A Novel CNN-RNN Model for E-Cheating Detection Based on Video Surveillance

Aqeel Zaffar [a] ✉, Muhammad Jawad [b], and Muzammil Shabbir [c]

[a]DexterCode Software Development Islamabad, Federal

[b]Pakistan Ordinance Factory, Wah Cantt

✉, [a] zaffaraqeel01@gmail.com

ABSTRACT

Nowadays, everything needs to be digitized, and scientific knowledge is constantly bringing comfort and change to everyday life. Autonomous systems have become a prominent technology in recent years, with a variety of applications in different fields. Our proposed system is designed to maintain academic integrity in exams. The system uses computer vision techniques to monitor the behavior of students during the exam and detect any suspicious activities, such as looking at someone else's paper or using unauthorized materials. The proposed E-Cheating Detection consists of four core steps: 1) Student/Person Detection and Tracking, 2) Detect suspicious activities, 3) Generating alerts, and 4) Mark attendance. Student detection from videos is performed by using YOLOv7 and DeepSort tracker is used to track the detected persons that are being detected by the YOLOv7 algorithm. To classify suspicious activities (such as the exchange of paper, and giving codes to one another, etc.) the system uses CNN-RNN architecture in which the inceptionV3 model is used for feature extraction. The system will generate real-time alerts for suspicious behavior by sending an email via SMTP to the exam administration /invigilator. The system marks the student's attendance by recognizing and matching student faces that are stored in the database. The performance of the system will be evaluated by conducting a series of experiments using simulated scenarios, and the results will demonstrate the effectiveness of the proposed system in detecting suspicious activities during physical exams. The proposed system has the potential to promote exam integrity and create a fair environment for all students, ultimately improving the education system's quality.

Keywords: *Detect Suspicious Activities; CNN-RNN, YOLOv7; DeepSort; Face Recognition; Physical Exam; Smart Attendance*

1. INTRODUCTION

In general, the term cheating refers to deceit, fraud, or dishonesty with someone. In online gaming, particularly competitive ones like First-Person Shooter (FPS) games, cheating prevails [1]. But in an academic context, the term cheating means the use of such materials that are prohibited during exams, copying assessments, and quizzes of another student. For example, copying data from other students during the exam or in assignments, and communicating with each other during the exam [2]. Cheating in exams and academic dishonesty has become a serious issue and kills students' creativity. Exams are part of the student's life. Two methods are frequently used for exams: Physical Mode and Online Mode [3]. Despite the level of development, cheating on exams has become a global phenomenon regardless of the method used by institutions. There are so many reasons students cheat in exams but a few of them are the most common; being fear of failure, helping their friends, and the "Pressure to

Succeed” phenomenon [4]. Education has become more concerned about cheating during written examinations because the students who cheat in exams undeservingly get higher scores than the deserving ones. Cheating often occurs due to different factors like the absence of an examiner or the negligence of the administration. So constant supervision by the proposed system during exams will lead to a cheating-free environment [5].

Full-time supervision during exams is difficult for the invigilator/examiner because full-time supervision requires time and energy. Studies have been made over the past years on participants' abnormal activities in exams and how these abnormal activities could be combated by educational institutes. A recent study published in the US shows that 95% of secondary school students admitted cheating had not been caught and 51% of secondary school students believe that cheating was not wrong [6]. According to a survey conducted by the International Center for Academic Integrity (ICAI), more than 60% of university students admit to cheating in some form. The same survey also revealed that 64% of high school students admitted to cheating on a test, 58% admitted to plagiarism, and 95% said they participated in some form of cheating, whether it was on a test, plagiarism, or copying homework [7].

The consequences of academic dishonesty can be far-reaching. Depending on the severity of the offense, the repercussions can range from a warning for a first offense to a failing grade in a course to expulsion from the university [8, 9]. Academic dishonesty not only compromises the credibility of academic institutions but also undermines the fundamental principles of fair competition and merit-based success. Instances of cheating lead to a distorted representation of students' actual capabilities, skewing academic evaluation processes. Furthermore, the normalization of dishonest practices cultivates a culture where ethical values take a backseat, nurturing detrimental attitudes toward integrity and fairness [10]. Cheating detection systems during physical exams have been developed to prevent students from cheating during test taking. These systems use various technologies such as biometrics, surveillance cameras, and proctoring software to monitor students and detect any suspicious behavior. Biometric systems use fingerprints, facial recognition, and other physiological characteristics to identify and track students [11].

Surveillance cameras capture footage of the testing environment, which can be used to detect cheating. Proctoring software uses machine learning algorithms to detect suspicious behavior, such as students looking away from the screen or typing on a second device. To provide a cheating-free environment so that everyone will be equally treated and utilize invigilator/examiner time and energy for more productive things [12]. Therefore, this article provides a novel CNN-RNN method that classifies the abnormal activities accurately. Additionally, the integration of Convolutional Neural Networks (CNNs) within cheating detection systems stands as a pivotal advancement in academic dishonesty identification. CNNs, renowned for their efficacy in image and video analysis, play a crucial role in processing visual data obtained from surveillance cameras in exam settings. These networks excel in pattern recognition and feature extraction, enabling the identification of suspicious behaviors or irregular activities among students.

By leveraging CNNs within the proposed system, it becomes more adept at accurately pinpointing anomalies within the visual data, thus enhancing the overall efficacy of academic dishonesty detection.

The core contribution steps are defined as follows:

- The core contributions of the methodology involve systematic integration of YOLOv7 and DeepSort for individual student detection and tracking within video frames. YOLOv7 is specifically fine-tuned to identify and track individuals, focusing solely on the "person" class while disregarding other detectable classes. This optimized approach ensures accurate and precise tracking of students throughout the video sequences.
- Simultaneously, a CNN-RNN sequential model is deployed to analyze the temporal sequence of video frames. This model encompasses distinct layers such as input, GRU (Gated Recurrent Unit), Dropout, Dense, and Softmax. Trained with carefully tuned hyperparameters utilizing 40 training epochs, a batch size of 64, a maximum sequence length of 30, a learning rate set at 0.1, and employing the Adam optimizer the CNN-RNN model specializes in classifying abnormal activities within the tracked video sequences.

The article is structured into five main sections. Section II provides a review of the relevant literature and prior work on the Automated Cheating Detection System. Section III details the proposed method and outlines the specific steps involved. Section IV presents the results of the system and includes a discussion of the findings. Finally, Section V involves the conclusion of the proposed system.

2. RELATED WORK

Academic dishonesty is not the new one but it comes from a decade. Some models have already been implemented to provide a cheating-free environment and maintain academic integrity [13]. Automated cheating detection systems for physical exams are not as common as those for online exams [14], as it might be difficult to monitor students' suspicious activities. Proctoring software such as ProctorU, ExamSoft, Respondus, etc [15] can be an effective way to detect cheating during online exams, but it is not without its drawbacks. Institutions should weigh the benefits and costs of using proctoring software and consider alternative methods to detect cheating, such as open-book exams or take-home exams. Several cheating detection methods have been developed over the years to uphold academic integrity, both in physical and online examination settings. In 2022, a physical method utilizing Viola and Jones's algorithm was introduced, focusing on head, hand, and iris movements to detect abnormal behavior during exams. This method, however, faces challenges in monitoring all suspicious activities effectively. Its dataset consists of labeled recordings of physical movements, differentiating between normal and abnormal behaviors [16]. In 2020, another physical method utilized OpenPose, ALEXNET, OpenCV LBPH, and SMTP Library technologies. This method captures a lateral view and is limited to monitoring one person, potentially limiting its overall effectiveness. The dataset emphasizes diverse poses such as bending back, stretching arms, bending down, and facing the camera, providing ground truth data recorded through video formats [6]. Similarly, in the same year, a different physical method relied on Embedded Technology, RFID, and Remote Server technologies. This method primarily depends on facial detection and is considered costly in its implementation. Its dataset comprises facial images or RFID-collected data from surrogate examinees during examination scenarios [17]. In 2019, a physical method utilizing Feature and AdaBoost faced limitations with an accuracy level below 75%. Its training dataset involved photos of students captured from specific angles, aimed at comparisons among students present in the classroom during exams [18]. Another physical method introduced in 2017 employed 3D CNN, LSTM, and XGBoost Classifier to detect abnormalities between subjects during examinations. Its dataset included distinct classes like NoCheat, LookLeft, LookRight, and others, facilitating the identification of various cheating behaviors [19]. In contrast, a physical method introduced in 2016 utilized statistical measures such as RMS and FFT, designed particularly for invigilators with impairments. However, the text did not specify a particular dataset used for this method [20].

Shifting to online methods, in 2021, a technique utilized a Regression Model with LSTM and KDE-based Outlier Detection for online exam proctoring. However, its accuracy faced doubts due to the absence of gesture and posture detection. The dataset employed for this method consisted of previous exam scores and quiz results [21].

Finally, in 2017, an online method incorporated Wearcam, Webcam, and Microphone Controller technologies for cheating detection. This method was deemed expensive for students, using authentication data comprising student faces and associated verification details [22, 23]. In extending the scope of cheating detection, a notable approach employed a data analysis strategy harnessing Machine Learning (ML) and feature engineering techniques to identify instances of Internet cheating as proposed in the ADMP plan of their previous study [24]. The method primarily entailed log file analysis, encountering a prevalent hurdle in obtaining labeled datasets. Addressing this challenge, the researchers introduced iQuiz3, an online quiz tool specifically designed to collect labeled datasets sourced from laboratory-based online assessments. To bolster the veracity of the acquired data, the study validated ground truth by employing established methodologies, including simulation and self-reports [25].

The systems that have been discussed so far have their limitations some of them were implemented for the online proctoring system and very few of them were implemented to detect suspicious activities in physical mode. The use of Cheating Detection Systems in physical exams can be challenging and may not be as effective as in online exams. It is important to consider the cost, privacy, and effectiveness of the technology before implementing it in physical exams. A combination of different methods and technologies may be more effective than relying on a single method [26].

3. METHODOLOGY

To address the problem and challenges that we discussed earlier, we propose an Automated Cheating Detection System that helps to make our education system more reliable. This system should be able to detect suspicious activity performed by the students. The system that we developed has two primary components one is to detect the cheating activities of students and the other one is to notify the invigilator in case cheating is being caught by the system. The core contribution steps of the proposed methodology are shown in Figure 1.

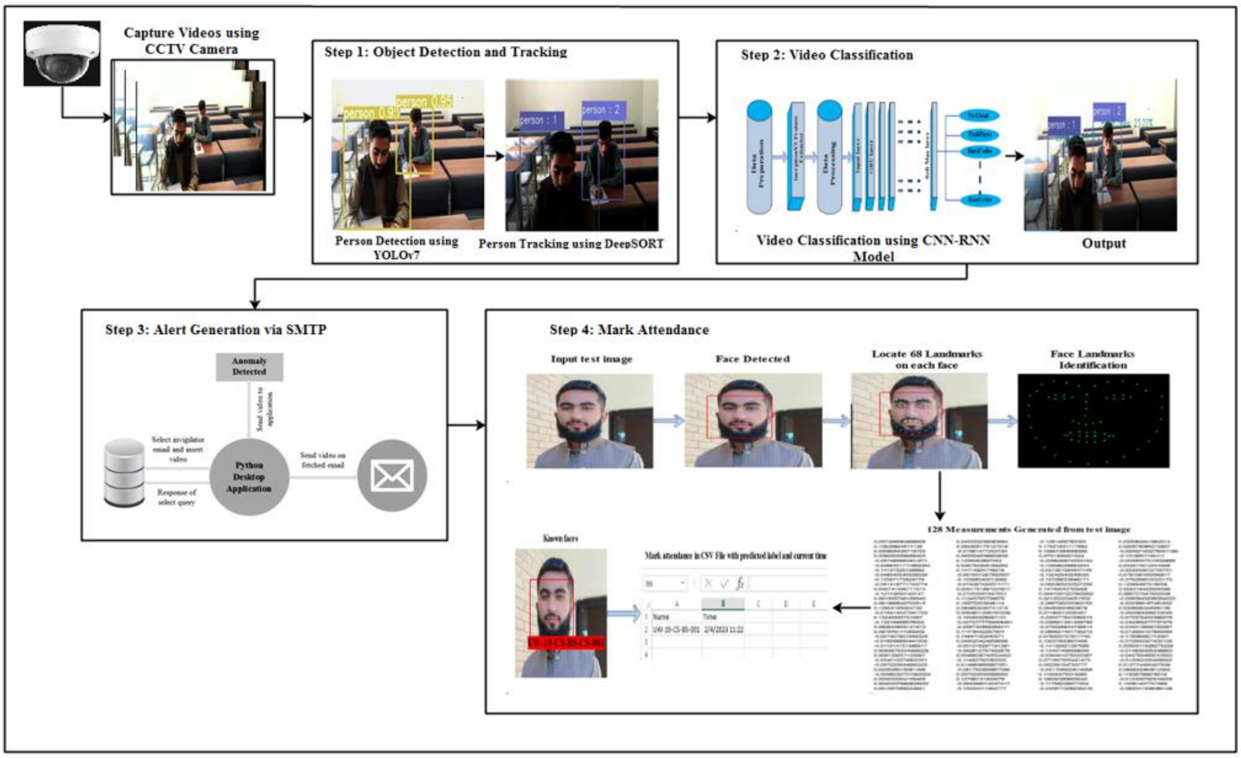

Capture Videos using CCTV Camera
Step 1: Object Detection and Tracking
person 0.9
person 0.95
person : 1
person : 2
Person Detection using YOLOv7
Person Tracking using DeepSORT
Step 2: Video Classification
Data Preparation
Data Processing
Soft Max layer
Video Classification using CNN-RNN Model
Output
Step 3: Alert Generation via SMTP
Anomaly Detected
Send video to application
Select invigilator email and insert video
Response of select query
Python Desktop Application
Send video on fetched email
Step 4: Mark Attendance
Input test image
Face Detected
Locate 68 Landmarks on each face
Face Landmarks Identification
128 Measurements Generated from test image
Known faces
Mark attendance in CSV File with predicted label and current time
Name
Time
UW-19-CS-BS-001
2/4/2023 11:22

Figure 1. Core contribution steps of the proposed methodology

Figure 1 illustrates that these steps collectively form the core functionalities and innovations of the work being reviewed, addressing various aspects of video analysis, object manipulation, classification, and automated alerting for monitoring purposes.

3.1 OBJECT DETECTION

Real-time object detection is an important component in computer vision systems. Different detection algorithms are used for real-time object detection such as Faster R-CNN, SSD, RetinaNet, and Yolo. The proposed Cheating Detection System uses the YOLOv7 which is a single-stage object detection algorithm because it can process video streams at over 100 frames per second (fps) on a GPU. This is significantly faster than other algorithms, such as Faster R-CNN, which can only process around 7-8 fps on a GPU. YOLOv7 uses extended ELAN which is the extension of ELAN architecture. The E-ELAN architecture uses expansion, shuffling, and merging of cardinality techniques to improve the learning ability of the network without affecting the original gradient path [27]. Two methods “extended” and “compound scaling” are used which offer significant improvements in accuracy while keeping inference costs low and high accuracy. The "extended" method elaborates upon the inherent architecture of YOLOv7, extending the ELAN framework by employing techniques like expansion, shuffling, and merging of cardinality. This augmentation significantly enhances the network's learning capability without disrupting the original gradient path. Moreover, the "compound scaling" approach is instrumental in achieving high accuracy without compromising on inference costs. By carefully balancing model scaling factors and optimizing network dimensions, this method remarkably improves accuracy metrics while keeping computation expenses in check. Notably, these methods collectively enable YOLOv7 to outperform other object detection models by reducing parameters by 40% and computation by 50% [28]. The amalgamation of "extended" and "compound scaling" methods within YOLOv7 not only elevates its accuracy but also sets a benchmark for balancing performance metrics, making it a robust choice for real-time object detection applications. YOLOv7 features a more effective feature integration method, a more robust loss function, and improved label assignment and model training efficiency. These advancements make YOLOv7 more efficient and require less expensive computing hardware. This system utilized a pre-trained configuration without further training or adjustment of epochs or optimizers. Test videos were passed through the pre-trained model to generate the results. Thus, no specific epoch counts or optimizer application was involved in this evaluation. YOLOv7 detector detects different kinds of objects and is localized to each object, but here our focus is to detect the person class so by optimizing YOLOv7 we get the desired result with a high confidence score as shown in Figure 2.

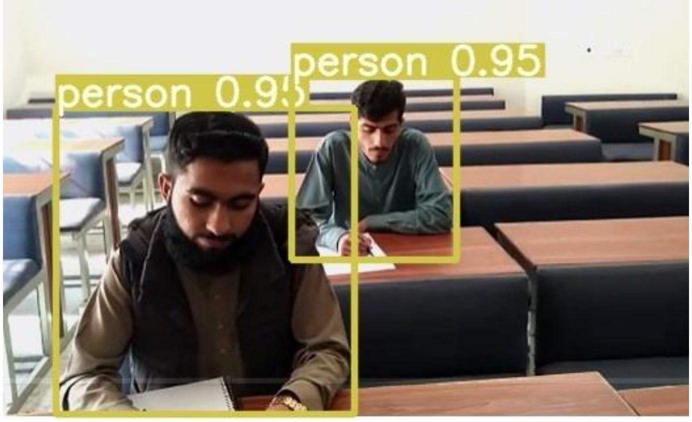


Figure 2. Person detection using Yolov7

3.2 OBJECT TRACKING AND CROPPING

DeepSORT (Deep Learning for Multi-Object Tracking) is a popular algorithm for tracking multiple objects in a video sequence using deep neural networks. Here are the steps involved in using DeepSORT for tracking objects in video series [29]:

- The first step is to detect objects in the video frames using a deep learning-based object detection algorithm, such as YOLO, SSD, or Faster R-CNN. We can detect the person class using YOLOv7 as we discussed in the above section.
- After detecting a person from the video frames, the subsequent stage is to obtain characteristics from them utilizing a deep neural network like a CNN. This enables us to correlate the persons across various frames. To achieve this, we use a CNN trained on a large-scale person-identification dataset. This dataset includes over 1,100,000 images of 1,216 pedestrians, enhancing the system's ability to handle missed detections and obstructions
- The extracted features are then used to associate the persons across frames using a data association metric of the Kalman filter algorithm that uses an eight-dimensional state-space that contains aspect ratio, height, bounding box center position, and their respective velocities. It assigns a unique ID to each person and tracks its position, velocity, and other properties over time.
- The association between newly arrived measurements and Kalman states creates an assignment problem that can be solved using the Hungarian algorithm. To enhance the problem formulation, we integrate both motion and appearance information by utilizing two relevant metrics. To take motion information into account, we calculate the Mean squared distance between the anticipated Kalman states and the recently obtained measurements.

The assigning IDs of persons using DeepSORT can be shown in Figure 3 given below.

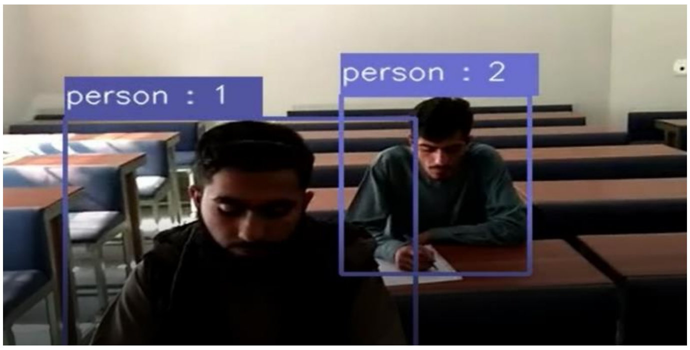


Figure 3. Person tracking using DeepSort

Once the person is being tracked we crop each person according to the coordinates of the bounding box and then pass 32 frames of each tracked person sequentially to the video classification model CNN-RNN for classification. The process of cropping is done using the Python library OpenCV.

3.3 VIDEO CLASSIFICATION

To classify whether the students perform a suspicious activity during exams or not we use the Keras sequential model with CNN-RNN architecture. Keras Sequential model is a high-level neural network that allows developers to easily define and train neural network models in Python. The Sequential model is a linear stack of layers, where each layer is connected to the next sequentially. CNN-RNN architecture is a combination of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) architectures. CNN is a type of neural network used for image classification and processing, while RNN is used for sequence modeling and processing. The combination of these two architectures enables the model to effectively process both spatial and temporal information, making it suitable for tasks such as video and audio classification [30]. To train this model for video classification we use the dataset available at [19]. This dataset consists of 8 different classes which include, NoCheat, LookLeft, LookRight, ExchPaper, FaceCodes, HandCodes, PocketSheet, and PantsSheet. Each class contains videos. A detailed description of this dataset is shown in Table 2.

Table 1. Description of the Benchmark dataset

Groups	**No. of videos in each group**	**No. of frames in each group**
NoCheat	1460	46720
LookLeft	1791	57312
LookRight	1635	52320
PocketSheet	884	28288
PantsSheet	984	31488
ExchPaper	582	18624
FaceCodes	864	27648
HandCodes	856	27392
Total	**9056**	**289792**

The total dataset consists of 9056 videos, collectively containing 289,792 frames, offering a comprehensive range of scenarios and behaviors crucial for training and validating the cheating instances. Once the videos have been loaded their labels and paths are stored in a CSV file for further processing. Finally, the dataset has been divided into train and test dataset which contains 80% of videos in the training dataset and the test dataset contains 20% of videos. These videos are then converted into frames and stored these frames into a Numpy array. These frames are then passed to Keras pre-trained feature extractor model InceptionV3 which is trained on the ImageNet-1k dataset. The labels of the videos are in string format while the neural network does not understand the string values so we need to encode these string format values into an integer using the StringLookup layer. Finally, we feed the processed data to a sequence model that contains different layers like the Input layer, Dropout, Dense, and, GRU layer. The dropout layer with a dropout rate of 0.4 applies to the output of the input layer. During training, the Dropout layer randomly sets 40% of the units in the input tensor to 0 to prevent overfitting. By stacking multiple GRU layers, the network can learn hierarchical representations of the sequential input data, with each layer capturing different levels of abstraction. The output of the final GRU layer can be fed into a dense layer with a softmax activation function to predict the class probabilities for a video sequence. The sequence model uses 99,936 trainable parameters. The Keras sequential model layers with CNN-RNN architecture are shown in Figure 4.

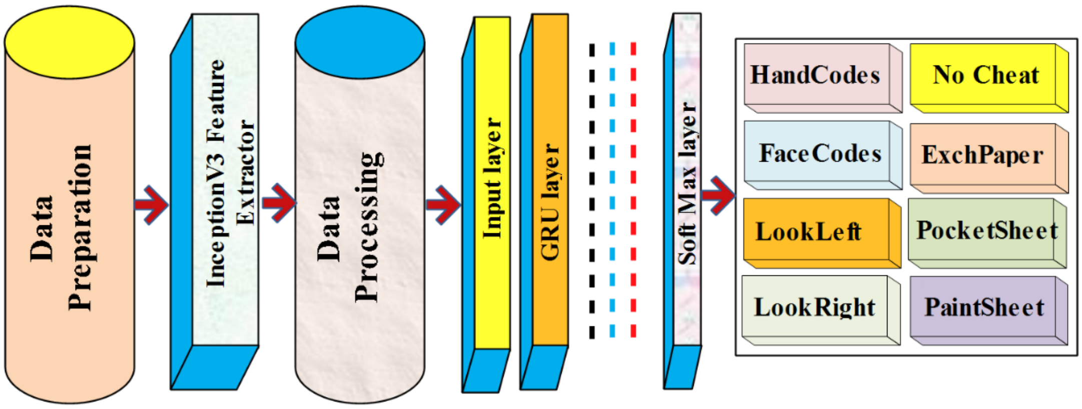


Figure 4. CNN-RNN Sequential model architecture

The hyperparameters for the sequential model that is used are shown in Table 3. These parameters are selected after the experimentation.

Table 2. Hyperparameters of the Training Model

Epochs	40
Batch Size	64
MAX_SEQ_LENGTH	32
NUM_FEATURES	2048
Learning rate	0.001

3.4 ALERT GENERATION

When a suspicious activity has been detected the system will automatically generate the alert by sending the suspicious activity of the student via email using the Python library SMTP [31] to the regulatory authorities so that necessary steps should be taken to maintain academic integrity. The Google server is used for sending an email. The video of suspicious activity is also stored in the SQLite database. This whole process is explained with the help of a diagram illustrated in Figure 5.

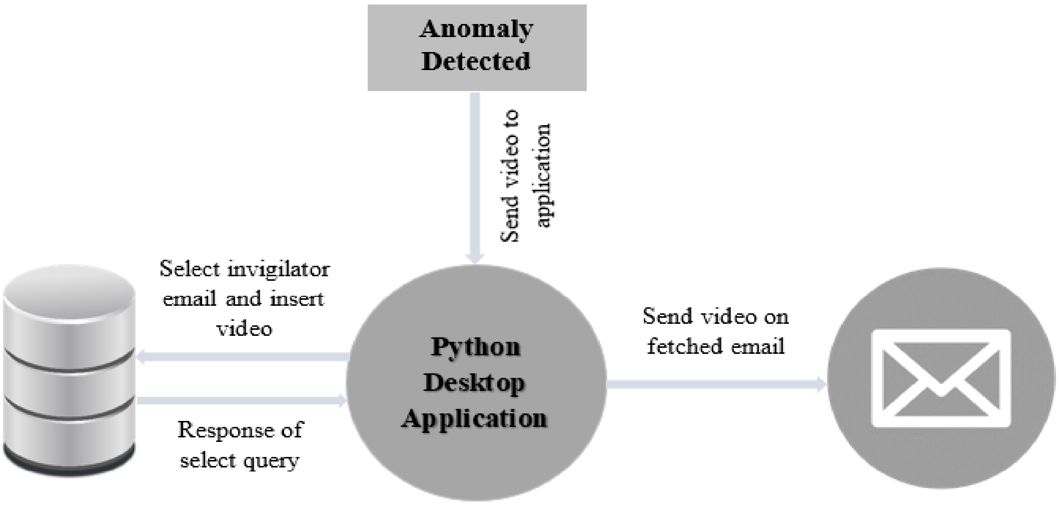

Anomaly Detected
Send video to application
Select invigilator email and insert video
Response of select query
Python Desktop Application
Send video on fetched email

Figure 5. Alert generation process

3.5 MARK ATTENDANCE

The smart attendance system uses face recognition technology which can provide an additional layer of security and helps to prevent cheating in exams. The process of face recognition involves several interrelated problems [32].

- Initially, the system must identify and locate all the faces present in a given image.
- Then, it needs to analyze each face, accounting for variations in orientation and lighting, and recognize the person behind it.
- This involves identifying distinctive features, such as eye size or face length that distinguish one individual from another.
- Finally, the system must compare these features against a database of known individuals to identify the person in question.

Humans have an innate ability to recognize faces automatically and almost instantaneously. However, this ability is so advanced that humans often perceive faces in ordinary objects. Unlike humans, computers do not possess this level of generalization and need to be trained to execute each step of the face recognition process individually [33]. To implement face recognition, a pipeline approach can be used where each step is solved separately, and the output of one step is passed as input to the next step. The pipeline consists of four main steps:

1. Find Faces from videos

   The system needs to identify and locate a face in a photo captured by a camera during a live video. It uses the Histogram of Oriented Gradients (HOG) method which was invented in 2005 [34] for detecting faces from videos. The output of face detection is shown in Figure 6.

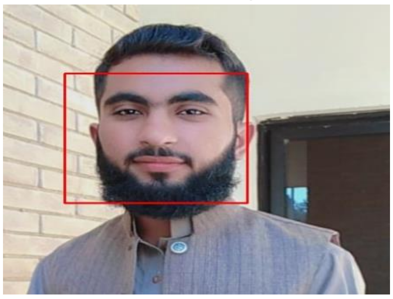
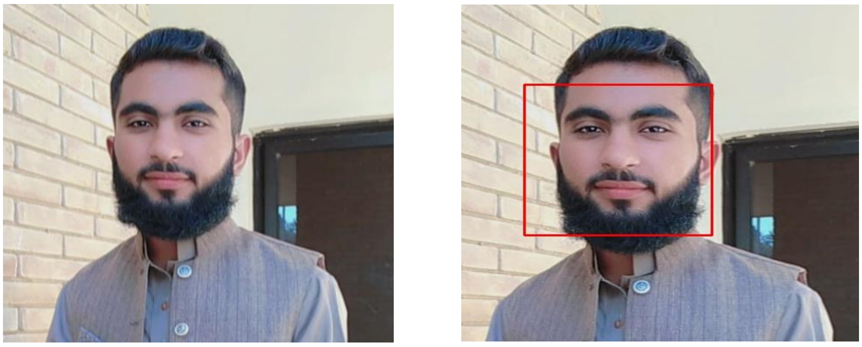


Figure 6. Face detection

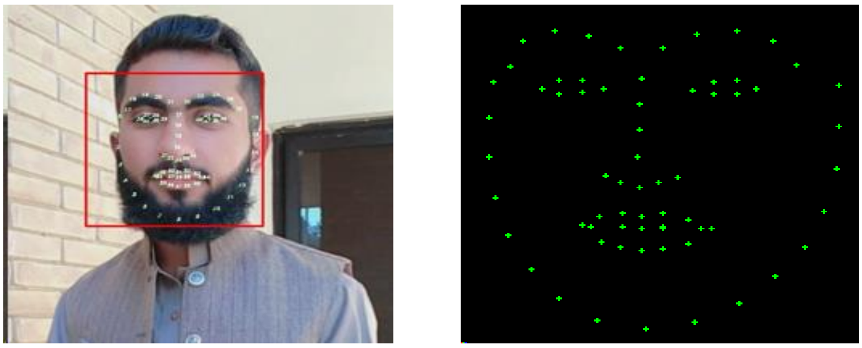
2. Analyze Facial Features

   The facial features of the identified face are analyzed, taking into account variations in orientation and lighting, among other factors. Facial landmark detection involves the identification of 68 key points on a person's face, including the top of the chin, the inner corners of the eyebrows, the outer edges of the eyes, and the lips. These points are used to wrap an image around the face, and the model is trained to accurately locate each of these 68 points. Figure 7 shows the complete process of analyzing facial features.



a

b



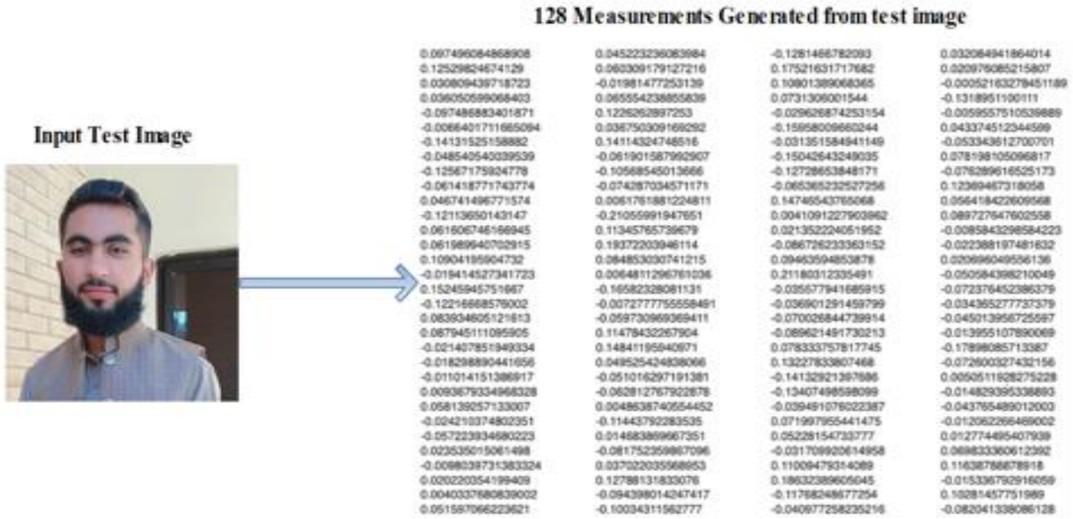
c

d

Figure 7. Analyzing Facial Features a) Input test image b) Face detection from the frames of the video c) Locate 68 Landmarks on each face d) Face Landmarks identification

3. Encoding Face Images

   The system compares the facial features of the identified face with those of familiar faces in a database to determine whether the person is known or unknown. The CNN (Convolutional Neural Network) model is designed to generate a set of 128 facial measurements, known as face encodings, through the process of training. These encodings are calculated for each face and can be used to represent and differentiate faces from one another. It will generate roughly the same encodings when two different images of the same person are processed [33]. The measurements of the tested image are shown in Figure 8.


Input Test Image

128 Measurements Generated from test image

0.097496084868908
0.12529824674129
0.030809439718723
0.036050599068403
-0.097486883401871
-0.0066401711665094
-0.14131525158882
-0.048540540039539
-0.12567175924778
-0.061418771743774
0.046741496771574
-0.12113650143147
0.061606746166945
0.061989640702915
0.10904195904732
-0.019414527341723
0.15245945751667
-0.12216668576002
0.083934605121613
0.087945111095905
-0.021407851949334
-0.018298890441656
-0.011014151386917
0.0093679334968328
0.058139257133007
-0.024210374802351
-0.057223934680223
0.023535015061498
-0.0098039731383324
0.020220354199409
0.0040337680839002
0.051597066223621

0.045223236083984
0.060309179127216
-0.01981477253139
0.065554238855839
0.12262628897253
0.036750309169292
0.14114324748516
-0.061901587992907
-0.10568545013666
-0.074287034571171
0.0061761881224811
-0.21055991947651
0.11345765739679
0.19372203946114
0.084853030741215
0.0064811296761036
-0.16582328081131
-0.0072777755558491
-0.059730969369411
0.11478432267904
0.14841195940671
0.049525424838066
-0.051016297191381
-0.062812767922878
0.0048638740554452
-0.11443792283535
0.014683869667351
-0.081752359867096
0.037022035568953
0.12788131833076
-0.094398014247417
-0.10034311562777

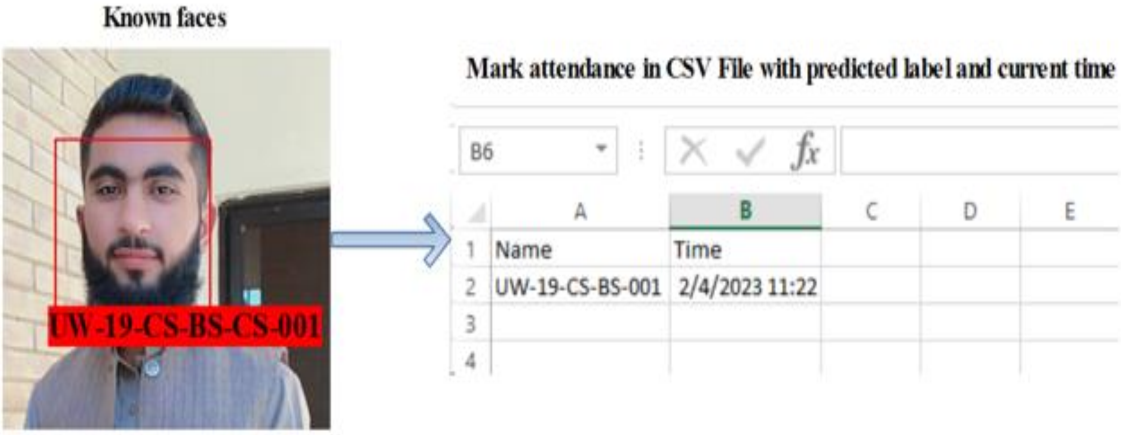
-0.1281466782093
0.17521631717682
0.10801389068365
0.0731306001544
-0.029626874253154
-0.15958009660244
-0.031351584941149
-0.15042643249035
-0.12728653848171
-0.065365232527256
0.14746543765068
0.0041091227903862
0.021352224051952
-0.086726233363152
0.094635948853878
0.21180312335491
-0.035577941685915
-0.036901291459799
-0.070026844739914
-0.089621491730213
0.078333757817745
0.13227833807468
-0.14132921397686
-0.13407498598099
-0.039491076022387
0.071997955441475
0.05228154733777
-0.031709920614958
0.11009479314089
0.18632389605045
-0.11768248677254
-0.040977258235216

0.032084941864014
0.020976085215807
-0.00052163278451189
-0.1318951100111
-0.0059557510539889
0.043374512344599
-0.053343612700701
0.078198105096817
-0.076289616525173
0.12369467318058
0.056418422609568
0.089727647602558
-0.0085843298584223
-0.022388197481632
0.020696049556136
-0.050584398210049
-0.072376452386379
-0.034365277737379
-0.045013956725597
-0.013955107890069
-0.17898085713387
-0.072600327432156
0.0050511928275228
-0.014829395338893
-0.043765489012003
-0.012062266469002
0.012774495407939
0.069833360612392
0.11638788678918
-0.015336792916059
0.10281457751989
-0.082041338086128

Figure 8. Measurements of the test image

4. Finding the Person’s Name from Encodings

   The system predicts the identity of the person and retrieves any additional details, such as their name or occupation, from the database and marks the student's attendance in CSV along with the time and registration number. Mark attendance of students using face recognition is shown in Figure 9.


Known faces
UW-19-CS-BS-CS-001
Mark attendance in CSV File with predicted label and current time
B6
fx
A
B
C
D
E

	A	B
1	Name	Time
2	UW-19-CS-BS-001	2/4/2023 11:22
3		
4		

Figure 9. Mark Attendance using Face Recognition

4. RESULTS AND DISCUSSION

The performance of the system is measured in terms of accuracy. The accuracy of the system is a crucial evaluation metric that determines how well the proposed model performs. The system is tested in a real-time environment and it detects the student cheating accurately and generates the real-time alert simultaneously by sending an email. The accuracy, precision, recall, and F1 score of the proposed system for overall classes are shown in Table 4 and the confusion matrix results between pairs of two classes are shown in Table 5. The formulas to find the accuracy, sensitivity, and specificity are as follows:

A. ACCURACY

$$\frac{(\mathrm{T+ive}) + (\mathrm{T-ive})}{(\mathrm{T+ive}) + (\mathrm{T-ive}) + (\mathrm{F+ive}) + (\mathrm{F-ive})} \times 100 \quad \ldots\ldots\ldots\ldots \quad (4.1)$$

B. SENSITIVITY

$$\frac{\mathrm{TP}}{\mathrm{TP + FN}} \quad \ldots\ldots\ldots\ldots \quad (4.2)$$

C. SPECIFICITY

$$\frac{\mathrm{TN}}{\mathrm{TN + FP}} \quad \ldots\ldots\ldots\ldots \quad (4.3)$$

Table 3. Confusion Matrix Overall Results of the Proposed Method

Classes	Accuracy	Precision	Recall	F1 Score
NoCheat	97.92%	0.87	0.96	0.91
LookLeft	95%	0.77	0.82	0.79
LookRight	96.67%	0.83	0.89	0.86
PocketSheet	94.58%	0.80	0.77	0.79
PantsSheet	96.25%	0.80	0.89	0.84
ExchPaper	90%	0.97	0.56	0.71
FaceCodes	95.83%	0.73	0.92	0.81
HandCodes	97.83%	0.77	1.0	87

Table 4 shows the result of the proposed methodology through which we depict the overall accuracy of the system which is 81.67%. The performance of the system in terms of accuracy is also expressed in the form of a histogram as shown in Figure 10.

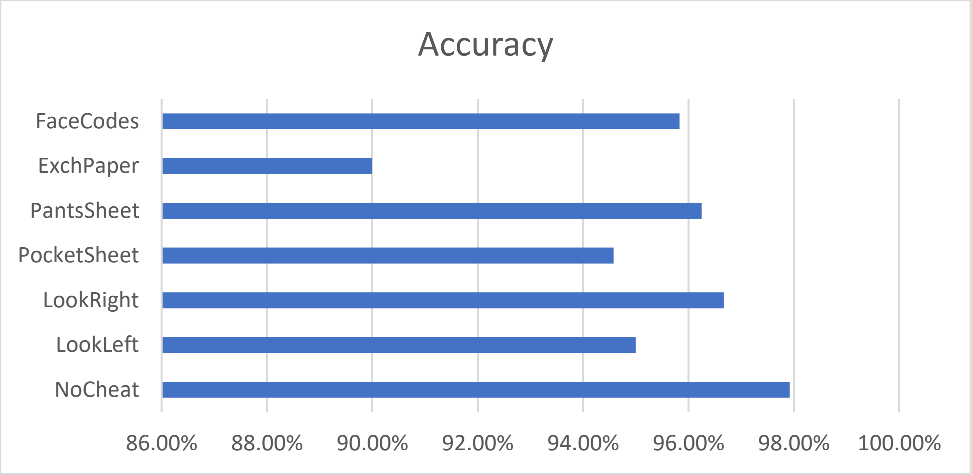


Figure 10. The proposed model histogram in terms of accuracy

Table 4. Confusion Matrix Results between pairs of classes of the Proposed Method

Classes	**Accuracy**	**Precision**	**Recall**	**F1 Score**
No Cheat vs. Look Left	96.92%	0.84	0.94	0.89
No Cheat vs. Look Right	97.67%	0.86	0.91	0.88
No Cheat vs. Pocket Sheet	95.58%	0.82	0.78	0.80
No Cheat vs. Pants Sheet	97.25%	0.82	0.90	0.86
No Cheat vs. Exch Paper	91.0%	0.98	0.58	0.73
No Cheat vs. Face Codes	96.83%	0.75	0.93	0.83
No Cheat vs. Hand Codes	98.83%	0.80	1.00	0.89
Look Left vs. Look Right	96.67%	0.83	0.89	0.86
Look Left vs. Pocket Sheet	94.08%	0.79	0.76	0.77
Look Left vs. Pants Sheet	96.00%	0.79	0.88	0.83
Look Left vs. Exch Paper	90.0%	0.97	0.56	0.71
Look Left vs. Face Codes	95.83%	0.73	0.92	0.81
Look Left vs. Hand Codes	97.83%	0.77	1.00	0.87
Look Right vs. Pocket Sheet	95.33%	0.81	0.76	0.78
Look Right vs. Pants Sheet	96.75%	0.81	0.90	0.85
Look Right vs. Exch Paper	90.0%	0.97	0.56	0.71
Look Right vs. Face Codes	95.83%	0.73	0.92	0.81
Look Right vs. Hand Codes	97.83%	0.77	1.00	0.87
Pocket Sheet vs. Pants Sheet	95.00%	0.80	0.76	0.78
Pocket Sheet vs. Exch Paper	90.0%	0.97	0.56	0.71
Pocket Sheet vs. Face Codes	95.83%	0.73	0.92	0.81
Pocket Sheet vs. Hand Codes	97.83%	0.77	1.00	0.87
Pants Sheet vs. Exch Paper	90.0%	0.97	0.56	0.71
Pants Sheet vs. Face Codes	95.83%	0.73	0.92	0.81
Pants Sheet vs. Hand Codes	97.83%	0.77	1.00	0.87
Exch Paper vs. Face Codes	90.0%	0.97	0.56	0.71
Exch Paper vs. Hand Codes	97.83%	0.77	1.00	0.87
Face Codes vs. Hand Codes	96.83%	0.75	0.93	0.83

In Table 6, we compare the results with the existing research that uses different methodologies. Furthermore, existing research methodologies with their limitations are shown in Table 2.

Table 5. Results Comparison with Existing Methods

Ref. No	**Year**	**Accuracy**
[14]	2023	89.1%
[35]	2023	90.0%
[16]	2022	63% of the test dataset
[6]	2020	77.8%
[18]	2019	below 75%
Proposed accuracy		90.67%

In Table 5, a comparative analysis is conducted, showcasing the proposed system's notable performance of prior research methodologies. The proposed system demonstrates substantial improvement over earlier studies: one achieved 63% accuracy, another showed 77.8% accuracy, and the third reported an accuracy below 75%. This comparison underscores the significant advancements and heightened efficacy of the proposed methodology in identifying and mitigating cheating behaviors among students.

The results presented in Tables 3, 4, and 5 collectively demonstrate the robustness and effectiveness of the proposed system in detecting various forms of cheating behavior among students. With an overall accuracy of 90.67% the system showcases promising potential in real-time cheating detection, outperforming prior methodologies as evidenced by the comparative analysis. These findings underscore the significance of the proposed model in maintaining academic integrity and upholding examination standards.

5. CONCLUSION

The system that we developed provides a promising solution for detecting cheating during physical exams. The system can detect suspicious behaviors, such as looking away from the exam or using unauthorized materials, and provides a reliable and objective method of identifying cheating. The system not only detects and identifies potential cheaters but also ensures accurate attendance monitoring through face recognition. The implementation of this system can significantly improve the integrity and reliability of physical exams while reducing the workload of invigilators. Furthermore, the system's real-time monitoring and alert system enables immediate intervention by exam proctors, allowing for the timely resolution of potential cheating incidents. While further testing and refinement of the system are necessary, this study's results demonstrate the potential of E-Cheating Detection to enhance exam integrity and ensure a fair and equitable testing environment for all students. In the future, many other functionalities can be added to the system by incorporating additional features, such as eye-tracking data or audio analysis, to provide a more comprehensive picture of a student's behavior during the exam. We can also train this model on more datasets so that the accuracy of the system should be exceptional. Furthermore, this system can be integrated with existing learning management systems or exam software to streamline the implementation and adoption of the system in educational institutions.

6. REFERENCES

[1] S. Zhao *et al.*, "VESPA: A General System for Vision-Based Extrasensory Perception Anti-Cheating in Online FPS Games," *IEEE Transactions on Games,* pp. 1-10, 2023.

[2] M. L. Farnese, C. Tramontano, R. Fida, and M. Paciello, "Cheating behaviors in academic context: Does academic moral disengagement matter?," *Procedia-Social and Behavioral Sciences,* vol. 29, pp. 356-365, 2011.

[3] D. L. McCabe, L. K. Treviño, and K. D. Butterfield, "Cheating in academic institutions: A decade of research," *Ethics &Behavior,* vol. 11, no. 3, pp. 219-232, 2001.

[4] P. M. Newton and K. Essex, "How common is cheating in online exams and did it increase during the COVID-19 pandemic? A Systematic Review," *Journal of Academic Ethics,* pp. 1-21, 2023.

[5] D. A. Odongo, E. Agyemang, and J. B. Forkuor, "Innovative approaches to cheating: An exploration of examination cheating techniques among tertiary students," *Education Research International,* vol. 2021, pp. 1-7, 2021.

[6] J. Nishchal, S. Reddy, and P. N. Navya, "Automated Cheating Detection in Exams using Posture and Emotion Analysis," in *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT),* 2-4 July 2020, pp. 1-6,2020.

[7] Y. Ma, D. L. McCabe, and R. Liu, "Students' academic cheating in Chinese universities: Prevalence, influencing factors, and proposed action," *Journal of Academic ethics,* vol. 11, no. 3, pp. 169-184, 2013.

[8] R. Bawarith, A. Basuhail, A. Fattouh, and S. Gamalel-Din, "E-exam cheating detection system," *International Journal of Advanced Computer Science and Applications,* vol. 8, no. 4, pp. 1-6, 2017.

[9] Y. Atoum, L. Chen, A. X. Liu, S. D. Hsu, and X. Liu, "Automated online exam proctoring," *IEEE Transactions on Multimedia,* vol. 19, no. 7, pp. 1609-1624, 2017.

[10] F. Choo and K. Tan, "Abrupt academic dishonesty: Pressure, opportunity, and deterrence," *The International Journal of Management Education,* vol. 21, no. 2, pp. 1-15, 2023.

[11] F. Ozdamli, A. Aljarrah, D. Karagozlu, and M. Ababneh, "Facial Recognition System to Detect Student Emotions and Cheating in Distance Learning," *Sustainability,* vol. 14, no. 20, p. 13230, 2022. [Online]. Available: https://www.mdpi.com/2071-1050/14/20/13230.

[12] N. Gupta and B. B. Agarwal, "Suspicious Activity Classification in Classrooms using Deep Learning," *Engineering, Technology & Applied Science Research,* vol. 13, no. 6, pp. 12226-12230, 2023.

[13] A. Nigam, R. Pasricha, T. Singh, and P. Churi, "A Systematic Review on AI-based Proctoring Systems: Past, Present and Future," *Education and Information Technologies,* vol. 26, pp. 6421-6445, 09/01 2021.

[14] T. Liu, "AI proctoring for offline examinations with 2-Longitudinal-Stream Convolutional Neural Networks," *Computers and Education: Artificial Intelligence,* vol. 4, pp. 1-15, 2023.

[15] L. Baran and P. K. Jonason, "Academic dishonesty among university students: The roles of the psychopathy, motivation, and self-efficacy," *PLOS ONE,* vol. 15, no. 8, pp. 1-18, 2020.

[16] R. a. M. Al_airaji, I. A. Aljazaery, H. T. S. Alrikabi, and A. H. M. Alaidi, "Automated Cheating Detection based on Video Surveillance in the Examination Classes," *International Journal of Interactive Mobile Technologies (iJIM),* vol. 16, no. 08, pp. pp. 124-137, 04/26 2022.

[17] X. G. Yu, J. Y. Sun, B. He, J. J. Zhuang, and Z. C. Dai, "Design and Implementation of Automatic Invigilation Functions Using the Embedded Technology," *Procedia Computer Science,* vol. 166, pp. 41-45, 2020/01/01/ 2020.

[18] M. Adil, R. Simon, and S. K. Khatri, "Automated Invigilation System for Detection of Suspicious Activities during Examination," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 4-6 Feb. 2019 , pp. 361-366, 2019.

[19] A. Arinaldi and M. I. Fanany, "Cheating video description based on sequences of gestures," in *2017 5th International Conference on Information and Communication Technology (ICoIC7)*, 17-19 May 2017 2017, pp. 1-6,2017.

[20] M. Asadullah and S. Nisar, "An automated technique for cheating detection," in *2016 Sixth International Conference on Innovative Computing Technology (INTECH)*, 24-26 Aug. 2016, pp. 251-255, 2016.

[21] F. Kamalov, H. Sulieman, and D. Santandreu Calonge, "Machine learning based approach to exam cheating detection," *PLOS ONE*, vol. 16, no. 8, pp. 1-15, 2021.

[22] 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.

[23] D. Felsinger, T. Halloluwa, and I. Fonseka, "Video based action detection for online exam proctoring in resource-constrained settings," *Education and Information Technologies,* pp. 1-15, 2023.

[24] M. Garg and A. Goel, "A systematic literature review on online assessment security: Current challenges and integrity strategies," *Computers & Security,* vol. 113, p. 1-14, 2022.

[25] M. Garg and A. Goel, "Preserving integrity in online assessment using feature engineering and machine learning," *Expert Systems with Applications,* vol. 225, p. 1-12, 2023.

[26] J. A. Oravec, "AI, Biometric Analysis, and Emerging Cheating Detection Systems: The Engineering of Academic Integrity?," *Education Policy Analysis Archives,* vol. 30, no. 175, pp. 1-18, 2022.

[27] S. Wang, D. Wu, and X. Zheng, "TBC-YOLOv7: a refined YOLOv7-based algorithm for tea bud grading detection," *Frontiers in Plant Science,* vol. 14, pp. 1-7 2023.

[28] A. B. Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," 6 July, 2023. [Online]. Available: https://arxiv.org/abs/2207.02696

[29] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, 17-20 Sept. 2017, pp. 3645-3649, 2017.

[30] C. Fan *et al.*, "ICaps-ResLSTM: Improved capsule network and residual LSTM for EEG emotion recognition," *Biomedical Signal Processing and Control,* vol. 87, p. 1-15, 2024.

[31] R. S. Priya, S. Shanmugavadivel, M. S. M. Sivaraja, and M. S. Francis, "Network based Learning Platform Application Model for Enhancement of Realtime Working Systems," in *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 2023: IEEE, pp. 1234-1239.

[32] J. A. Mahajan and A. N. Paithane, "Face detection on distorted images by using quality HOG features," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 10-11 March 2017, pp. 439-444, 2017.

[33] S. Syed Danish Ahmad, U. Muhammand, and G. Eman, "An Intelligent Health Control Security Robotic System," *University of Wah Journal of Computer Science,* vol. 4, no. 1, pp. 17-30, 01/02 2023. [Online]. Available: https://uwjcs.org.pk/index.php/ojs/article/view/55.

[34] A. Liang, W. Liu, L. Li, M. R. Farid, and V. Le, "Accurate facial landmarks detection for frontal faces with extended tree-structured models," in *2014 22nd International Conference on Pattern Recognition*, 2014: IEEE, pp. 538-543.

[35] W. Alsabhan, "Student Cheating Detection in Higher Education by Implementing Machine Learning and LSTM Techniques," *Sensors,* vol. 23, no. 8, pp. 1-19, 2023.