attempt as impersonation. Upon successful verification of the candidate's identity through facial recognition, the system proceeds to conduct liveliness detection. This involves assessing the authenticity of the detected face to ensure it is not a spoof attempt. Any indication of spoofing leads to the immediate flagging of the attempt as impersonation. Upon successful verification and liveliness detection, the system initiates three parallel threads for concurrent execution. These threads are responsible for face pose estimation, eye gaze estimation, and mouth state estimation, respectively, enhancing the real-time analysis of the candidate's facial expressions and gestures.

**Algorithm 4** Plagiarism Detection between two text strings

**Require:** Two Text Strings
**Ensure:** Plagiarism Label $L$ (either *Is Plagiarised* or *Not Plagiarised*)

```
Find word embedding of both strings using word2vec model
if Both embedding are same then
    L ← Is Plagiarised
else
    Pass them to a Siamese Bidirectional LSTM Model
    Compute the Exponent Negative Manhattan Distance
    if Prediction indicates text is plagiarised then
        L ← Is Plagiarised
    else
        L ← Not Plagiarised
    end if
end if
return L
```

## 6.3 Plagiarism Framework

The plagiarism detection model, as previously outlined, is invoked for verification purposes upon the candidate's submission of answers. Utilizing the established plagiarism detection mechanism, the submitted content undergoes scrutiny, comparing it against existing data sources. Any instance where a significant resemblance of 15% or higher is identified between the submitted answers and pre-existing materials warrants immediate flagging as plagiarism. This proactive approach ensures the integrity of the assessment process and upholds academic standards by discouraging dishonest practices.

## 6.4 External Assistance Framework

This framework is evoked in different stages, depending on the assistantship checked.

### 6.4.1 Online Collaboration

Screen-sharing activities are closely monitored to further enhance the integrity of the assessment process and prevent potential instances of online collaboration. The system incorporates mechanisms to verify the veracity of

screen-sharing activities during the assessment period. If the screen-sharing information indicates collaborative behavior, it is promptly flagged for further investigation. This proactive measure serves to deter unauthorized collaboration and maintain the authenticity of individual candidate submissions.

### 6.4.2 Offline Collaboration

As explained, offline collaboration is checked through the temporal variations in head pose, eye gaze, and mouth states. Each 15s sample is sent to the LSTM for verification, and if any such window is classified as malpractice, it is flagged.

### 6.4.3 Multiple Devices

The MAC ID of the user is obtained, and if a user has multiple MAC IDs, it is flagged as Multiple Devices malpractice.

# 7 Results

This section presents the comprehensive results for impersonation detection, plagiarism check as well as potential external assistance through variations in head pose, eye gaze, and mouth state classification. The performances are compared with respect to the accuracy, precision, recall, F1-score as well as runtime through frames per second (fps).

## 7.1 Impersonation

The impersonation detection has been performed as shown in Table 2. These were tested on real images. We also performed spoof face verification through face liveliness, whose results are tabulated in Table **??**. In Table 4, we find that our model has better performance as compared to the other standard nets.

## 7.2 Plagiarism

Table 5 provides the plagiarism detection comparison, which reveals that our model performs better.

# 8 Conclusion

This work identifies different forms of malpractice occurring in online examinations using a multimodal approach, the modes being facial images, submitted answers in text format, MAC ID, and screen sharing status. Notably, for impersonation detection, we have implemented a feature assessing face liveliness, which scrutinizes whether an imposter is employing a printout or live video to mimic the appearance of the legitimate candidate. A novel contribution of this research lies in the proposal to analyze spatiotemporal variations in head pose, eye gaze, and mouth states to gauge the probability of malpractice with experimental validation. The GUI incorporates these innovative features for malpractice detection, in addition to existing functionalities such as detecting simultaneous logins from multiple devices and screen-sharing activities. The classification of head pose, eye gaze, and mouth states is achieved using cutting-edge deep learning frameworks, with rigorous comparisons conducted to identify the most effective models.
However, it's crucial to acknowledge a significant limitation that falls beyond the scope of this study: the potential for malpractice through the discreet use of small external devices, like smartphones, positioned directly in front of the camera. In such cases, where the individual's gaze remains directed toward the primary screen while utilizing the external device, detecting malpractice becomes challenging.

# 9 Publications

We have submitted our research manuscript titled "A Comprehensive Multimodal Approach for Detecting Malpractice in Online Exams" to IEEE Transactions on Education for consideration. This research paper is currently under review at the leading academic journal.

# References

[1] S. Padmanabhan, "Digital transformation in higher education: Advantages and challenges in 2023," in *The Impact of Digitalization in a Changing Educational Environment*. IGI Global, 2023, pp. 59–69.

[2] K. Nataraj, D. Xavier, T. Vishrutha, S. Kavya, and M. Hemalatha, "Malpractice detection system for online examination using ai," in *International Conference on Data Science, Machine Learning and Applications*. Springer, 2022, pp. 381–389.

[3] D. Kangane, S. Pappu, V. Shah, and N. Shaikh, "Problems & malpractices during online exams with possible solutions," 2021.

[4] K. Sambell, L. McDowell, and S. Brown, ""but is it fair?": an exploratory study of student perceptions of the consequential validity of assessment," *Studies in educational evaluation*, vol. 23, no. 4, pp. 349–371, 1997.

[5] L. Wei, Z. Cong, and Y. Zhiwei, "Fingerprint based identity authentication for online examination system," in *2010 Second International Workshop on Education Technology and Computer Science*, vol. 3, 2010, pp. 307–310.

[6] M. A. Sarrayrih and M. Ilyas, "Challenges of online exam, performances and problems for online university exam," *IJCSI International Journal of Computer Science Issues*, vol. 10, no. 1, pp. 1694–0784, 2013.

[7] A. Ullah, H. Xiao, and M. Lilley, "Profile based student authentication in online examination," in *International Conference on Information Society (i-Society 2012)*, 2012, pp. 109–113.

[8] M. Ghizlane, B. Hicham, and F. H. Reda, "A new model of automatic and continuous online exam monitoring," in *2019 International Conference on Systems of Collaboration Big Data, Internet of Things Security (SysCoBIoTS)*, 2019, pp. 1–5.

[9] Z. Jiang and J. Huang, "Effective and efficient strategies and their technological implementations to reduce plagiarism and collusion in non-proctored online exams," *IEEE Transactions on Learning Technologies*, vol. 15, no. 1, pp. 107–118, 2022.

[10] R. Apoorv, A. Dahiya, U. Sreeram, B. R. Patil, I. Irish, R. Graziano, and T. Starner, "Examinator: A plagiarism detection tool for take-home exams," in *Proceedings of the Seventh ACM Conference on Learning@ Scale*, 2020, pp. 261–264.

[11] E. F. Gehringer, X. Liu, A. Kariya, and G. Wang, "Comparing and combining tests for plagiarism detection in online exams." in *EDM*, 2020.

[12] S. Prathish, K. Bijlani *et al.*, "An intelligent system for online exam monitoring," in *2016 International Conference on Information Science (ICIS)*. IEEE, 2016, pp. 138–143.

[13] C. S. 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 (Online)*, vol. 16, no. 8, p. 47, 2021.

[14] Y. Zhang, Z. Yin, Y. Li, G. Yin, J. Yan, J. Shao, and Z. Liu, "Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. Springer, 2020, pp. 70–85.

[15] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.