

**MASTER THESIS NO. 2023: 19**

**College of Information Technology**

**Department of Information Systems and Security**

**CHEATING DETECTION IN ONLINE EXAMS BASED  
ON CAPTURED VIDEO USING DEEP LEARNING**

*Aysha Sultan Alkalbani*



*April 2023*

United Arab Emirates University  
College of Information Technology  
Department of Information Systems and Security

CHEATING DETECTION IN ONLINE EXAMS BASED ON  
CAPTURED VIDEO USING DEEP LEARNING

Aysha Sultan Alkalbani

This thesis is submitted in partial fulfilment of the requirements for the degree of Master  
of Science in Information Technology Management

April 2023

**United Arab Emirates University Master Thesis**  
**2023: 19**

Cover: Different Deep Learning techniques used to detect cheating

(Photo: <https://images.app.goo.gl/G2wLzEv6rQEuPK9A6>)


© 2023 Aysha Sultan Alkalbani, Al Ain, UAE

All Rights Reserved

Print: University Print Service, UAEU 2023

## Declaration of Original Work

I, Aysha Sultan Alkalbani, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled “*Cheating Detection in Online Exams based on Captured Video using Deep Learning*”, hereby, solemnly declare that this is the original research work done by me under the supervision of Dr. Mohammad Mehedy Masud, in the College of Information Technology at UAEU. This work has not previously formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature:  \_\_\_\_\_

Date: 13/March/2023



## **Advisory Committee**

1) Advisor: Dr. Mohammad Mehedy Masud

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

2) Member: Amir Ahmad

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

## Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

- 1) Advisor (Committee Chair): Dr. Mohammad Mehedy Masud

Title: Associate Professor

Department of Information Systems and Security

College of Information Technology

Signature *M. Masud* Date 23/04/2023

- 2) Member: Dr. Fady Alnajjar

Title: Associate Professor

Department of Computer Science and Software Engineering

College of Information Technology

Signature *Fady Alnajjar* Date 25/04/2023

- 3) Member (External Examiner): Prof. Adel Al-Jumaily

Title: Professor

Department: Faculty of Engineering

Institution: University of Technology Brunei, Brunei

Signature *Adel Al-Jumaily* Date 26/04/2023

This Master Thesis is accepted by:

Dean of the College of Information Technology: Professor Taieb Znati

Signature Taieb Znati

Date 20/06/2023

Dean of the College of Graduate Studies: Professor Ali Al-Marzouqi

Signature Ali Hassan

Date 22/06/2023

## Abstract

Today, e-learning has become a reality and a global trend imposed and accelerated by the COVID-19 pandemic. However, there are many risks and challenges related to the credibility of online exams which are of widespread concern to educational institutions around the world. Online exam system continues to gain popularity, particularly during the pandemic, due to the rapid expansion of digitalization and globalization. To protect the integrity of the examination and provide objective and fair results, cheating detection and prevention in examination systems is a must. Therefore, the main objective of this thesis is to develop an effective way of detection of cheating in online exams. In this work, a system to track and prevent attempts to cheat on online exams is developed using artificial intelligence techniques. The suggested solution uses the webcam that is already connected to the computer to record videos of the examinee in real time and afterwards analyze them using different deep learning methods to find best combinations of models for face detection and classification if cheating/not cheating occurred. To evaluate the system, we use a benchmark dataset of exam videos from 24 participants who represented examinees in online exam. An object detection technique is used to detect face appeared in the image and crop the face portion, and then a deep learning based classification model is trained from the images to classify a face as cheating or not cheating. We have proposed an effective combination of data preprocessing, object detection, and classification models to obtain high detection accuracy. We believe that the suggested invigilation methodology can be used in colleges, institutions, and schools to look for and keep an eye on suspicious student behavior. Hopefully, by putting the proposed invigilation method into place, we can aid in eliminating and reducing cheating incidences as it undermines the integrity and fairness of the educational system.

**Keywords:** e-cheating detection, intelligent system, deep learning.

## Title and Abstract (in Arabic)

كشف الغش في الاختبارات عبر الإنترنت بناء على فيديو تم التقاطه باستخدام التعلم العميق

### الملخص

اليوم، أصبح التعلم الإلكتروني حقيقة واتجاهًا عالميًا فرضته جائحة كورونا وتسارعها. ومع ذلك، هناك العديد من المخاطر والتحديات المتعلقة بمصداقية الامتحانات عبر الإنترنت والتي تشكل مصدر قلق واسع النطاق للمؤسسات التعليمية في جميع أنحاء العالم. يستمر نظام الامتحانات عبر الإنترنت في اكتساب شعبية، لا سيما أثناء انتشار الأوبئة، بسبب التوسع السريع للرقمنة والعولمة. ولأهمية حماية نزاهة الامتحانات وتقديم نتائج موضوعية وعادلة، فإن كشف الغش والوقاية منه في أنظمة الامتحانات والمراقبة أمر لا بد منه. لذلك، فإن الهدف الرئيسي من هذه الأطروحة هو تطوير طريقة فعالة للكشف عن الغش في الامتحانات عبر الإنترنت. في هذا العمل، تم تطوير نظام لتتبع ومنع محاولات الغش في الاختبارات عبر الإنترنت باستخدام تقنيات الذكاء الاصطناعي. يستخدم الحل المقترح كاميرا الويب المتصلة بالفعل بجهاز الكمبيوتر الخاص بالفاحص لتسجيل مقاطع فيديو للممتحنين للاختبار في الوقت الفعلي ثم تحليلها بعد ذلك باستخدام طرق التعلم العميق المختلفة للعثور على أفضل مجموعات من النماذج لاكتشاف الوجه وتصنيفه في حالة حدوث الغش / عدم الغش. لتقييم النظام، نستخدم مجموعة بيانات معيارية من مقاطع فيديو الامتحان من 24 مشاركًا يمثلون الممتحنين في الامتحان عبر الإنترنت. يتم استخدام تقنية الكشف عن الكائن لاكتشاف الوجه الظاهر في الصورة واقتصاص جزء الوجه، ثم يتم تدريب نموذج تصنيف قائم على التعلم العميق من الصور لتصنيف الوجه على أنه غش أو ليس غش. لقد اقترحنا توليفة فعالة من نماذج المعالجة المسبقة للبيانات، واكتشاف الأشياء، والتصنيف للحصول على دقة عالية في الكشف. نعتقد أنه يمكن استخدام منهجية المراقبة المقترحة في الكليات والمؤسسات والمدارس للبحث عن سلوك الطلاب المشبوه ومراقبته. نأمل، من خلال وضع طريقة المراقبة المقترحة في مكانها، يمكننا المساعدة في القضاء على حوادث الغش والحد منها لأنها تضعف النزاهة وعدالة النظام التعليمي.

**مفاهيم البحث الرئيسية:** كشف الغش الإلكتروني، نظام ذكي، التعلم العميق.

## **Acknowledgements**

I would like to thank my committee for their guidance, support, and assistance throughout my preparation of this thesis, especially my advisor Prof. Mohammad Mehedy Masud for his guidance and advice to make this project possible.

Special thanks go to my family who support me through all the journeys.

## **Dedication**

*To my beloved parents and family*

## Table of Contents

Title .....	i
Declaration of Original Work.....	iii
Advisory Committee .....	iv
Approval of the Master Thesis .....	v
Abstract .....	vii
Title and Abstract (in Arabic) .....	viii
Acknowledgements .....	ix
Dedication .....	x
Table of Contents .....	xi
List of Tables.....	xiii
List of Figures .....	xiv
List of Abbreviations.....	xv
Chapter 1: Introduction .....	1
1.1 Overview.....	1
1.2 Statement of the Problem.....	2
1.2.1 Research Questions .....	2
1.3 Research Objectives and Contributions .....	3
1.4 Background.....	4
1.4.1 Machine Learning.....	4
Chapter 2: Literature Review .....	11
2.1 Cheating Detection without Machine Learning.....	11
2.2 Cheating Detection with Traditional Machine Learning .....	15
2.3 Cheating Detection with Deep Learning.....	19
Chapter 3: Research Methodology .....	29
3.1 Faster RCNN and Bi-GRU .....	30
3.2 SSD MobileNet and Bi-GRU.....	33
3.3 Faster RCNN and CNN-LSTM .....	34
3.4 YOLOv5 and CNN .....	36
3.5 YOLOv5 and Bi-GRU .....	39
Chapter 4: Dataset .....	40
Chapter 5: Experiments and Results .....	45



5.1 Experimental Setup .....	45
5.1.1 Dataset .....	45
5.1.2 Software.....	45
5.1.3 Evaluation Setup.....	45
5.1.4 Parameter Setting .....	46
5.2 Object Detection Techniques .....	47
5.2.1 Faster RCNN .....	47
5.2.2 SSD MobileNet .....	50
5.2.3 YOLOv5 .....	53
5.3 Classification Techniques .....	58
5.3.1 Bi-GRU .....	58
5.3.2 CNN-LSTM.....	61
5.4 Combinations of 5 Proposed Techniques .....	66
5.5 Parameters Tunning .....	69
5.5.1 Faster RCNN .....	69
5.5.2 SSD MobileNet .....	70
5.5.3 YOLOv5 .....	71
5.5.4 Bi-GRU .....	72
5.5.5 CNN-LSTM.....	73
5.5.6 CNN.....	74
5.5.7 Best Accuracy for All Models.....	75
5.6 Runtime Performance .....	76
5.7 Summary and Analysis of Result.....	77
Chapter 6: Conclusion .....	79
References .....	80

## List of Tables

Table 1: Cheating detection with Traditional Machine Learning .....	18
Table 2: Cheating detection with Deep Learning.....	26
Table 3: Dataset Information.....	43
Table 4: Types of Cheating .....	44
Table 5: Faster RCNN Results .....	50
Table 6: SSD MobileNet Results .....	53
Table 7: YOLOv5 Results.....	56
Table 8: Summary of detection models.....	57
Table 9: Statistical Analysis using t-test for object detection .....	58
Table 10: Bi-GRU Results .....	58
Table 11: CNN-LSTM Results.....	61
Table 12: CNN Results.....	63
Table 13: Summary of classification models .....	65
Table 14: Statistical Analysis using t-test for classification.....	66
Table 15: All models accuracy .....	67
Table 16: Statistical Analysis using t-test for combined model.....	68
Table 17: Faster-RCNN parameters tuning.....	70
Table 18: SSD MobileNet parameters tuning .....	71
Table 19: YOLOv5 parameters tuning.....	72
Table 20: Bi-GRU parameters tuning .....	72
Table 21: CNN-LSTM parameters tuning.....	73
Table 22: CNN parameters tuning.....	74
Table 23: Improved accuracy .....	76
Table 24: Runtime performance.....	76
Table 25: Training time for each algorithm .....	77

## List of Figures

Figure 1: CNN Architecture .....	5
Figure 2: Faster R-CNN: Regions with CNN Features .....	6
Figure 3: Bi-GRU Architecture .....	7
Figure 4: YOLO Architecture for Object Detection and Localization .....	9
Figure 5: The general architecture of MobileNet .....	10
Figure 6: Face detection using Faster RCNN .....	31
Figure 7: Face Classification using Bi-GRU .....	32
Figure 8: Face Detection using SSD MobileNet .....	34
Figure 9: Face Classification using CNN-LSTM .....	35
Figure 10: Face Detection using YOLOv5 .....	37
Figure 11: Face Classification using CNN .....	38
Figure 12: Video extraction .....	41
Figure 13: Input image to the object detection model .....	42
Figure 14: Object detection and classification process .....	42
Figure 15: Faster RCNN loss curves for all folds .....	49
Figure 16: SSD MobileNet loss curves for all folds .....	52
Figure 17: YOLOv5 loss curves for all folds .....	55
Figure 18: Bi-GRU Loss vs Number of Epochs for all folds .....	60
Figure 19: CNN-LSTM Loss vs Number of Epochs for all folds .....	62
Figure 20: CNN Loss vs Number of Epochs for all folds .....	64
Figure 21: Predicted Cheating/Not Cheating faces using YOLOv5 + CNN .....	69
Figure 22: Accuracy Vs Batch Size for Faster RCNN .....	70
Figure 23: Accuracy Vs Batch Size for SSD MobileNet .....	71
Figure 24: Accuracy Vs Batch Size for YOLOv5 .....	72
Figure 25: Accuracy Vs Batch Size for Bi-GRU .....	73
Figure 26: Accuracy Vs Batch Size for CNN-LSTM .....	74
Figure 27: Accuracy Vs Batch Size for CNN .....	75

## **List of Abbreviations**

AI	Artificial Intelligence
ANN	Artificial Neural Networks
Bi-GRU	Bi-directional Gated Recurrent Unit
Bi-LSTM	Bi-directional Long Short-Term Memory
CNN	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit
LSTM	Long Short-Term Memory
ML	Machine Learning
R-CNN	Region-based Convolutional Neural Networks
RNN	Recurrent Neural Networks
RPN	Region Proposal Network
SSD	Single Shot Detector
SVM	Support Vector Machine
YOLO	You Only Look Once

# Chapter 1: Introduction

## 1.1 Overview

Due to the advantages of distant working conditions and the COVID-19 pandemic, remote examination and job interviews have grown in popularity in recent years. These systems are used by the majority of businesses and educational institutions for both recruitment and online tests. However, conducting exams in a secure setting is one of the major drawbacks of remote examination systems. We offer a solution for online exam cheating analysis in this thesis. The system just requires a video of the candidate during the exam, which is recorded. The cheating detection pipeline is then used to detect the presence of another person, the use of mobile phone, and the state of the candidate if they used external sources. Face detection and classification has been applied from different methods. We used a public video dataset to evaluate the students' performance.

To ensure the authenticity of the online tests, these tests must be proctored in a secure setting. According to research analysis by Hasri, a significant proportion of students hold the belief that online learning facilitates academic dishonesty to a greater extent than conventional modes of learning. It is essential to consistently implement and monitor strict measures to ensure the provision of quality education. In addition, cheating can confer an inequitable advantage to the culprit and potentially distort the accuracy of data regarding students' actual learning outcomes.

Proctoring, on the other hand, is a monotonous and time-consuming process. It simply entails keeping an eye on a single person or a group of individuals during a test to prevent pre-planned cheating. Even after the epidemic has passed, automating the proctoring procedure at universities or colleges might help trained staff operate more efficiently. Automated proctoring systems are a beneficial inclusion to the examination systems.

In addition, the study was conducted to examine academic cheating behaviors and perceived online effectiveness on academic performance during the period of COVID-19 among schools, colleges, and university students in Pakistan (Malik et al., 2023).

According to the findings, a majority of 60% of students acknowledged engaging in cheating practices frequently during online examinations, while 30% of students admitted to having cheated at least once during an online exam. Additional measures are necessary to safeguard academic integrity considering students' engagement in exam cheating.

We conduct comprehensive literature review of the related works, which contains a full examination of the existing strategies. Many methods are utilized to identify cheating in online tests. Until now, some strategies are less accurate, and those that are very accurate have a time-complexity problem since the latter types of procedures require a long time to identify cheating. As a result, it has a negative impact on overall performance.

In this thesis, we propose an effective two-model combination for cheating detection problem, where the first model is used for identification (i.e., face detection) and the second model is used to classify the detected face into cheating or not cheating. To address the concerns raised above, we proposed a number of deep learning strategies based on various model combinations that use video recording during a test to determine whether or not cheating occurred.

## **1.2 Statement of the Problem**

Several cutting-edge methods have been proposed for detecting cheating in recorded videos. In most cases, these methods suffer limitations, in terms of accuracy and computational complexity. Some achieve high detection accuracy, but suffer limitations in terms of runtime overhead. Others succeed to minimize time complexity, but suffer high false alarm rates and low detection accuracy. These shortcomings underscore the need for a new approach, which both minimizes false alarms and computational complexity.

### *1.2.1 Research Questions*

In order to address the above shortcomings, we will first investigate and assess different techniques for detecting cheating in online exams, using recorded videos of the examinee during the examination. Based on this study, we will explore the following

research questions to derive an effective solution to detect cheating in online exams, using recorded video.

- RQ1: What combination of existing Machine Learning or Deep Learning algorithms may provide the basis for the design of a solution with low false alarm rates and minimal computational complexity?
- RQ2: How the basic algorithm derived in (RQ1) can be further improved to achieve higher accuracy?
- RQ3: How the run time performance of the algorithm developed in (RQ2) can be improved?

### **1.3 Research Objectives and Contributions**

The scope of this project is used recorded videos during online exams to detect cheating, and the following study objectives must be met based on the preceding research questions.

1. Conduct literature review and investigate the applicability of different machine learning and deep learning algorithms in cheating detection using exam videos. (RQ1).
2. Propose an advanced of machine learning technique based on different combinations of high performing algorithms found on objective 1 and prove its superiority of other existing solutions (RQ2).
3. Improve the performance of the proposed technique by hyper parameter tuning (RQ3).
4. Investigate different techniques to improve run time performance of the proposed algorithm.

This thesis makes the following contribution:

- Several model combinations for object detection and classification used to have the most accurate combined model.
- We have used a publicly available dataset; we perform several pre-processing.
- We ran a number of experiments to evaluate the efficiency of the proposed work.

- Proposed an efficient and effective technique for online exam cheating detection.

## **1.4 Background**

In this section, we are going to describe the background of different techniques that are used for object detection and cheating detection, such as machine learning, deep learning, and different categories of deep learning algorithms.

### *1.4.1 Machine Learning*

Machine learning is a branch of computer science that tries to replicate human intelligence by learning from its environment. In the modern world of so-called big data, they are regarded as the workhorse. In addition, it facilitates autonomous learning in machines based on prior experiences. Whereas the process entails a variety of algorithms that utilize data to improve their efficacy. On the other hand, machine learning can be classified into three sections. First, supervised learning, which involves training a machine learning model using labeled data. Second, unsupervised learning which is a type of machine learning in which the data lacks labels, and the formation of groups or clusters is based on the similarity of the data. This results in inter-group data that differs from intra-group data (Houssein et al., 2022). Finally, the reinforcement Learning which involves the training of a model through a system of rewards and punishments based on the real-time steps the model selects.

#### *1.4.1.1 Deep Learning*

Deep learning is a computational approach in machine learning that involves using artificial neural networks to acquire and develop representations. In recent times, these techniques have been utilized in various applications, including speech recognition, Natural language processing (NLP), machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs (Sharma et al., 2021). Deep learning goal to build neural network that automatically find patterns for detecting features (Wang et al., 2021). The outcomes produced by these applications are comparable to, and in some cases superior to, human performance. For



example, artificial neural networks were inspired by the information processing and distributed communication nodes observed in biological systems.

#### 1.4.1.2 Convolutional Neural Network

Convolutional Neural Networks (CNN) were initially designed for image recognition, but they have since evolved into a very versatile model that can be applied for a wide range of applications. CNNs can recognize local features in a multidimensional environment (Krizhevsky et al., 2012).

For instance, CNN's will be able to identify particular objects in a picture, such as a wheel or a smile, wherever they are in the picture. Simple CNNs (shown in Figure 1) pass multidimensional data, including pictures, word embeddings, and other types of data, to a convolutional layer, which is composed of several filters that each learn a different feature. It is important to note that these filters are applied sequentially to various parts of the input. Before being sent into a connected layer, output is frequently pooled or sub-sampled to lower dimensions.

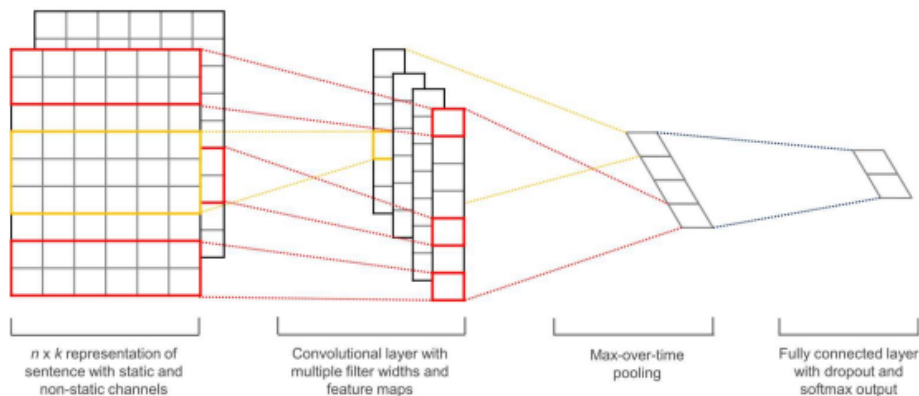


Figure 1: CNN Architecture (Krizhevsky et al., 2012)

#### 1.4.1.3 Faster R-CNN

Region-based Convolutional Neural Network is known to as R-CNN. R-CNN, or Regions with CNN Features, is a bottom-up object identification model that leverages high-capacity CNNs to create bottom-up region recommendations. It uses selective

search to find a number of candidates bounding-box object region candidates (regions of interest), then extracts feature from each region separately for categorization.

The concept of region proposals underpins the R-CNN series. Region suggestions are used to locate objects inside a photograph. The R-CNN technique, which uses pre-trained CNNs to extract visual properties, is slow. Consider the task of picking thousands of region suggestions from a single input image: object recognition necessitates thousands of CNN forward propagations. R-CNNs cannot be widely used in real-world applications because of their high processing requirements. As a result, we use Fast R-CNN since one of the key differences between fast R-CNN and CNN is that R-CNN forward propagation is only done on the entire image.

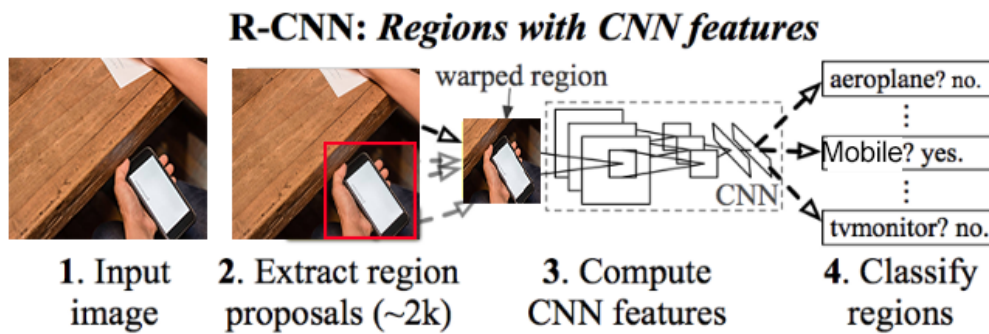


Figure 2: Faster R-CNN: Regions with CNN Features

Figure 2 shows that the RGB data containing cheating material is first fed into a feature extraction network, which extracts regions of interest such as the presence of a mobile phone, eye movement, facial movement, and so on, and then the recovered features are fed into a classifier network (Liu, Chen & Wang, 2018). Before transferring a picture including a mobile phone over a network, as shown in the figure above, it extracts region proposals or regions of interest from the image using a selective search technique. The extracted crops must next be resized (wrapped) and sent across a network. Following that, the system is classed using several classifiers, and a verification process is carried out using the selected characteristics.

#### 1.4.1.4 Bi-directional Gated Recurrent Units (Bi-GRUs)

GRUs, illustrated in Figure 3 are a gating mechanism that has been discovered to be analogous to the LSTM in artificial recurrent neural networks (Liu, Chen, & Wang, 2018). GRUs have demonstrated superior performance on small to medium-sized datasets. Word Net generates word embedding by concatenating the words in a tweet into a single vector using Bi-GRU (Li et al., 2022). The substitution is required to guarantee that all tweets are the same length. Here are the equations and the whole GRU network  $z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$ . where  $z_t$  shows the update gate output at time step  $t$ ,  $\sigma$  presents the sigmoid activation function,  $W_z$  indicates the weight matrix for the current input  $x_t$ ,  $U_z h_{t-1}$  the weight matrix for the previous hidden state  $t-1$  and finally  $b_z$  presents the bias term.

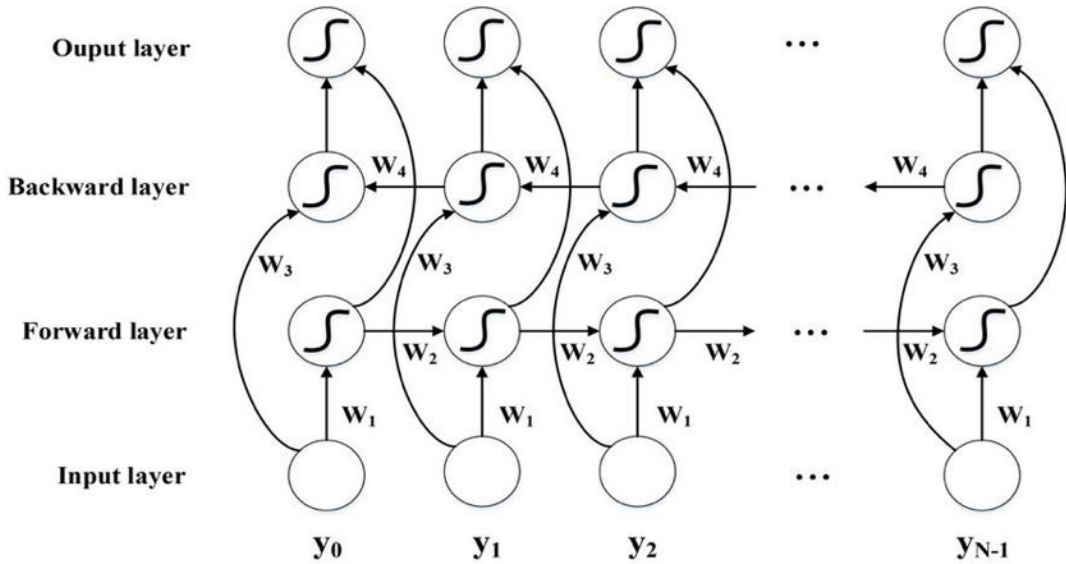


Figure 3: Bi-GRU Architecture (Li et al., 2022)

The primary distinction between GRU and LSTM is that, although LSTM's bag has three gates (input, output, and forget-bag), GRU's bag contains just two gates (reset and update). GRU is easier to understand than LSTM since it contains fewer gates. GRU is preferred when the data set is small; LSTM is suggested when the data set is huge.

#### 1.4.1.5 YOLO

Object detection is a key computer vision task that identifies and localizes regions of interest by incorporating an image and video. There have been numerous ways to

detect objects in image data, but most rely on statistically expensive and time-consuming techniques such as sliding window or region proposal-based algorithms.

Redmon introduced YOLO (You Only Look Once), a novel approach for object recognition and detection to address these challenges. As Figure 4 shows, the novelty is attained by framing object detection as a regression problem to structurally separate bounding boxes and related class probabilities, where a single CNN is trained on the acquired image. More specifically, the YOLO algorithm processes a whole image without requiring region recommendations or several processing phases. It converts the picture into a grid and adds objectness scores and class probabilities to each grid cell in order to generate bounding boxes and class predictions based on CNN's feature maps. YOLO also employs a single loss function that accounts for both localization and classification errors, thus enhancing precision.

In addition, it employs a network for feature extraction, including Darknet-53, to abstract features from the acquired image. Similarly, a single neural network is utilized for the whole image, eliminating the requirement for several passes across the image. Anchor boxes are pre-defined bounding boxes that the model utilizes to make predictions, thus limiting the number of candidates boxes the model needs to consider. In conclusion, YOLO employs a method known as spatial attention to enhance the prominent characteristics of objects and suppress the irrelevant ones. Later its first release, YOLO received various revisions, including YOLOv2, YOLOv3, and YOLOv4. These improvements have enhanced the model's precision and speed and incorporated new features as well.

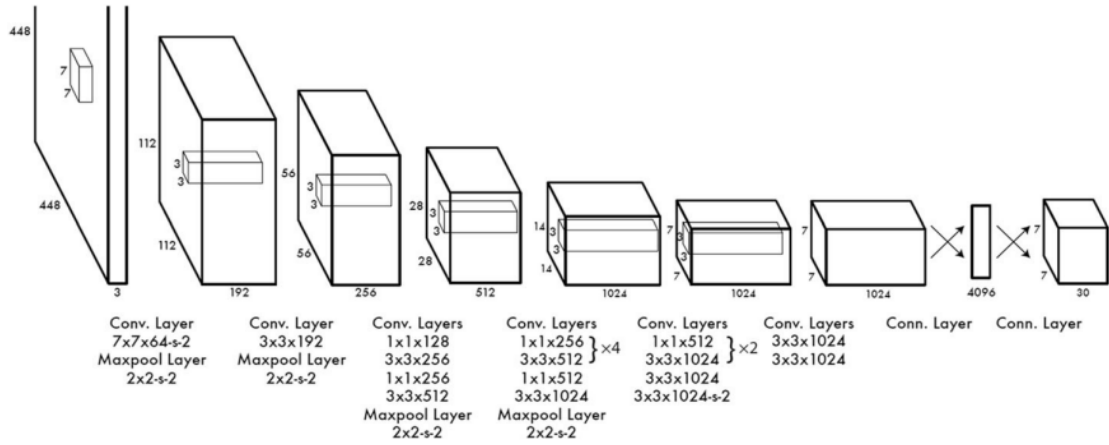


Figure 4: YOLO Architecture for Object Detection and Localization (Redmon et al., 2016)

#### 1.4.1.6 SSD MobileNet

The MobileNet architecture is widely utilized in the field of mobile and embedded vision applications due to its efficient deep neural network design.

The design of the system prioritizes a low weight and computational efficiency, without compromising the ability to achieve high levels of accuracy in tasks such as image classification and object detection.

In addition, it can be integrated with Single Shot Detector (SSD) to enable instantaneous object detection on mobile and embedded devices. The fundamental concept underlying SSD involves utilizing a neural network to forecast a collection of bounding boxes and their corresponding class probabilities directly from the input image (Poddar et al., 2019).

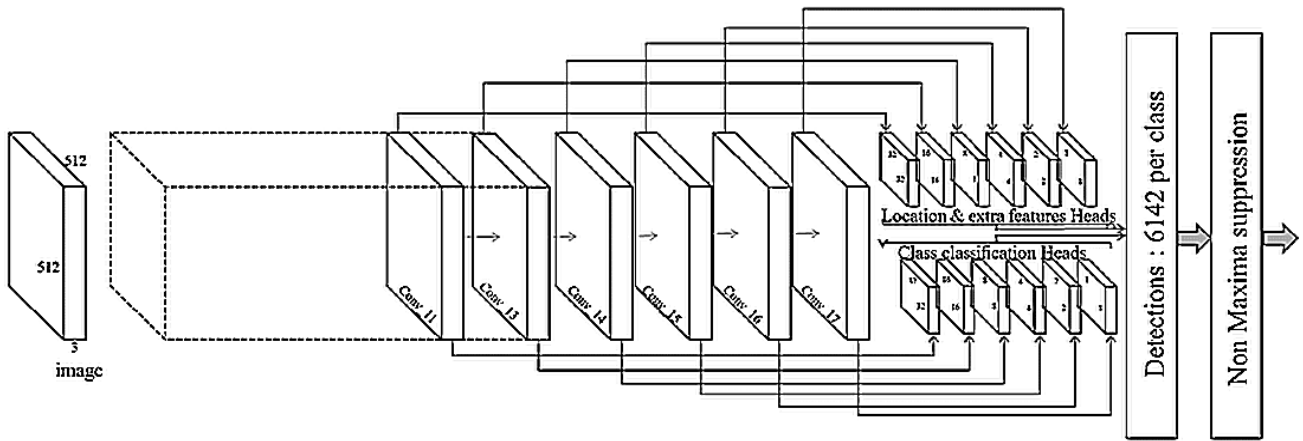


Figure 5: The general architecture of MobileNet (Poddar et al., 2019)

MobileNet's architectural design as Figure 5 shows, it comprises of depth wise separable convolutions that are comprised of two separate convolutional operations, namely a depth wise convolution and a pointwise convolution. The application of depth wise convolution involves the individual application of a singular filter to each distinct input channel, resulting in the generation of a collection of feature maps. The procedure results in a decrease in the quantity of parameters and computations in contrast to a conventional convolution, wherein an individual filter is employed for every input channel. After the depth wise convolution, a collection of 1x1 filters is employed by the pointwise convolution to process the resulting output.

The procedure executes a linear incorporation of the characteristic maps, resulting in a fresh collection of characteristic maps with a distinct count of channels.

MobileNet employs depth wise separable convolutions to attain a comparable degree of precision to other convolutional neural network architectures, such as VGG and ResNet, while necessitating a reduced number of variables and calculations.

The rest of the thesis is organized as follows. Chapter 2 discusses the relevant literature works, then Chapter 3 explains the research methodology. Chapter 4 describes the datasets used and the preprocessing done on them, followed by Chapter 5, which reports the experiments, results, and analysis. Finally, Chapter 6 concludes with directions to future work.

## **Chapter 2: Literature Review**

The present study's literature review section is organized into three distinct categories: (cheating detection without Machine Learning, cheating detection with Traditional Machine Learning and cheating detection with deep learning).

New cheating tactics have emerged as a result of the introduction of new exam procedures and equipment. The adoption of novel examination protocols and equipment has resulted in the emergence of innovative cheating strategies. Over the past few years, information technology has dramatically aided instructors, which has offered a range of tools such as Learning Management Systems, SCORM modules, and paid and open-source solutions. The electronic generation of course and test materials is valuable to augment pedagogy. However, it is imperative to consider the possible risks associated with novel forms of academic dishonesty. The research delves into the various methods employed by academically dishonest students to engage in cheating during examinations. Additionally, the study underscores the strategies that educators can implement to deter cheating.

Currently, considering the global pandemic, educational institutions are widely employing online examinations as a means of assessing the academic performance of students (Dilini et al., 2021). The difficulty in detecting instances of academic dishonesty arises from the lack of human supervision, as is typically present in traditional in-person examinations. Implementing novel educational resources has enhanced pedagogical methodologies, thereby requiring a heightened emphasis on devising efficacious measures to prevent academic dishonesty.

### **2.1 Cheating Detection without Machine Learning**

The study of Kamalov et al. (2021), presented a new approach to detect possible instances of academic dishonesty during the final examination by conducting a retrospective analysis of the students' grades. The approach considers the pre-final exam academic performance of students, their final exam scores, and the collective performance of the class to arrive at a determination. LSTM models are utilized in conjunction with a KDE-driven approach for detecting outliers to ascertain instances of

possible academic dishonesty. The findings obtained from this research could potentially aid scholars and educational leaders who are concerned with safeguarding the academic authenticity of course evaluations. This study establishes a foundation for forthcoming academic research on outlier detection methods. It suggests a supplementary approach to commercial plagiarism detection software and potentially a less intrusive preventative measure to controversial remotely proctored examinations. It is important to note that the present study had certain limitations in terms of both sample size and context. In addition, it is strongly recommended to conduct additional research and experimentation on the proposed approach and its ability to detect academic integrity violations. The administration of exams in a remote setting presents a significant obstacle in maintaining the academic authenticity of the assessment. The matter at hand holds significance in the contemporary era and is poised to retain its relevance in the times to come. The approach author has developed presents a valuable resource for mitigating concerns related to academic honesty in the context of exams that are administered remotely.

Academic dishonesty is a serious issue while taking tests online. According to experts, cheating on online tests must be prevented at all costs. Their study's findings suggest that the absence of monitoring during online tests might increase the likelihood of misbehavior. Also, the results demonstrate how appropriate technologies, such Web-based proctoring, may be used to counteract cheating and deception in online tests, thus adding another degree of discouragement to the unauthorized use of such data platforms.

In order to proctor online exams on remote learning systems, Asep and Bandung (2019) recommend using continuous user verification. The training data set was gathered from the remote learning online lecture sessions. The authors suggest a way to undertake an incremental training procedure utilizing this data set to increase stability for posture and illumination fluctuations. According to the authors, an accurate, affordable, and practical online exam proctoring for mobile learning is available. The limitations of Smartphone memory and computing capacity, however, limit the application of this approach.



Studies compare online testing systems and their features in-depth. Creating online test settings that can deter cheating by proactive measures, enabling them to cut down on the number of tries without a proctor (Li et al., 2021). In addition, the study proposed a system for automated surveillance has been introduced, which is capable of detecting any atypical conduct displayed by a student during examinations (Al-airaji et al., 2022). The utilization of this particular system holds significant importance in the identification and acknowledgement of atypical conduct (such as cheating) in the context of an academic assessment. The efficiency of this system surpasses that of humans due to the potential for human error to arise from factors such as fatigue or illness, which may hinder the performance of human invigilators.

Some methods, such as integrate autonomous detection systems with human proctors and well-defined workflows (Ketar et al., 2017).

Multi-modal techniques combine visual and audial cues from several cheater analysis paradigms. Evaluating a candidate's hirability based on these characteristics and psychological factors (Shdaifat et al., 2020). This study looked at how test-takers behaved during online examinations to see if they could spot instances of possible cheating. Additionally, the study investigated the impact of time delay and head posture in detecting cheating in a lab based online assessment session. In another study of Chotikakamthorn and Tassanaprasert (2020), the proposed system examines the challenge of administering off-site examinations in a context of unanticipated circumstances, wherein a number of limitations have restricted the range of viable options. A method of proctoring was devised with the objective of offering a solution that can be implemented as extensively as feasible within the limitations imposed by the circumstances. The approach relies on open-source software and complimentary services. The approach utilizes solely hardware and devices that are widely accessible to the intended student population. In addition, a novel proctoring approach utilizing E-Cam and S-Cam, along with a standardized protocol and a portable cross-platform proctor monitoring tool, has been developed and explicated. The outcomes of implementing the suggested tool and methodology in a real-world off-campus examination have been documented and analyzed. The text presents insights gained, and recommendations offered.

To avoid cheating in online exams, online proctoring is required. At the moment, online proctoring is done manually by humans rather than being automated. The present study aimed to examine the efficacy of utilizing a 360-degree security camera as opposed to a conventional webcam in order to improve exam security and reduce the imposition of stressful constraints (Turani et al., 2020). To ascertain this objective, a case study was conducted on a cohort of volunteer students enrolled in the College of Computer Science and Engineering. This paper proposes an automated proctoring model as a solution to eliminate the requirement of real-time proctoring and mitigate scheduling constraints, with the aim of preventing cheating.

In order to determine if Webcam-based proctoring has a deterrent impact on cheating during online tests, Hylton et al. (2016), presented an approach. In this study, the experimental and the control subjects were contrasted. Both groups attended the same course, used the same e-learning platform, had the same instructor, and completed the same online tests. One group had a Web-based proctor, whereas the other did not. The results indicated no significant statistical difference in their scores, despite the non-proctored group having somewhat higher scores. The proctored group took significantly less time to finish the online examinations, which demonstrated a statistically significant difference. Academic dishonesty is a serious issue while taking tests online. Their study's findings suggest that the absence of monitoring during online tests might increase the likelihood of misbehavior. Also, the results demonstrate how appropriate technologies, such Web-based proctoring, may be used to counteract cheating and deception in online tests, thus adding another degree of discouragement to the unauthorized use of such data platforms.

In recent times, the issue of academic dishonesty during undergraduate examinations in 2021 has become a progressively alarming matter, as evidenced by a 116% surge in instances of fraudulent activity reported by the Moroccan Ministry of National Education, Vocational Training, Higher Education, and Scientific Research in comparison to the preceding year. The observed trend can be ascribed to multiple factors (Bilen & Matros, 2021). In order to tackle this matter, two methodologies were proposed for the purpose of detecting cheating, and subsequently, their outcomes were paralleled. The findings indicate that although the second approach exhibited a shorter duration, the

first approach exhibited superior efficacy in identifying instances of academic dishonesty, as evidenced by an average detection precision that was 0.1 higher than that of the second approach. The implications of these findings are noteworthy in terms of devising efficacious strategies to deter academic dishonesty and uphold the authenticity of educational evaluations.

## **2.2 Cheating Detection with Traditional Machine Learning**

Using machine learning approaches, we suggest a novel method for identifying probable instances of exam-day plagiarism. The approach the challenge of locating possible instances of fraud as an outlier identification problem (Bilen & Matros, 2021). To find out whether final test scores are out of the ordinary, they look at the students' performances from continuous assessments. But the student evaluation data demands us to take into account its sequential character, unlike a typical outlier detection problem in machine learning. Applying recurrent neural networks and methods for anomaly identification, they solve this problem. It is expected of students who take their examinations at home to work independently without assistance from others. In reality, nevertheless, a sizeable percentage of students make an effort to evade the requirements of academic honesty by, for example, engaging in contract cheating or digital cheating, which is paying a third party to complete their assignments. Universities utilize tools including remote proctoring, cameras, LockDown Browser (Respondus), plagiarism detection programs like Turnitin, SafeAssign, and iThenticate, as well as monitoring software that runs in the background while the test is being taken.

A system that incorporates one webcam, one wear cam and a microphone to assess the visual and auditory circumstances of the distant test site (Atoum et al., 2017a). The proposed method is made up of several components, including user identification, text and voice recognition, dynamic windows detection, eye gaze estimation, and cell phone detection. In order to detect cheating trends, the authors created a variety of violating situations and logged the data in an Online Examination Proctoring database. To appropriately classify behaviors as cheating or normal, the suggested technique employs binary Support Vector Machines for classifier learning.

Online interview systems employ computer vision and machine learning methodologies, including face verification and object detection, to verify the authenticity of candidate videos. The utilization of face verification guarantees the verification of the identity of the interviewee. In contrast, object detection is employed to identify unapproved objects or persons present during the interview proceedings (Prathish & Bijlani, 2016).

The current investigation thoroughly examines the critical scholarly sources identifying academic dishonesty in virtual examinations (Kasliwal, 2015). Various methods have been utilized to identify instances of academic dishonesty. However, specific approaches have demonstrated diminished precision, whereas others encounter challenges related to computational efficiency that may impede their overall efficacy (Chiang et al., 2022). In order to address the limitations, present in current cheating detection methods, we suggest a two-model approach. The first model focuses on detecting faces, specifically through face detection. The second model involves the classification of the detected faces into two distinct categories: cheating and non-cheating. In order to achieve this objective, we suggest a variety of deep learning techniques that involve diverse model amalgamations, incorporating video footage captured during examinations to identify occurrences of academic dishonesty.

Centers on the analysis of academic growth among students through the utilization of machine learning techniques (Dhilipan et al., 2021). The Binomial logical regression, Decision tree, Entropy, and KNN classifier are utilized for analysis. This procedure can assist the educator in making informed decisions regarding the students' performance and devising more effective strategies to enhance their academic progress. In the future, supplementary features will be incorporated into our dataset in order to enhance its precision.

To effectively detect students' unusual learning processes, or cheating in Massive Open Online Courses, many researchers in the field of education have recently studied outlier detection techniques (Khan et al., 2022; Malhotra & Chhabra, 2022). Outlier detection can be classified into two ways Semi-supervised and unsupervised. An initial dataset that represents the population of unfavorable (non-outlier) observations is

available in a semi-supervised outlier detection method. To determine the boundary of the initial observations' distribution, a machine learning tool like one-class SVM can be taught. New observations are then divided into groups based on how far away from the border they are. The system is trained using unsupervised techniques without the benefit of a pure initial dataset of negative observations.

To identify online cheating using AI methods, an e-cheating intelligence agent that is based on the relationship model (Malhotra & Chhabra, 2022). They construct an IP (Intrusion Prevention) detector and a behavior detector that makes use of a densely connected long short-term based systems, is a novel strategy for the detection of cheating during e-Exams. Due to the Covid-19 pandemic, most governments worldwide are proposing that online examinations be proctored, and this method will assist the proctors in spotting any form of uncertain occurrence during the exam. The use of CNN-based detection, which has a 97 percent accuracy rate, has proven to be more sensitive in identifying any ambiguous behavior on the part of the students during the e-Exam.

This system has been put through its paces in an online test environment, which makes it simple to keep track of the results. Experiments have demonstrated that the proposed method outperforms existing systems. A set of methodologies have been developed for online evaluation and monitoring of deviant behavior using image data (Chuang et al., 2017). During the online test, abnormal behavior such as moving heads and speaking is monitored using convolutional neural network-based head posture estimation and threshold-based lip state evaluation, as well as a combination of decision rules.

A cheating detection pipeline for online interviews and exams, which combines keystroke dynamics analysis, facial recognition, and screen recording to identify suspicious behavior (Ozgen et al., 2021). They trained Support Vector Machine (SVM) classifier with (Histogram of Oriented Gradients) HOG features. The system achieved an accuracy of 96.5% in detecting cheating behavior.

In another study, Alnassar et al. (2021), present study which aimed to examine the OU dataset in order to predict student performance through the application of various machine learning algorithms. The experimental procedure comprised of three primary

phases, namely: (a) data preparation, (b) application of machine learning algorithms, and (c) demonstration of outcomes and critical evaluation. Based on the experimental results, it was observed that the k-NN algorithm exhibited superior performance when compared to both SVM and ANN across various feature permutations. The analysis of historical student data suggests that the type of assessment and the number of previous attempts exert a significant impact on the final academic outcome of the student.

Examination malpractice is intentional misbehavior that goes against official examination guidelines to provide a candidate an undeserved advantage or disadvantage. Table 1 shows some of the traditional machine learning techniques. From the table, we can see that some studies have reported higher accuracies compared to others. However, it's important to note that the reported accuracies may depend on several factors such as the dataset used, the choice of features, algorithms, and the experimental setup.

Table 1: Cheating detection with Traditional Machine Learning

Sr No	Title	Year	Publisher	Algorithms	Accuracy %
1	Cheating Detection Pipeline for Online Interviews and Exams	2021	Özgen et al.	HOG-based SVM classifier model	96.5
2	How Well a Student Performed? A Machine Learning Approach to Classify Students' Performance on Virtual Learning Environment	2021	Alnassar et al.	k-Nearest Neighbour (k-NN) and Artificial Neural Network (ANN)	96.27
3	Prediction of Students Performance using Machine learning	2021	Dhilipan et al.	Binomial logical regression, Decision tree, and Entropy and KNN classifier	97.05
4	Detecting probable cheating during online assessments based on time delay and head pose	2017	Chuang et al.	logistic regression and Fast Correlation-Based Filter (FCBF)	75.6
5	An intelligent system for online exam monitoring.	2016	Prathish & Bijlani	yaw angle variations, iterations method	80

## 2.3 Cheating Detection with Deep Learning

One of the most difficult tasks is to keep an eye on the examiner's unusual conduct during the examination. Traditional monitoring programs are primarily concerned with determining the identity of testers and do not effectively detect anomalous tester behavior. Faced with the challenge of monitoring abnormal behaviors in online examinations, this study recommends using the webcam to acquire information on the examinee's head posture and mouth state, as well as to distinguish between examiners' abnormal behavior throughout the online test.

Artificial intelligence-assisted online proctoring solutions have a significant impact on the reliability and security of exam administration (Cote et al., 2016). Another person detection and electronic device detection are part of their system, which is identical to that of the study (Atoum et al., 2017a). An eye monitor was also used to assess where the participant was gazing during the examination. Dual vision cameras used to get more data from a person face (Jalali & Noorbehbahani, 2017). Yet, it is not viable to provide an eye detector or a dual sight camera to every applicant in their system. The proctoring processes used for online tests are a serious concern for the scientific community. A multi-modal approach offered for eliminating the physicality of a proctor during the test (Ketab et al., 2017). To collect audio and video, they used cameras and active window capture. This data is supplied into a sophisticated rule-based inference engine, which can identify whether any errors occurred. The examinee's face is recognized, and feature points are retrieved, enabling a head place to be approximated. Misbehavior is detected via yaw angle variations, sound presence, and dynamic window capture. A very well response element that can help the teacher maintain track of students who take an online test. As part of the system's various anticipated features, the authors included the ability to do features point collection and yaw angle detection. The project's scope is limited to a single scenario involving a single student. New video capturing tools such as Bullet Cameras and Wireless IP Cameras can also be used to create the system.

An e-cheating intelligence agent used, which is made up of two key modules: an IP detector and a behavior detector, to detect online cheating techniques (Tiong & Lee,

2021). The intelligence agent keeps a close eye on the students' actions and can spot and prevent any nefarious behavior. It can be used to create randomized multiple-choice questions for a course test and connected with online learning tools to track student behavior. The efficiency of the proposed strategy was confirmed after it was tested on several data sets. Students and educational institutions are discovering that online learning is a new and interesting option. E-learning brings new potential as well as obstacles in today's climate. Academic integrity is a critical concern in online examinations, as students may employ a range of deceptive strategies, particularly cheating, which presents a considerable obstacle. The present study aimed to investigate the effectiveness of four deep learning algorithms, namely DNN, Dense LSTM, LSTM, and RNN, in addressing the issue. The evaluation was conducted on two datasets consisting of mid-term and final-term exams. The aim of the present investigation was to evaluate the efficacy of said algorithms in identifying occurrences of academic dishonesty in the context of remote assessments. The Dense LSTM has the highest overall accuracy (95.32%). The average accuracy rate is 90%, which is high enough to warn academics to re-evaluate a questionable exam result.

The most popular exam format in an online learning setting is the online exam. Naturally, administering exams online presents a far larger barrier to maintaining academic integrity than administering exams in- person. The likelihood of cheating is greater because there is no on-site human proctor regulating the examinee. Educational institutions all around the world employ a variety of online exam proctoring technologies, which provide diverse methods to lower the likelihood of cheating. The method most frequently used by these tools is to capture the examinee's video and audio during the whole test. A method for detecting cheating that uses video analysis of a test to extract four different types of data, which are then put into a trained classification model and achieved 97% accuracy (Masud et al., 2022).

Advocated developing a durable, flexible, transparent, and ongoing e-assessment authentication procedure (Shdaifat et al., 2020). The system uses integrated biometrics, a safety layer that captures the person's eye movement, and voice recognition to detect the sound in order to track the examinee and ensure that only the authentic person is taking the exam. 3D face authentication was being researched and tested. An experiment was



conducted to evaluate the recommended platform's potential to identify any attempts at cheating. Throughout the study, participants' biometric data, eye movements, and head motions were captured using custom software. This experiment had 51 participants. The False Rejection Rate (FRR) of all valid respondents in two-dimensional and three-dimensional facial recognition modes was 0 and 0.0063, correspondingly. Facial recognition, being the most transparent multimodal biometric, combined with unique security features like eye tracking, head motions, and speech recognition, enables a robust and adaptable e-invigilation solution. A set of situations with 51 individuals proved to be successful both of detecting abuse as well as logistically in regard to the amount of data acquired and processed.

The rise of handheld devices prompted the creation of courses that were compatible with a variety of mobile phone models, brands, and platforms. To boost mobile learning credentials, greater focus should be paid to tests. The verification and identification of students before to and during the exam session is a crucial challenge with mobile exams. In the absence of a proctor, several options are available to ensure student identification, including a traditional username and password as well as biometric iris recognition to validate student identity before and during the mobile exam. Two forms of authentications are presented as part of a new model for authenticating pupils. The student uses a combination of standard login methods, such as username and password, and biometric identification, such as iris recognition, to prove his or her identity. Furthermore, to prevent impersonality, the proposed system checks for students at random during the exam, the focus is on video summary of anomalous behavior for online exam invigilation from a distance (Cote et al., 2016). The number of students that can be handled simultaneously through live remote invigilation is restricted, and manual post-exam evaluation is time consuming. The author presented a unique video content analysis method based on computer vision for the automatic production of video summaries of online tests to aid remote proctors in post-exam reviews. The proposed method uses head posture estimations and a semantically relevant two-state hidden Markov model to model normal and deviant student behavior patterns. Detected sequences of anomalous activity are used to construct video summaries. The experimental results are encouraging, demonstrating the practicality of the proposed

method, which might be easily expanded to create real-time alerts for live remote invigilation. By automatically extracting the features, deep learning (DL) solves this issue. The main deep learning (DL) techniques include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Generative Adversarial Networks (GAN).

In a different publication, Atoum et al. (2017b), proposed a CNN-based method for face anti- spoofing. It says that the recommended solution uses the fusion of two CNN streams to detect face spoofing as an illustration.

In the examination, the automated behavior analysis poses a serious issue. Examinees utilize a variety of unethical tactics to defraud the staff while invigilators and examiners are present. Examinees may make a variety of motions, including head and eye movements and the usage of various suspicious objects (Khan et al., 2022). Using a tweaked version of the Faster RCNN algorithm, the forbidden objects and suspicious head movements have been watched. At the same time, interactive Open-Pose architecture is used to determine the use-age of forbidden objects.

Academic institutions must keep fighting this on an equal level since cheaters will continue to employ new technologies to get around the laws and limitations that are now in place. With motion recognition utilizing several RGB images, a novel method of deep learning developed for spotting cheating on tests among students (El Kohli et al., 2022). They also created a system for identifying cheating in videos and generating written descriptions of the events in the videos. With 95% accuracy, the gesture recognition model can identify the cheating movements.

The proctoring of exams and teaching are two of the system's most important components. An essential component of evaluating the academic process is human proctoring, which entails keeping an eye on candidates throughout exams. Scalability in education is critically dependent on the ability to proctor exams. Such methods are time- and money-consuming, though. The framework for the detection and categorization of cheating video frames (Hussein et al., 2022). This sort of research assists in the timely identification of academic dishonesty.

To prevent and identify various sorts of cheating during exams, an automatic and yet authentic system must be developed. Several invigilation systems, including human and automated systems, have been addressed by (Malhotra & Chhabra, 2022), they include numerous ways for cheating and techniques for intercepting them. Deep learning-powered surveillance systems have shown to be more effective and precise than traditional techniques. Such as Faster RCNN, yolo etc.

A deep learning-based solution for assessing applicants' cheating activities (Wan et al., 2021). The classic object recognition method YOLO is integrated with the human postures estimation project to retrieve the location data of the candidates and identify the applicants who are suspects of cheating by studying the examinee's activity in a single image of the surveillance video. Because of the uniqueness of cheating activity, a dataset for supervised learning training is created that identifies two forms of human conduct in the examination scenario, including two kinds of cheating behaviors, peeping and passing notes.

Traditional monitoring methods primarily focus on tester identity and lack efficient detection of anomalous tester actions. Faced with the challenge of monitoring aberrant behaviors in online exams, using a camera to acquire information about the examinee's head position and mouth condition and to discern the abnormal behaviors of the examiners throughout the online test using deep learning (Hu et al., 2018).

A novel application of technology to detect malfeasance in e-exams, which is critical given the expansion of online education (Indi et al., 2021). The present solutions for such a challenge either demand entire manual effort or have several flaws that an examinee can exploit. The system employs a hybrid classifier technique, which employs two distinct classifiers. One when gaze parameters are being read correctly. If this fails for a variety of reasons, such as poor signal strength or glare from his eyeglasses, the model reverts to the default classifier, which merely reads the head posture values to categories the attention metric. In identifying the attention measure, the model attained an accuracy of 96%.

A novel clustering algorithm that is appropriate for clustering mixed data (Noorbehbahani et al., 2015). This method employs a novel distance metric that

combines supervised and unsupervised data. The proposed method for managing categorical information is based on the SOINN (Self-Organizing Incremental Neural Network) approach and employs an updated rule and a new distance metric. The proposed technique is then exhaustively assessed with various cluster evaluation metrics and datasets.

The algorithmic proposal offers an automated method to streamline online proctoring, thereby mitigating the laboriousness of its traditional equivalent (Kumaran et al., 2021). The proposed approach employs transfer learning to achieve deep learning and integrates three distinct models, specifically YOLO for detecting fraudulent objects and multiple individuals, MPGazeII for detecting abnormal gaze, and VGG16 for recognizing faces. The outcomes of every single anomaly detection algorithm are combined to determine whether the examinee has engaged in malpractice, enabling appropriate measures to be taken.

A proposed deep neural network for application in the field of higher education (Li & Liu, 2021). The study aims to identify and predict the scientific behavior of higher education students by comparing their academic levels and grades. A number of procedural stages have been suggested for the deep neural network algorithm, encompassing data initialization and pre-processing. In order to enhance the precision of predictions, two models, namely Adams and RMS prop, were employed to optimize the system's performance.

By presenting a collection of attributes generated from an online test based on the Moodle platform. Duham et al. (2022), suggested a recommendation system for assessing students' replies and detecting cheating when in an online exam using statistical approaches, similarity measurements, and clustering algorithms. According to the findings, the proposed online examination method efficiently eliminates cheating and delivers a credible online exam. Finally, offering an efficient and equitable method that preserves academic integrity, the most crucial part of education.

Many studies utilizing deep learning approaches have been undertaken on traffic sign identification and recognition (TSR). A system intends to present a system that delivers a quick, dependable, and inexpensive solution (Al Khafaji et al., 2022).

Combining the YOLOv5 network with a CNN improves detection and recognition accuracy and accelerates processing in the proposed system. The whole system is split into a sequence of steps. The image is segmented into  $N$  grids of  $S \times S$  pixel dimensions in the first stage. Each of these  $N$  grids is tasked with identifying and localizing the object within it. Each grid can only uniquely identify a single object. But, when several objects are within a single grid, the author utilized an anchor box. Second, the feature extraction phase involves bounding box prediction (having five parameters). Unlike other algorithms, the YOLO method splits the image into several cells depending on the number of objects it covers. The confidence ratings represent the model's level of confidence that the box contains an object and how well the box predicts an object. The proposed system utilizes the highest overlap divided by thresholding for all anchor boxes. This produces many bounding boxes for all images, some of which may not contain any objects, as well as the intersection of bounding boxes that share the uniform image region. Non-Maximum Suppression (NMS) is employed to address these problems. With Non-Maximal Suppression, YOLO outperformed all bounding boxes with lower probability values. Furthermore, Convolutional Neural Networks are applied to accomplish object identification tasks (CNN). After training, CNN was able to classify the traffic sign. As previously discussed, YOLOv5s detects the traffic sign. After extracting a bounded box, the traffic sign is sent to the CNN classification model, but only after scaling the image to  $50 \times 50$ , abstracting features, determining the most significant class probability, and selecting the label for the class in the CNN model. Finally, draw a box and print the label on the sign. The proposed system attained a recognition rate of 94% and a classification accuracy of 99.95% in CNN (Redmon et al., 2016). Table 2 presents some of the deep learning techniques with achieved different levels of accuracy.

Table 2: Cheating detection with Deep Learning

Sr No	Title	Year	Publisher	Algorithms	Accuracy %
1	An incremental mixed data clustering method using a new distance measure	2015	Noorbehbahani et al.	Adjusted Self Organizing Incremental Neural Network (ASOINN) algorithm	Credit data set (0.27, F-measure), Kr-vs-kp dataset (0.22 F-measure), and CMC dataset (F-measure, 0.38)
2	Smart online exam proctoring assist for cheating detection	2022	Masud et al.	LSTM	97.7
3	Detecting of malpractice in e-exams by head pose and gaze estimation	2021	Indi et al.	FSA-Net	96.04
4	E-cheating Prevention Measure	2021	Tiong & Lee	DNN, LSTM, DenseLSTM, RNN	68, 92, 95, 86
5	You Only Look Once, Unified, Real-Time Object Detection	2016	Redmon et al.	Fast R-CNN + YOLO	75
6	Recognition of Cheating Behavior in Examination Room Based on Deep Learning	2021	Wan et al.	YOLO with openpose	96.2
7	Deep learning: new approach for detecting scholar exam fraud	2022	El Kohli, et al.	3D CNN	95
8	An Automatic method for cheating detection in online exams by processing the students webcam images	2017	Jalali & Noorbehbahani	K-medoids clustering algorithm	78

Table 2: Cheating detection with Deep Learning (Continued)

Sr No	Title	Year	Publisher	Algorithms	Accuracy %
9	Student Invigilation Detection Using Deep Learning and Machine After Covid-19: A Review on Taxonomy and Future Challenges	2022	Malhotra & Chhabra	YOLO, LSTM, ResNet, Faster-RCNN	-
10	Face anti-spoofing using patch and depth-based CNNs	2017b	Atoum et al.	CNN and holistic depth maps	EER= 2.67 and HTER =2.27 on CASIA FASD dataset, EER= 0.79 and HTER =0.72 on Replay-Attack dataset, EER = 0.35 and HTER=0.21 on MSU-USSA
11	Performance Prediction for Higher Education Students Using Deep Learning	2021	Li & Liu	Adams and RMSprop	78.5

In this section we have discussed several traditional machine learning techniques and deep learning techniques. The fundamental difference between machine learning and deep learning is the performance. In case of smaller dataset, Machine learning work better than Deep Learning methods. This is the main reason why deep learning algorithms require a vast quantity of data to fully comprehend it by extracting the features. In addition to this, Deep Learning relies on high-end equipment, whereas machine learning algorithms can also work even on low-end devices. High computation GPUs are required for Deep Learning (Krizhevsky et al., 2012). That is an important aspect of how it works. It also performs a significant amount of matrix multiplication. It's a standard procedure. Domain knowledge is used to create feature extractors in order to minimize data complexity and make patterns more obvious so that the algorithm may be learned. It is, nevertheless, quite difficult to comprehend. As a result, it takes time and requires experience (Deng, 2014).

Our work is different from previous works because we experiment with combination of several techniques including pr-processing, detection, and classification to find best combination model that will prove which model is more accurate. In addition, our work is more related to Atoum et al. (2017a) while our solution is more practical. They are using wearcam videos additional to the webcam, but we are using webcam videos only which is more suitable for the online exams and more user-friendly.

However, papers mentioned on this section has some limitations. The present system was not implemented on web-based tracking systems for e-cheating. DNNs, LSTMs, DenseLSTMs, and RNNs. On the other hand, it can be computationally costly to train and run, particularly when faced with large datasets or complex frameworks (Tiong & Lee, 2021).

The proposed system lacks additional examinee behavior data and requires additional performance enhancements. Additionally, the system is intended for use with input images of fixed size, which can be an issue for applications that require input images of variable size (Wan et al., 2021).

Moreover, the current system has several limitations, such as requiring students to use a different web browser for accessing the examination applications and requiring the instructor to change the questions each semester using a randomized approach (El Kohli et al., 2022). On the other hand, 3D CNNs demand a substantial amount of training data, which may not always be accessible or feasible to obtain.

Also, the proposed system involved multiple cameras to record unique perspectives from every angle. In addition, the method does not scale well with huge datasets and may not be appropriate for real-time applications or online learning (Jalali & Noorbehbahani, 2017).



## Chapter 3: Research Methodology

In this section, we discussed our proposed methods to solve research question 2. This approach performs well in terms of accuracy and time complexity. Five experiments were conducted, by using same custom dataset for all experiments.

- Faster RCNN and Bi-GRU.
- SSD MobileNet and Bi-GRU.
- Faster RCNN and CNN-LSTM.
- YOLOv5 and CNN (existing technique).
- YOLOv5 and Bi-GRU.

From each image we will get the key images which shows not cheating and cheating such as:

Mobile phone showing in the video: mobile phones with increased network coverage enable constant connectivity, quick information flow, and accessibility. In mobile phone technology, there are various ways to cheat from them, such as reading saved class materials, text messaging classmates or outside help, browsing the Internet, and taking a snapshot of the exam to share with other examiners.

Multiple faces showing in the video: one of the biggest concerns in online exams is that the examiner take assistance from another person during online-exam. The examiner is also expected to take the exam alone without any assist of another person in the room.

Behavior of reading books and notes to determine where the student was looking during the exam: when the student's face looking down from the screen for a long period of time (face turns left, right, or down).

For this reason, these three things are the most crucial for identifying online exam fraud. All exams are based on these three variables because students can cheat by using their phones to look up answers or by making eye contact with an examiner to see whether they are being observed, and because seeing numerous faces can indicate that

students are working together to answer questions. Students can't use any exam cheating methods if these three things are caught.

In this case, we used different combinations to find best combination from all the experiments.

### **3.1 Faster RCNN and Bi-GRU**

In Faster RCNN, the architecture comprises two main modules: a region proposal network (RPN) and a detection network, as Figure 6 shows.

The RPN generates candidate object regions by scanning the image with a sliding window of different scales and aspect ratios. For each window, the RPN produces a set of bounding box proposals and a corresponding objectness score that indicates the likelihood of the region containing an object.

The detection network takes the proposed regions from the RPN and performs further classification and refinement to produce the final set of detections. The network applies a set of convolutional layers and fully connected layers to extract features from the proposed regions. These features are then fed into two sibling output layers: one for object classification and another for bounding box regression.

The object classification layer assigns a class label to each proposed region, indicating whether it contains a face or not. The bounding box regression layer refines the coordinates of the proposed regions to improve their accuracy.

The output of Faster-RCNN will be the input to Bi-GRU classification model.

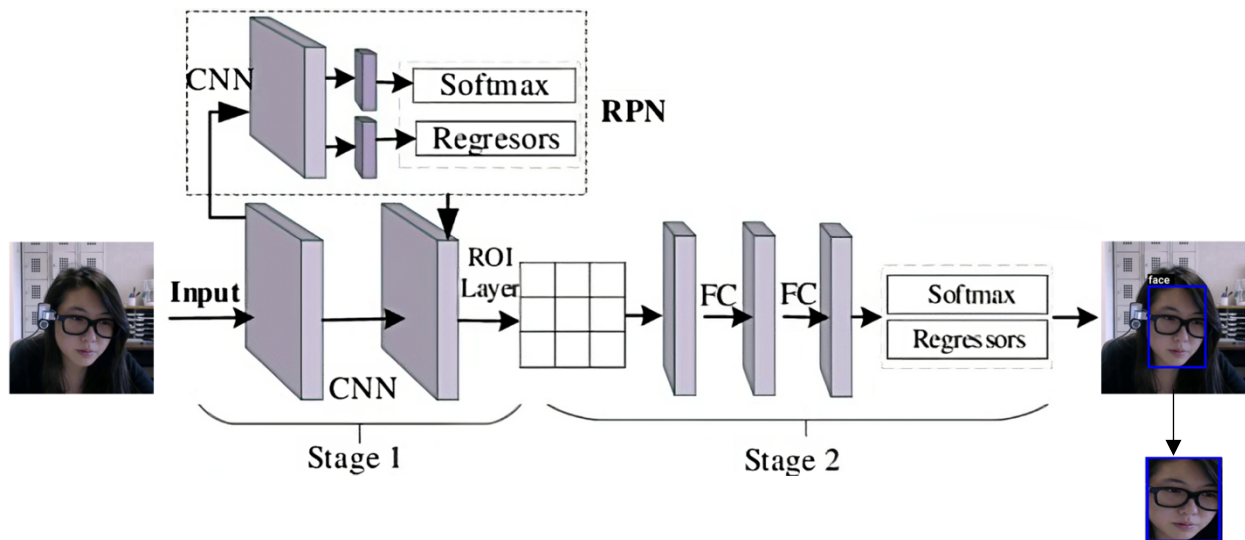


Figure 6: Face detection using Faster RCNN

BiGRU used for face classification. This model has a total of 14,078 parameters, out of which 14,014 are trainable and 64 are non-trainable. Bidirectional Gated Recurrent Unit model is a type of recurrent neural network (RNN) that uses gates to selectively remember or forget information from previous time steps; as Figure 7 shows, the input is the face portion and the output is the classification of cheating/not cheating. The bidirectional aspect of the model allows it to learn from both past and future context, which makes it more effective in capturing long-term dependencies in sequences of data.

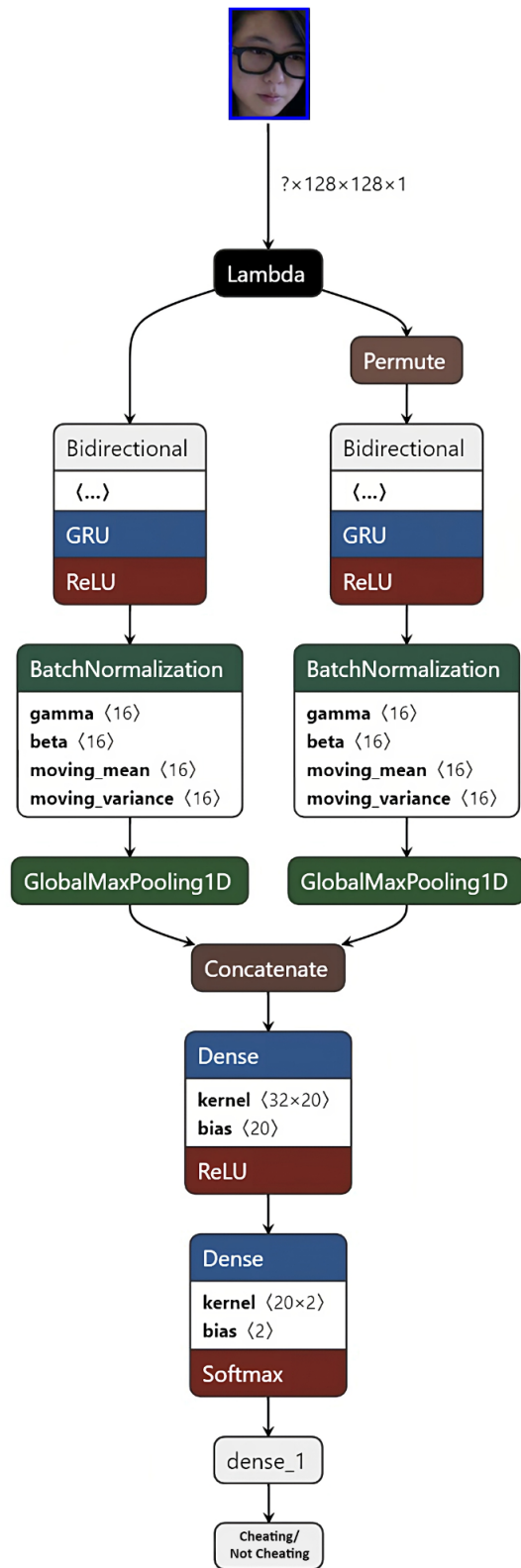


Figure 7: Face Classification using Bi-GRU

To conclude, Faster R-CNN helping in detecting the face appeared, Bi-GRU help to classify those faces to cheating and not cheating. We implement Faster R-CNN and Bi-GRU to do all the cheating detection mentioned in Chapter 4.

### **3.2 SSD MobileNet and Bi-GRU**

We proposed other technique such as SSD MobileNet for face detection and Bi-GRU for face classification. SSD The Single Shot Detector (SSD) and MobileNet models are combined in the MobileNet object detection architecture. The architecture is designed to be fast and efficient, making it suitable for real-time applications on mobile and embedded devices.

The SSD MobileNet architecture as Figure 8 shows, consists of a base MobileNet network followed by a set of convolutional feature maps at different scales. These feature maps are then fed into a set of detection heads, which predict the locations and classes of objects in the input image.

The MobileNet base network uses depth-wise separable convolutions to reduce the number of parameters and computation required compared to traditional convolutional networks. The feature maps produced by the base network are used to detect objects at different scales, with smaller objects being detected at higher resolution feature maps and larger objects at lower resolution feature maps.

The detection heads predict object locations and classes using a set of convolutional layers and anchor boxes. The anchor boxes are predefined bounding boxes at different aspect ratios and scales that are used to predict object locations and sizes. The detection heads output a set of confidence scores and offsets for each anchor box, which are used to refine the predicted object locations and filter out false detections.

Overall, the SSD MobileNet architecture is a powerful and efficient object detection model that can achieve high accuracy on a variety of datasets while running in real-time on mobile and embedded devices. So, the output of SSD MobileNet will be the input to Bi-GRU classification model.

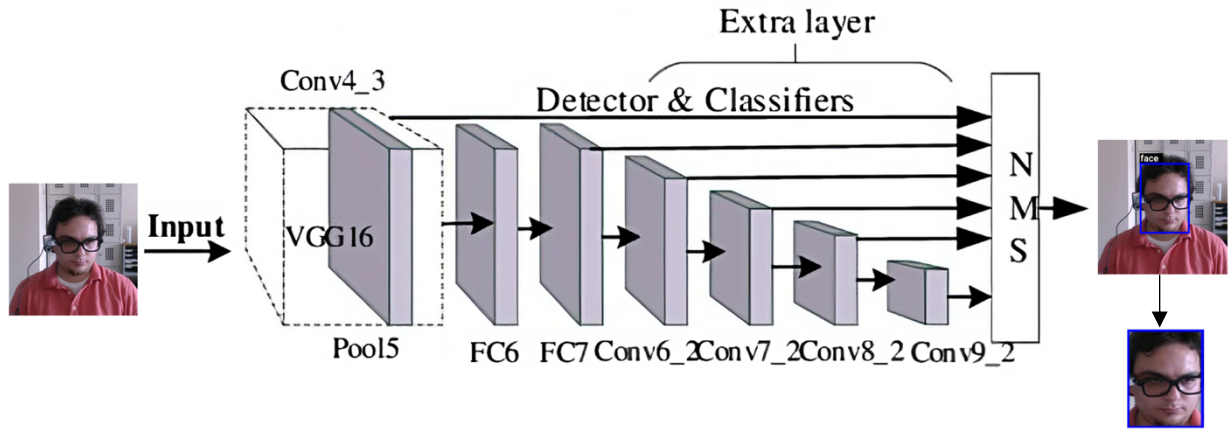


Figure 8: Face Detection using SSD MobileNet

SSD MobileNet can be used for face detection by training it on a dataset of annotated face images. The architecture can be modified to predict the locations and sizes of faces in input images using anchor boxes that are specifically designed for detecting faces. In Figure 7, Bi-GRU process applied for the classification to classify cheating/not cheating.

### 3.3 Faster RCNN and CNN-LSTM

We proposed Faster RCNN to do all the face detection and CNN-LSTM for face classification. In Figure 8, same Faster-RCNN process has been applied to do all the detection, then it will be sent to do all the face classification. The output of Faster-RCNN will be the input to CNN-LSTM classification model.

CNN-LSTM is a deep learning architecture used for face classification tasks. Figure 9 display the architecture of CNN-LSTM, a convolutional neural network (CNN) and a long short-term memory (LSTM) network are its two fundamental components. The CNN extracts feature from the input face images, while the LSTM processes these features over time to capture temporal dependencies. In addition, the output of the model is to classify cheating/not cheating.

The model is trained on a large dataset of labelled face images, and the learned features can be used to classify new faces. The total number of parameters in the model is 1,980,954, and all of them are trainable.

The CNN-LSTM architecture has several hyperparameters that can be tuned to optimize performance, including the number of CNN and LSTM layers, the size of the convolutional filters, and the number of neurons in the LSTM layers.

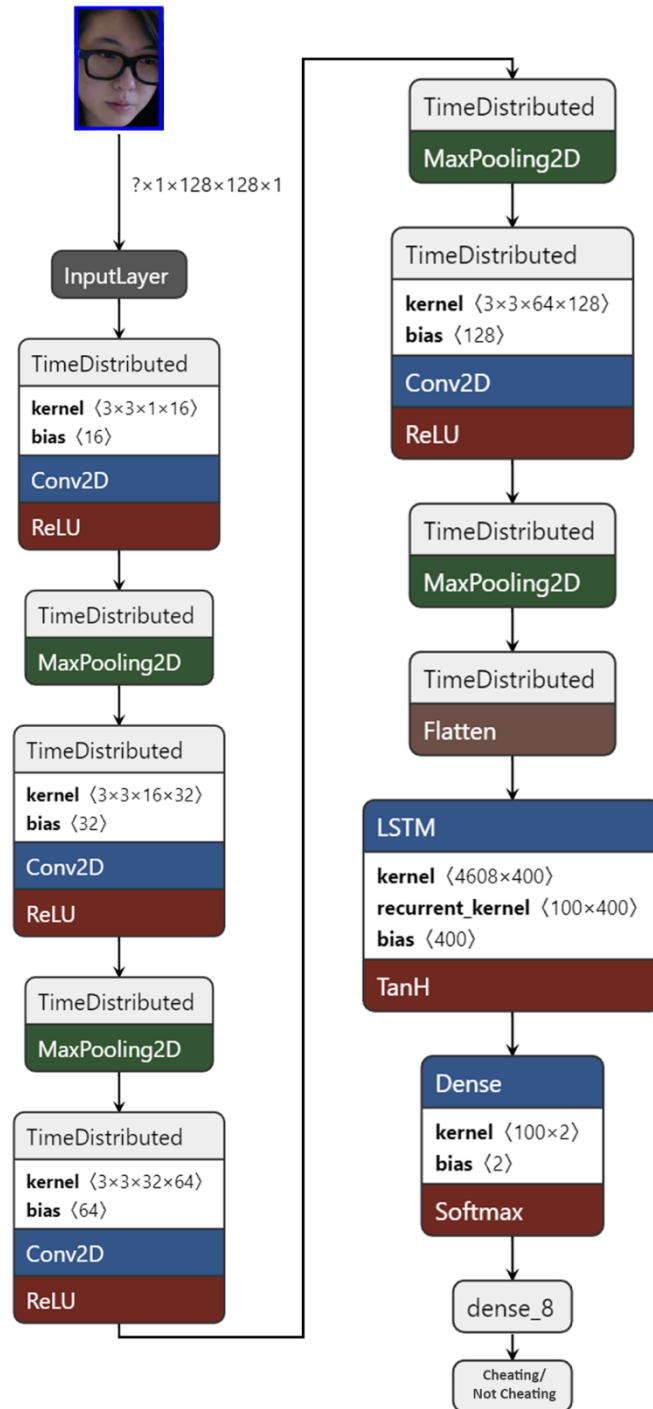


Figure 9: Face Classification using CNN-LSTM

### 3.4 YOLOv5 and CNN

This technique combined of YOLOv5 and CNN. The technique of YOLOv5 which will do all the face detection and CNN for the face classification. YOLOv5 is a state-of-the-art object detection model that can be used for face detection tasks. It is a variant of the YOLO (You Only Look Once) family of object detection models that uses a single neural network to predict bounding boxes and class probabilities for objects in an input image. YOLOv5 improves upon its predecessors by introducing several architectural improvements and training techniques, resulting in improved accuracy and speed.

The model uses anchor boxes to predict object locations and applies a combination of objectness and classification losses during training. YOLOv5 can be fine-tuned for face detection by training it on face-specific datasets and has been shown to perform well on a variety of face detection benchmarks.

The YOLOv5 architecture as Figure 10 shows, is a deep learning model used for object detection tasks, including face detection. It is a one-stage detector that uses a convolutional neural network (CNN) to directly predict the bounding boxes and class probabilities for all objects in an input image. Moreover, YOLOv5 uses a smaller network architecture than previous versions (such as YOLOv4), with fewer parameters and faster inference times, while still maintaining high accuracy. It also includes various optimizations such as weight pruning and advanced augmentation techniques to improve the model's performance. YOLOv5 can be fine-tuned on face detection tasks with labelled face datasets, making it a useful tool for face recognition and other related tasks.

The output of YOLOv5 will be the input to CNN classification model.



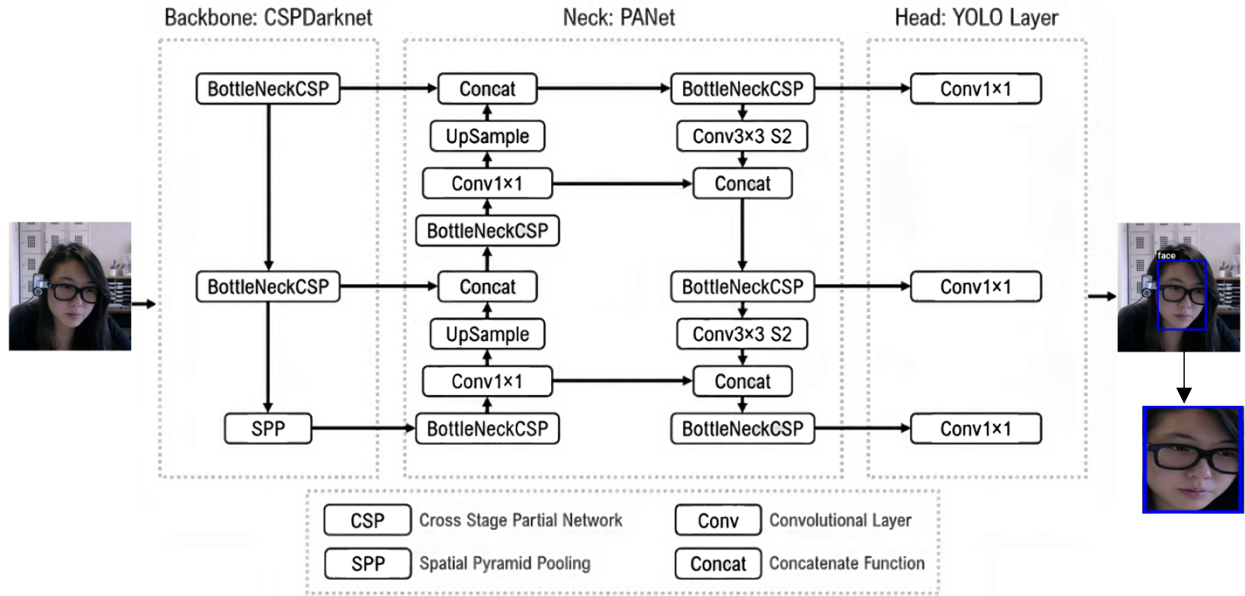


Figure 10: Face Detection using YOLOv5

A Convolutional Neural Network (CNN) architecture as Figure 11 display is a type of deep learning model used for image classification tasks. It is made up of several layers, including as convolutional layers, pooling layers, and fully connected layers. The convolutional layers perform feature extraction by applying filters to the input image, which capture important patterns and features. The pooling layers reduce the dimensionality of the feature maps, making the model more computationally efficient. The fully connected layers combine the features to make a prediction about the input image's class. The final layer typically uses a softmax activation function to output the predicted class probabilities. The architecture can be customized by varying the number and size of the layers and filters, among other parameters, to optimize performance for a given dataset. The output of CNN is to classify the input image to cheating/not cheating.

The CNN model for face classification has a total of 1,849,509 parameters, out of which 1,849,282 are trainable and 227 are non-trainable. The model is designed to classify images into two categories - cheating and not cheating. The model uses convolutional layers to extract features from the input image and then flattens the output to feed it into fully connected layers for classification. The number of filters and size of the kernels used in the convolutional layers vary across different layers. The model has

been trained on a labelled dataset to achieve high accuracy in classifying cheating and non-cheating images.

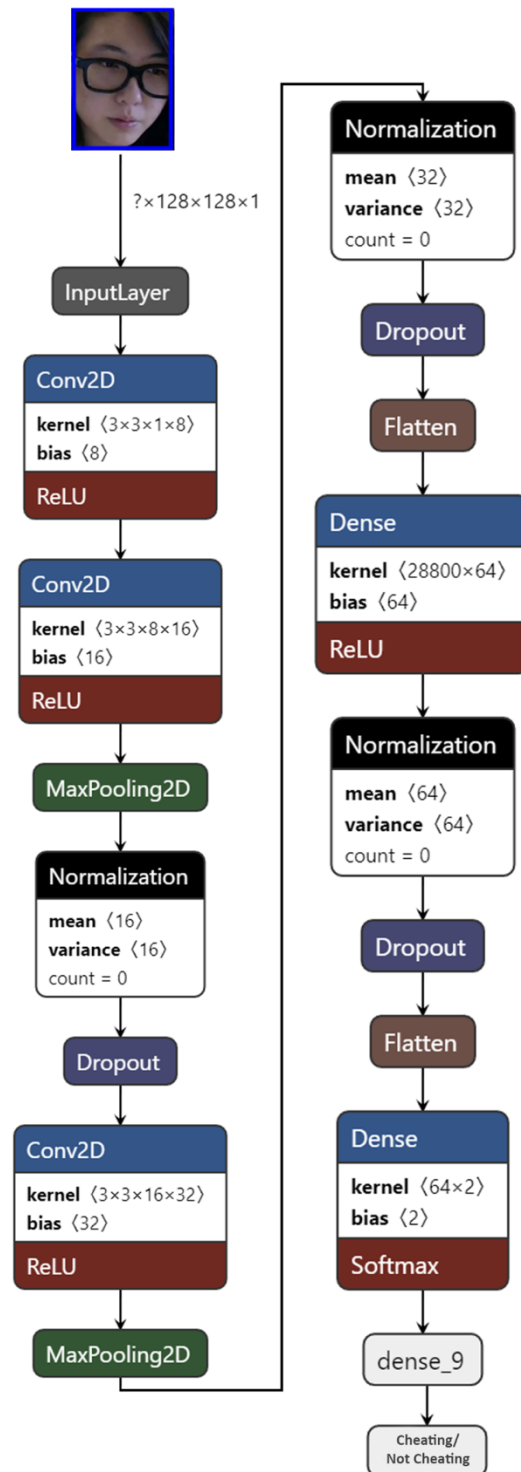


Figure 11: Face Classification using CNN

### **3.5 YOLOv5 and Bi-GRU**

A proposed technique that combined YOLOv5 and Bi-GRU. YOLOv5 did all the object detection while Bi-GRU did classification part (cheating or not cheating). Same process has been applied such as in Figure 7 and Figure 10.

## Chapter 4: Dataset

In this study, we have used a public dataset. A total of 24 subjects, all of whom are students at Michigan State University, involved in the data collection. An online test in mathematics was prepared with two types of questions, namely, multiple choices and fill in the blank. Link of the dataset <http://cvlab.cse.msu.edu/project-OEP.html> .

Before the exam, the examiner explains the following rules to the examinees: (a) No books, notes, or texts are allowed in the room. (b) Phones and laptops are banned. (c) The examinee must solve the exam alone. (d) No Internet uses. Moreover, 15 actors pretended to take the test, but nine students took a real exam and had their scores recorded. During the exam session, the actors were asked to cheat without being told what or how to cheat. However, these subjects may act artificially, which may not reflect cheating in real exam scenario. On the other hand, the nine students who took real exams, know that they can't cheat in the room. The proctor tests them into committing cheating by talking, approaching, or handing them a book. The combination of these two types of subjects (actors, and real students) adds cheating strategies and exam involvement to the database.

For each of 24 subjects, in the original dataset they capture audio and two videos from both cameras (wearcam and webcam), but we only used webcam videos. Each student video averaged 17 minutes long. Human annotation and tagging is done offline after viewing both videos and audio simultaneously. The tagging shows start time, end time, and type of cheating are labelled for each cheating incident. 5 types of cheating are identified: (1) cheating from a book or notes. (2) chatting with another person in the room. (3) using the Internet. (4) asking someone a question over the phone. (5) using mobile phone. Nearly 20% of the video shows cheating, whereas 80% shows typical exam taking behavior (Atoum et al., 2017a).

We perform several pre-processing for data to make it useful for our detection technique by using webcam videos only. We segmented these videos into small video by using the webcam located above the monitor and recognize their actions.

Furthermore, we did the segmentation of the videos in different way from the original dataset, the segmentation process done by 3 types of cheating. The purpose of this process, we didn't use the wearcam videos from the original dataset, therefore so we can't identify one type of cheating (using internet), and we combine using phone and asking someone a question over the phone in one category. Instead, we did for 3 types of cheating such as (1) cheating from a book or note. (2) chatting with another person in the room or if any other face detected than examiner. (3) using phone.

Video is sequence of images; we extracted frame with 2 frame per second interval by using python script (OpenCV library). Image annotation is one of the crucial parts in computer vision for detecting objects. As Figure 12 shows, we make the dataset by using Labellmg annotation tool. Labellmg is an open-source graphical image annotation tool that allows users to label images for object detection tasks. Besides, it allows users to easily create bounding boxes, polygons, and other shapes around objects in an image. Labellmg is particularly useful to quickly create large datasets with labeled images, which can then be used to train object detection models (Patel & Patel, 2020). Additionally, this tool saves a .xml files which contain the label data for each image, and in each image generated will be one .xml file. Approximately 23,000 images annotated to cheating and not cheating. After preparing the dataset of xml files, we use different object detection models which extract features from this annotated area and learn by itself. Then, different classification models would classify if cheating occurred or not.

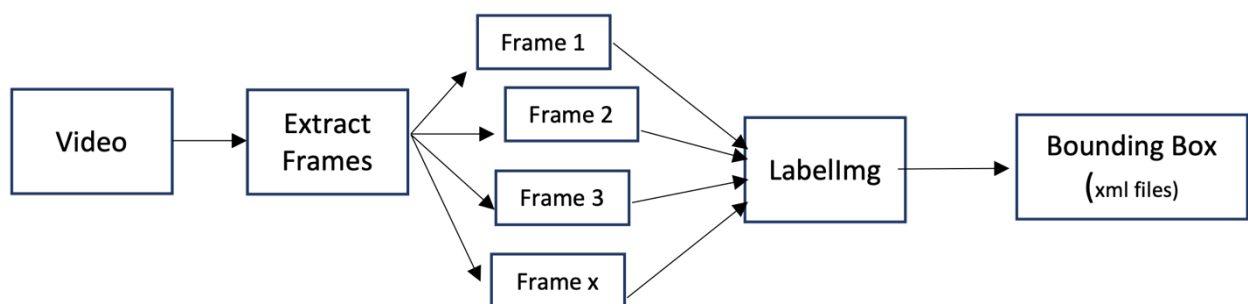


Figure 12: Video extraction

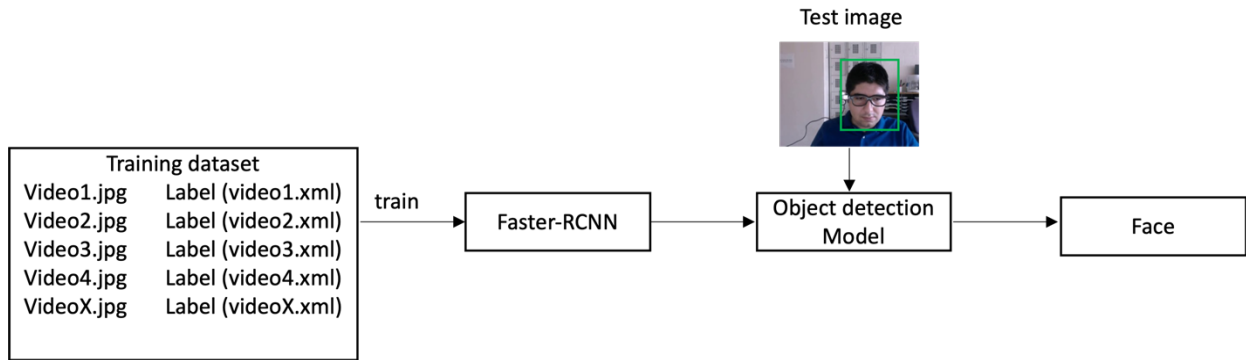


Figure 13: Input image to the object detection model

In Figure 13, after preparing the training dataset. Our training data for each frame of the video; we have corresponding label. Then model trained with face detection models (Faster-RCNN, SSD MobileNet, YOLOv5) and it can detect faces in any test image by drawing the bounding box around the object.



Figure 14: Object detection and classification process

Figure 14 shows the combination of two models (detection and classification). The process of an image which is input to the object detection model to detect any face appeared and it cropped the face portion and send it to face classification model (Bi-GRU, CNN-LSTM, CNN) to classify if cheating occurred or not. The power of our combined model, it can be tested with different dataset to detect cheating.

Table 3 shows the dataset information, each student number and length of the original video. After segmentation part the videos split into small videos into two parts (cheating and not cheating), there is no direct relation between the segments and the video time because we only used images, doesn't matter the video length. This table illustrate number of segmented videos that indicates cheating and not cheating. And if

the student acting during the exam or real student doing the exam. The segmentation process will be helpful to train classification models.

Table 3: Dataset Information

Subjects	Number of Segmented Videos of cheating	Number of Segmented Videos of not cheating	Total Video Length	Acting/real exam
1	10	12	00:14:24	Acting exam
2	5	15	00:16:00	Acting exam
3	9	15	00:15:20	Acting exam
4	7	15	00:14:39	Acting exam
5	18	15	00:14:16	Acting exam
6	10	19	00:18:18	Acting exam
7	14	18	00:15:43	Acting exam
8	5	16	00:18:34	Acting exam
9	9	15	00:17:40	Acting exam
10	17	21	00:25:37	Real exam
11	13	8	00:17:34	Real exam
12	22	10	00:22:45	Real exam
13	27	6	00:24:17	Real exam
14	12	6	00:12:16	Real exam
15	20	12	00:19:21	Real exam
16	21	6	00:15:42	Real exam
17	15	10	00:15:45	Acting exam
18	33	7	00:25:07	Real exam
19	21	2	00:19:03	Real exam
20	10	12	00:15:28	Acting exam
21	23	22	00:15:13	Acting exam
22	16	17	00:13:46	Acting exam
23	19	16	00:17:43	Acting exam
24	5	14	00:14:20	Acting exam

In Table 4, it shows video segmentation for each student separately (24 students). It represents three types of cheating (chatting with another person in the room/another face detected, using phone, cheating from a book or notes). It shows total of each type of cheating for all students and total of cheating per subject.

Table 4: Types of Cheating

<b>Types of Cheating</b>	<b>Cheating from Book or Notes</b>	<b>Chatting with Another Person in the Room/Another Face Detected</b>	<b>Using Mobile Phone</b>	<b>Total of Cheating per Subject</b>
Subject 1	6	1	3	10
Subject 2	3	1	1	5
Subject 3	5	1	3	9
Subject 4	6	1	-	7
Subject 5	13	1	4	18
Subject 6	5	1	4	10
Subject 7	5	-	9	14
Subject 8	4	1	-	5
Subject 9	6	1	2	9
Subject 10	13	4	-	17
Subject 11	10	3	-	13
Subject 12	16	6	-	22
Subject 13	10	16	1	27
Subject 14	7	5	-	12
Subject 15	13	7	-	20
Subject 16	13	8	-	21
Subject 17	2	13	-	15
Subject 18	10	23	-	33
Subject 19	2	19	-	21
Subject 20	7	3	-	10
Subject 21	5	7	11	23
Subject 22	3	5	8	16
Subject 23	10	2	7	19
Subject 24	4	1	-	5
Total of each cheating type for all subjects	178	130	53	361



## Chapter 5: Experiments and Results

### 5.1 Experimental Setup

#### 5.1.1 Dataset

In this thesis, five experiments were conducted. The splits are made with kfold split during kfold cross validation which it equals to 5-fold. 23,149 images are used for face classification where 12,328 images are labelled as “cheating” & 10,821 images are labelled as “not cheating”. Experiments used are Faster RCNN with Bi-GRU classification, SSD MobileNet with Bi-GRU, Faster RCNN with CNN-LSTM, YOLOv5 with CNN classification and YOLOv5 with Bi-GRU. Same dataset is used for all of the experiments.

#### 5.1.2 Software

To implement the presented model, we utilized Python 3.8, TensorFlow 2.6.0, PyTorch 1.13.1+cu116 and other dependencies related to deep learning. We emphasize the significance of dividing the dataset into separate training and validation sets. The training set is used to train the network, whereas the validation set is utilized to validate the data that the network has not previously encountered.

#### 5.1.3 Evaluation Setup

Data splitting can be accomplished through two main criteria, namely normal split and cross-validation. However, the cross-validation criteria are often preferred, where the dataset is split into a predetermined number of folds, and each fold is used in turn as the testing set while the rest act as the training set. Cross-validation is a perfect technique to prevent overfitting situations. In our experiment, we utilized a 5-fold cross-validation strategy to divide our data into training and testing sets.

To measure the culmination of the model, the ROC Curve is often utilized. It provides an idea of the true-positive rate for a given false-positive rate, which is a good summary indicator of the classifier's attainment. The following performance metrics can be evaluated:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Accuracy = \frac{Number\ of\ instances\ correctly\ classified}{Number\ of\ instances}$$

Some metrics are commonly used to evaluate the performance of object detection models such as precision and recall. Overall, these metrics provide a comprehensive evaluation of the object detection model's performance and can help identify areas for improvement in the model's accuracy and recall.

#### 5.1.4 Parameter Setting

We apply same parameter setting to have fair comparisons for all the detection and classification models, to see which model is performed better than other.

For Faster RCNN, the image size is 640x640 px, batch size is 8, number of steps is 2500 with 5-fold cross validation, number of evaluation steps is 1000, optimizer is momentum\_optimizer, score converter is SOFTMAX, cosine decay learning rate is used where learning\_rate\_base is 0.04 & warmup\_learning\_rate is 0.013333.

For SSD MobileNet, the image size is 640x640 px, batch size is 8, number of steps is 2500 with 5-fold cross validation, number of evaluation steps is 1000, optimizer is adam\_optimizer, initial learning rate is 0.001.

For YOLOv5, the image size is 640x640 px, batch size is 8, epochs is 2 with 5-fold cross validation as it equals to 2500 steps in TensorFlow Object Detection, initial learning rate is 0.01, optimizer is 'SGD', optimizer weight decay is 0.0005, model depth multiple is 0.33, layer channel multiple is 0.25.

$$Total\ number\ of\ training\ steps = \frac{Total\ number\ of\ images}{Batch\ size} * Number\ of\ epochs$$

For BiGRU, CNN-LSTM & CNN, the image size is 128x128 px, batch size is 8, activation is 'softmax', optimizer is 'adam', learning\_rate is 0.001, loss is 'Binary\_Crossentropy'. Number of epochs is 5 with 5-fold cross validation.

## 5.2 Object Detection Techniques

This part is divided into 3 models: Faster RCNN, SSD MobileNet and YOLOv5. All used for object detection (face). 5-fold cross validation has been applied for all models.

### 5.2.1 *Faster RCNN*

The Faster RCNN architecture has shown high accuracy and efficiency in face detection tasks. In our experiments, we achieved promising results using this architecture with a small dataset. Faster RCNN utilizes different types of losses to optimize the region proposal network and the detection network. The RPN loss, consisting of classification loss and bounding box regression loss, is computed for the region proposal network, while the detection loss, also comprising classification loss and bounding box regression loss, is computed for the detection network. The total loss is the sum of the RPN and detection losses. The aim of optimizing the losses is to maximize the detection accuracy of the network by minimizing the difference between the predicted and ground-truth values.

When training a Faster R-CNN model for object detection using the TensorFlow Object Detection API, there are several different types of loss curves that can be generated to monitor the training process. Here are some common examples:

**Classification Loss Curve:** This curve shows the loss due to incorrect classification of objects in the images.

**Localization Loss Curve:** This curve shows the loss due to inaccurate localization of objects in the images.

**RPN Localization Loss Curve:** This curve shows the loss due to inaccurate localization of region proposals generated by the RPN.

RPN Objectness Loss Curve: This curve shows the loss due to incorrect classification of region proposals generated by the Region Proposal Network (RPN).

Total Loss Curve: This curve shows the overall loss during training, which is the sum of the classification loss, localization loss, and regularization loss.

Overall, these loss curves can be used to monitor the training process and identify areas for improvement in the model's performance. By analysing the different components of the loss function separately, it is possible to determine which aspects of the model need to be improved to achieve better results. x-axis represent number of steps and y-axis is the loss. Total loss of Faster RCNN is shown below in the Figure 15 of the training results including classification loss curve, localization, RPN localization and RPN localization loss. The loss is decreasing which we indicate that our training performs very well.

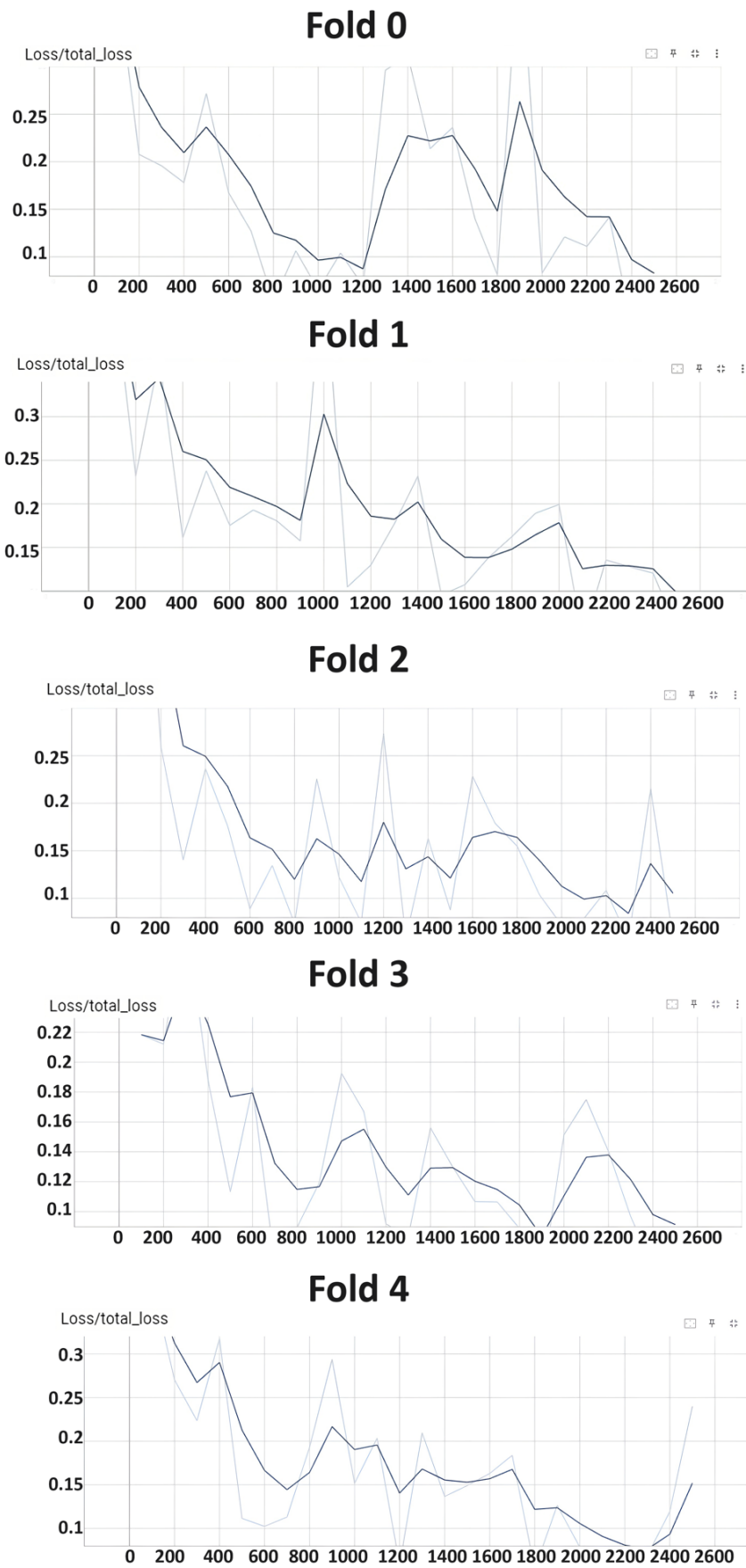


Figure 15: Faster RCNN loss curves for all folds

Table 5 shows the precision and recall values for different folds in a validation accuracy chart. The results indicate that the model achieved high precision and recall values, with an average precision 0.99 and an average recall 0.891. Moreover, the total loss values for all folds were relatively low, ranging from 0.067571 to 0.076963. Based on the overall accuracy, fold 0 & 1 both give the overall best results among 5 folds. The overall average accuracy of Faster-RCNN for all folds is 0.938.

Table 5: Faster RCNN Results

<b>Fold No.</b>	<b>Precision</b>	<b>Recall</b>	<b>Total Loss</b>	<b>Overall Accuracy</b>
0	0.990	0.895	0.067571	0.940
1	0.990	0.894	0.076963	0.940
2	0.989	0.892	0.073690	0.938
3	0.990	0.887	0.076743	0.936
4	0.990	0.887	0.074346	0.936
Overall Average	0.9898	0.891	0.0738626	0.938

### 5.2.2 SSD MobileNet

When training a Tensorflow Object Detection model with SSD MobileNet architecture for object detection, there are several different types of loss curves that can be generated to monitor the training process. Here are some common examples:

**Classification Loss Curve:** This curve shows the loss due to incorrect classification of objects in the images.

**Localization Loss Curve:** This curve shows the loss due to inaccurate localization of objects in the images.

**Regularization Loss Curve:** This curve shows the loss due to weight decay, which is a technique used to prevent overfitting by adding a penalty term to the loss function.

Total Loss Curve: This curve shows the overall loss during training, which is the sum of the classification loss and the localization loss.

Overall, all the loss curves are shown in Figure 16 summarized in total loss. Total loss during training is combined of the losses of classification, localization and regularization. By analysing the different components of the loss function separately, it is possible to determine which aspects of the model need to be improved in order to achieve better results X-axis represent number of steps and y-axis is the loss. The total loss is fluctuated our training perform well when the loss decrease.

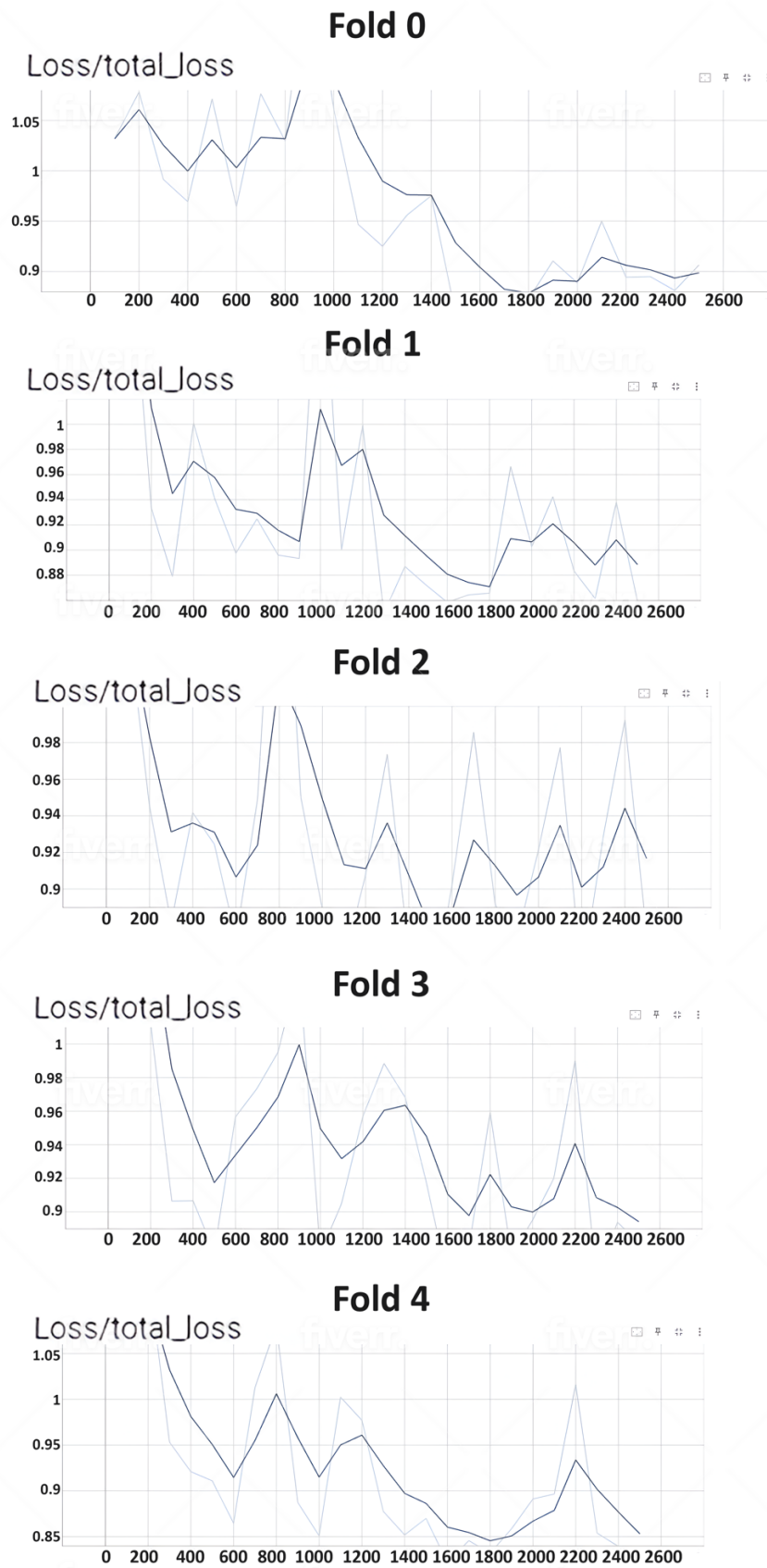


Figure 16: SSD MobileNet loss curves for all folds



Table 6 shows the performance metrics for the SSD MobileNet model on five different folds. The model's precision and recall are evaluated. The model's precision and recall are evaluated. The total loss is also reported for each fold. Overall, the SSD MobileNet model achieves relatively high precision and recall scores across all folds, with precision ranging from 0.977 to 0.989, and recall ranging from 0.791 to 0.827. The model has a total loss that ranges from 0.822270 to 0.950652, which indicates that the model is effectively learning the task of face detection. Fold 1 gives the best results among all the 5 folds based on overall accuracy. The overall average accuracy of SSD MobileNet for all folds is 0.886.

Table 6: SSD MobileNet Results

<b>Fold No.</b>	<b>Precision</b>	<b>Recall</b>	<b>Total Loss</b>	<b>Overall Accuracy</b>
0	0.988	0.801	0.899853	0.885
1	0.986	0.827	0.822270	0.900
2	0.980	0.801	0.917941	0.882
3	0.977	0.791	0.950652	0.874
4	0.989	0.807	0.896923	0.889
Overall Average	0.984	0.8054	0.8975278	0.886

### 5.2.3 YOLOv5

YOLOv5 uses a combination of different losses to optimize the network during training. The primary loss function used is the binary cross-entropy loss, which measures the difference between the predicted class probabilities and the true class labels for each object. This loss is used to classify each object in the image as a face or a non-face.

Additionally, YOLOv5 also employs two regression loss terms: the smooth L1 loss and the generalized IoU (GIOU) loss. The smooth L1 loss is used to calculate the difference between the predicted and ground-truth bounding box coordinates for each

object. The GIoU loss is a more generalized version of the IoU loss, which is used to calculate the overlap between the predicted and ground-truth bounding boxes.

Finally, YOLOv5 uses a focal loss, which assigns higher weights to hard examples (objects that are difficult to detect) during training. This helps to address the problem of class imbalance in the dataset, where there are fewer face objects than non-face objects in the training set. Overall, the combination of these loss functions helps to optimize the network for face detection tasks, improving its accuracy and robustness.

When training a YOLOv5 model for object detection, there are several performance metrics and loss components that are commonly used to evaluate the model's performance. Here is an overview of these metrics:

**Box Loss:** This is a component of the total loss that measures the accuracy of the predicted bounding boxes. The box loss penalizes the model for incorrect predictions of the box's center point, width, and height, as Figure 17.

**Objectness Loss (Obj Loss):** This is a component of the total loss that measures the confidence score of the predicted objectness. The objectness loss penalizes the model for incorrect predictions of the presence or absence of objects in the image.

**Classification Loss (Class Loss):** This is a component of the total loss that measures the accuracy of the predicted object classes. The classification loss penalizes the model for incorrect predictions of the object classes.

**Precision:** Precision measures the proportion of true positive detections (correctly identified objects) among all detected objects. A high precision indicates that most of the detected objects are actual objects.

**Recall:** Recall measures the proportion of true positive detections among all actual objects in the image. A high recall indicates that the model can detect most of the objects present in the image.

In Figure 17, these performance metrics and loss components can be used to monitor the training process and evaluate the YOLOv5 model's performance. x-axis

represent number of epochs and y-axis is the loss. The loss is dropped, and we conclude that the model is trained well.

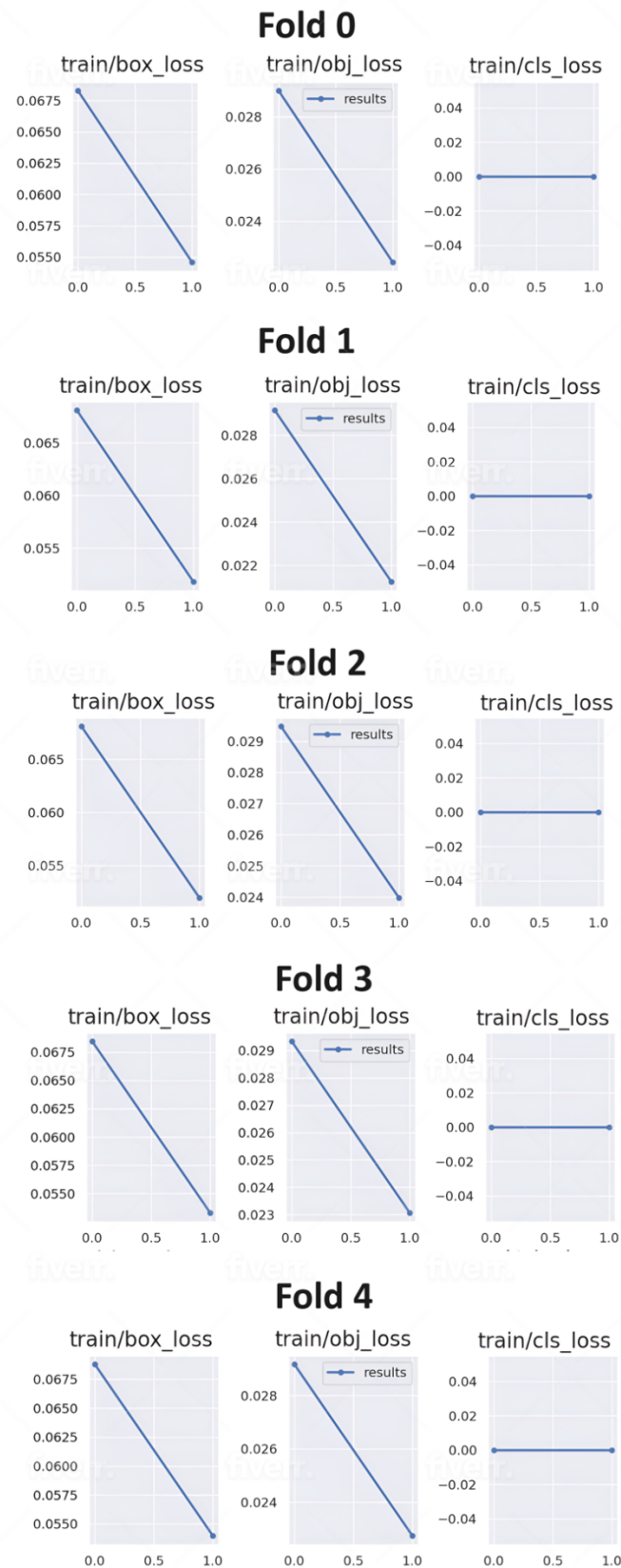


Figure 17: YOLOv5 loss curves for all folds

The YOLOv5 face detection model was evaluated using precision, recall metrics for different folds of the dataset. Table 7 presents the results show that the model achieved high precision and recall scores across all folds. Fold 1 gives the best result among all of 5 folds based on the overall accuracy. The overall average accuracy of YOLOv5 for all folds is 0.927.

Table 7: YOLOv5 Results

<b>Fold No.</b>	<b>Precision</b>	<b>Recall</b>	<b>Overall Accuracy</b>
0	0.948	0.917	0.932
1	0.975	0.928	0.951
2	0.958	0.873	0.914
3	0.956	0.877	0.915
4	0.948	0.903	0.925
Overall Average	0.957	0.8996	0.927

Table 8 shows summary of all the results of object detection models (Faster-RCNN, SSD MobileNet and YOLOv5). For our dataset, Faster-RCNN works better in terms of accuracy with 93.8%, and YOLOv5 accuracy is perform well with 92.8%. In addition, Faster-RCNN performs well in terms of precision for each fold and YOLOv5 works the best among all models in terms of recall. SSD MobileNet gave us lowest accuracy compared to Faster-RCNN and YOLOv5.

Table 8: Summary of detection models

<b>Fold</b>	<b>Technique</b>	<b>Precision</b>	<b>Recall</b>	<b>Overall Accuracy</b>
Fold 0	Faster-RCNN	0.990	0.895	0.940
	SSD MobileNet	0.988	0.801	0.885
	YOLOv5	0.948	0.917	0.932
Fold 1	Faster-RCNN	0.990	0.894	0.940
	SSD MobileNet	0.986	0.827	0.900
	YOLOv5	0.975	0.928	0.951
Fold 2	Faster-RCNN	0.989	0.892	0.938
	SSD MobileNet	0.980	0.801	0.882
	YOLOv5	0.958	0.873	0.914
Fold 3	Faster-RCNN	0.990	0.887	0.936
	SSD MobileNet	0.977	0.791	0.874
	YOLOv5	0.956	0.877	0.915
Fold 4	Faster-RCNN	0.990	0.887	0.936
	SSD MobileNet	0.989	0.807	0.889
	YOLOv5	0.948	0.903	0.925
Average of all folds	Faster-RCNN	0.9898	0.891	0.938
	SSD MobileNet	0.984	0.8054	0.886
	YOLOv5	0.957	0.899	0.927

In Table 9 we perform t-test to compare the accuracy result for the 5-fold for each detection model. T-test is useful technique for comparing values of two sets and to prove that X is better than Y. However, we conclude from the table that Faster RCNN and YOLOv5 is not significant at  $p < .05$ . Also, Faster RCNN is significantly better than SSD MobileNet and YOLOv5 is significantly greater than SSD MobileNet.

Table 9: Statistical Analysis using t-test for object detection

Techniques	t-value	p-value	Significance at $p < 0.5$
Faster RCNN and YOLOv5	1.55179	.079656	The result is not significant at $p < .05$
Faster RCNN and SSD MobileNet	11.89835	$< .00001$	The result is significant at $p < .05$
YOLOv5 and SSD MobileNet	5.16854	.000427	The result is significant at $p < .05$

### 5.3 Classification Techniques

This part is divided into 3 models: Bi-GRU, CNN-LSTM and CNN. All used for classification to classify if cheating occurred or not. We apply 5-fold for all the models.

#### 5.3.1 Bi-GRU

Table 10 presents the 5-fold cross validation for face classification accuracy is ranging from 0.7771 to 0.8251. Fold 3 gives the best accuracy among them. The overall average accuracy of Bi-GRU for all folds is 0.8048.

Table 10: Bi-GRU Results

Fold No.	Overall Accuracy
0	0.8153
1	0.8011
2	0.7771
3	0.8251
4	0.8054
Overall Average	0.8048

In the Figure 18, the loss curve in Bi-GRU classification for cheating and not cheating faces shows the trend of the loss function during training. It indicates if the model is learning from the data and if it's overfitting. By monitoring the loss curve, the training process can be adjusted to improve the model's performance. All the curves are shown below as x axis is for number of epochs & y axis is for loss. Here is the Loss vs Number of Epochs for each fold shown below, we conclude it decreasing so it shows our model perform well. The loss average for Bi-GRU ranging from 0.37 to 0.58.

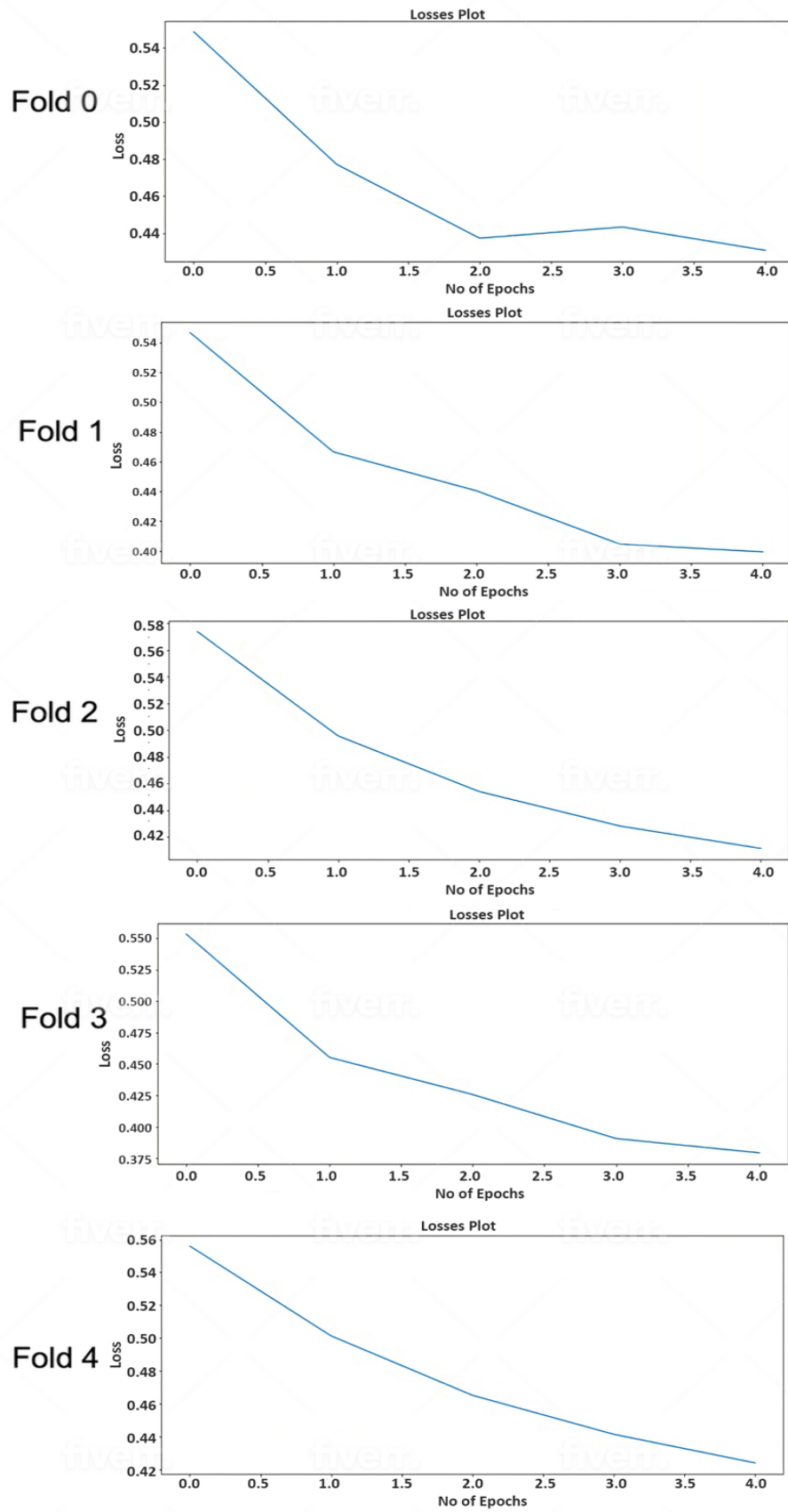


Figure 18: Bi-GRU Loss vs Number of Epochs for all folds



### 5.3.2 CNN-LSTM

Table 11 presents the 5-fold cross validation for face classification is ranging from 0.5251 to 0.5415. Fold 3 gives the best accuracy among them. The overall average accuracy of CNN-LSTM for all folds is 0.53256. Moreover, the CNN-LSTM results compare to other classification model is not performing well. If we trained the model with more temporal video data, the result could be utilized better.

Table 11: CNN-LSTM Results

Fold No.	Overall Accuracy
0	0.5315
1	0.5251
2	0.5305
3	0.5415
4	0.5342
Overall Average	0.53256

In the Figure 19, it shows the total loss curve in CNN-LSTM classification model during training. All the curves are shown below as x axis is for number of epochs & y axis is for loss. The graphs below show the loss is increasing which we conclude that our training not performing very well with loss average between 0.69 to 0.70.

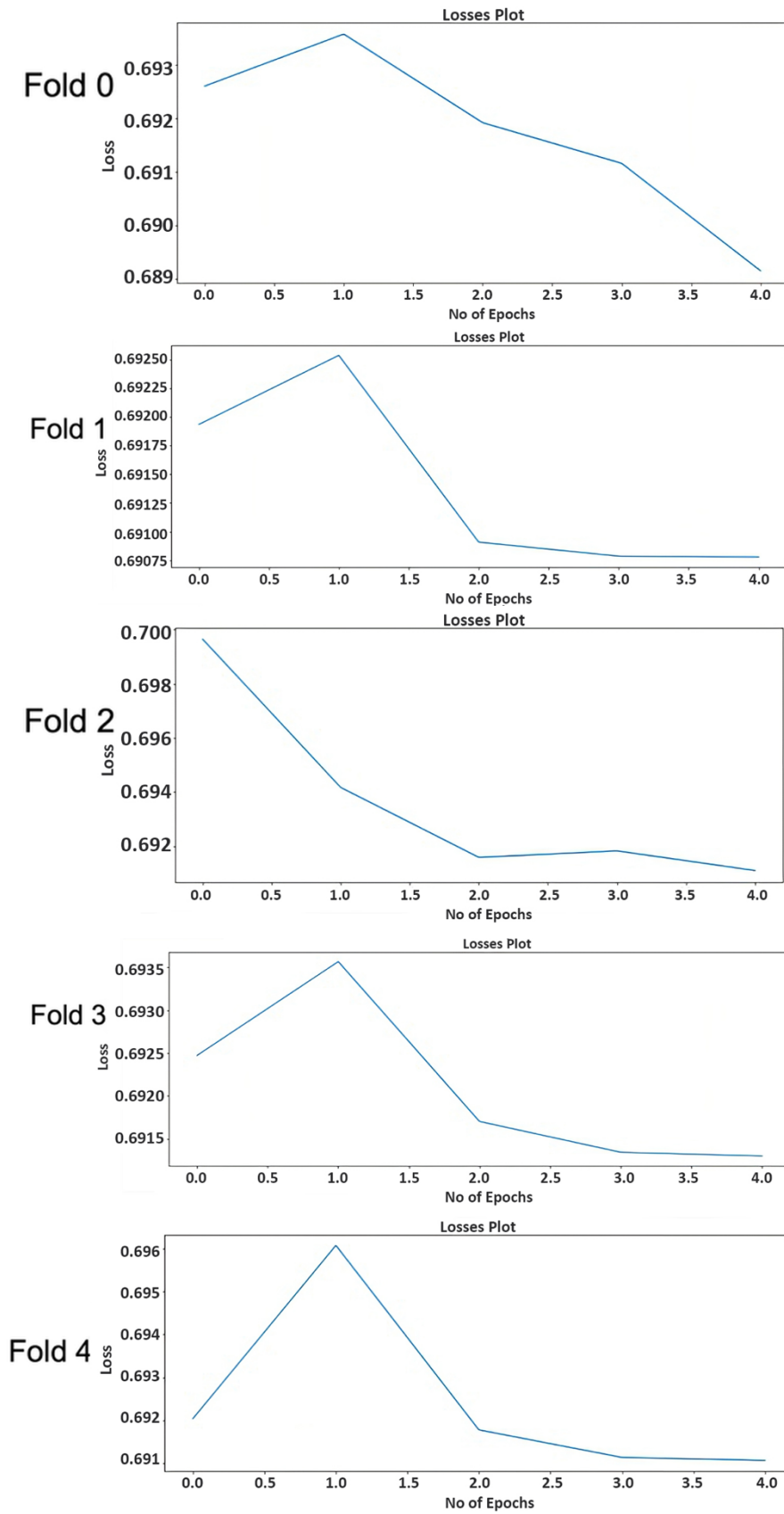


Figure 19: CNN-LSTM Loss vs Number of Epochs for all folds

### 5.3.3 CNN

Table 12 presents the 5-fold cross validation for face classification accuracy ranging from 0.8508 to 0.8713. Fold 4 gives the highest accuracy among them. The overall average accuracy of CNN for all folds is 0.86316. CNN results perform the best compared to CNN-LSTM and Bi-GRU because it is extremely powerful for image classification.

Table 12: CNN Results

Fold No.	Overall Accuracy
0	0.8508
1	0.8566
2	0.8693
3	0.8678
4	0.8713
Overall Average	0.86316

In the Figure 20, the total loss curve in CNN classification model. By monitoring the loss curve, the training process can be adjusted to improve the model's performance. All the curves are shown below as x axis is for number of epochs & y axis is for loss. The model loss average between 0.32 to 0.48, the loss is decreasing which it gave as our training perform extremely well compared to other classification models used.

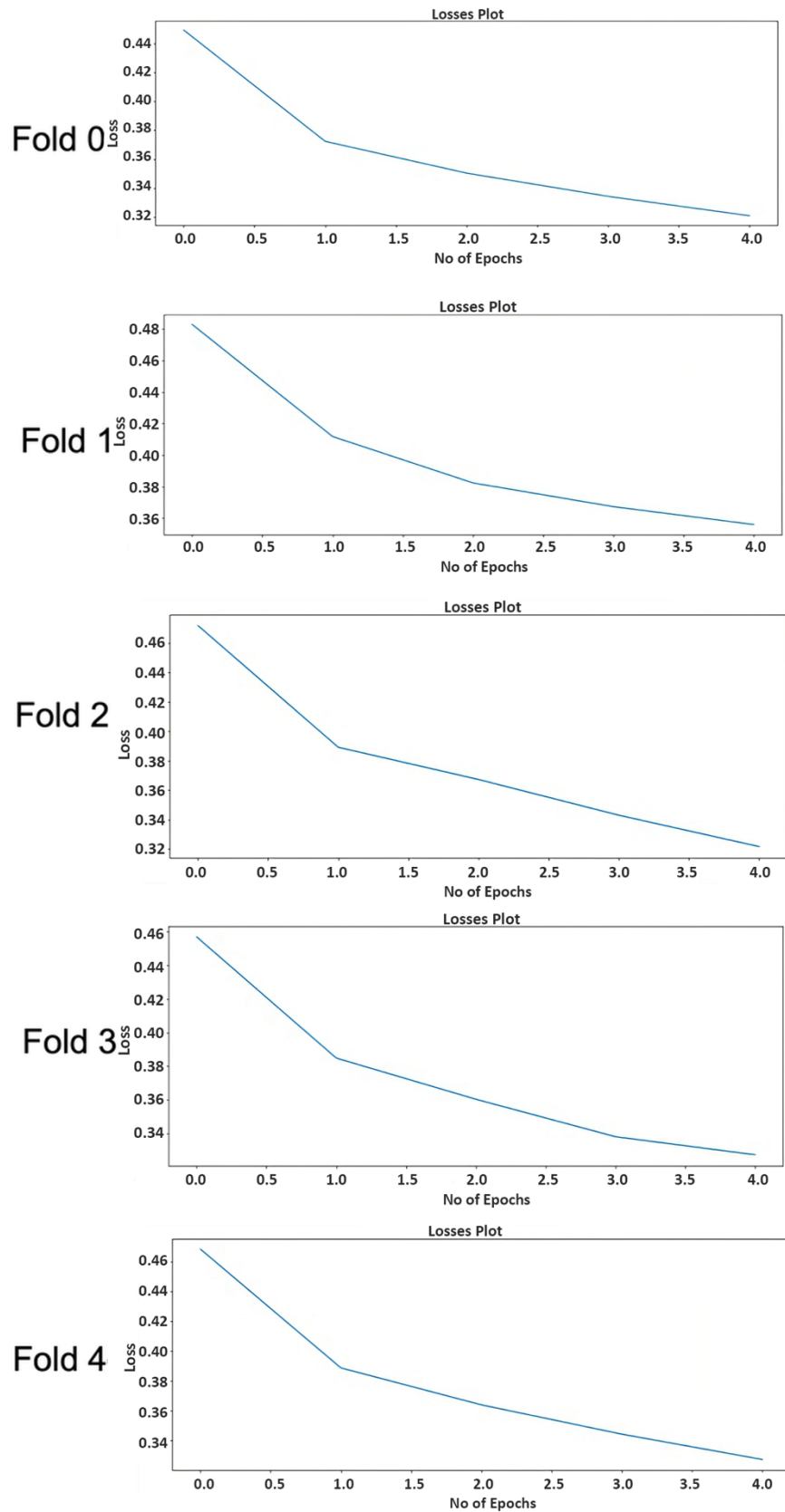


Figure 20: CNN Loss vs Number of Epochs for all folds

Table 13 shows a summary performance of all classification techniques mentioned above. CNN-LSTM result is extremely low because our technique is relying on frames and CNN-LSTM model perform well if we used more temporal data and based on the neural networks. While CNN model accuracy shows the highest among other classification model with accuracy of 86.3%, because it is effective at classifying image.

Table 13: Summary of classification models

<b>Fold</b>	<b>Technique</b>	<b>Overall Accuracy</b>
Fold 0	Bi-GRU	0.815
	CNN-LSTM	0.531
	CNN	0.850
Fold 1	Bi-GRU	0.801
	CNN-LSTM	0.525
	CNN	0.856
Fold 2	Bi-GRU	0.777
	CNN-LSTM	0.530
	CNN	0.869
Fold 3	Bi-GRU	0.825
	CNN-LSTM	0.541
	CNN	0.867
Fold 4	Bi-GRU	0.805
	CNN-LSTM	0.534
	CNN	0.871

Table 13: Summary of classification models (Continued)

Fold	Technique	Overall Accuracy
Average of all folds	Bi-GRU	0.804
	CNN-LSTM	0.532
	CNN	0.863

In Table 14, we perform t-test to compare the accuracy result for the 5-fold for each classification model. We conclude from the table that CNN is significantly better than Bi-GRU and CNN-LSTM.

Table 14: Statistical Analysis using t-test for classification

Model	t-value	p-value	Significance at $p < 0.5$
CNN and Bi-GRU	6.41992	.000102	The result is significant at $p < .05$
CNN and CNN-LSTM	68.01176	$< .00001$	The result is significant at $p < .05$
Bi-GRU and CNN-LSTM	32.12497	$< .00001$	The result is significant at $p < .05$

#### 5.4 Combinations of 5 Proposed Techniques

Our first proposed technique is Faster RCNN with Bi-GRU. For face detection, Faster RCNN is used. The overall accuracy of the Faster RCNN model is selected for face detection as it gives 93.8% accuracy. For face classification as “cheating” or “not cheating”, Bi-GRU is used. The overall accuracy for Bi-GRU for face classification it gives 80.48% accuracy.

Our second proposed technique is SSD MobileNet with Bi-GRU. For face detection, SSD MobileNet is used. The overall accuracy of the SSD MobileNet model is selected for face detection as it gives 88.6% accuracy. For face classification as “cheating” or “not cheating”, the same Bi-GRU is used. The overall accuracy for Bi-GRU for face classification it gives 80.48% accuracy.

In addition, third proposed technique is Faster RCNN with CNN-LSTM. For face detection, the same Faster RCNN is used. The overall accuracy of the Faster RCNN

model is selected for face detection as it gives 93.8% accuracy which is the highest among all. For face classification as “cheating” or “not cheating”, CNN-LSTM is used. The overall accuracy for CNN-LSTM for face classification it gives it gives 53.26% accuracy.

Other proposed technique is YOLOv5 with Bi-GRU. For face detection, YOLOv5 is used. The overall accuracy of the YOLOv5 model is selected for face detection as it gives 92.7% accuracy. For face classification as “cheating” or “not cheating”, Bi-GRU is used. The overall accuracy for Bi-GRU for face classification it gives 80.48% accuracy.

Finally, the technique uses combination of YOLOv5 with CNN. For face detection, YOLOv5 is used. The overall accuracy of YOLOv5 model is selected for face detection as it gives 92.7% accuracy. For face classification, simple CNN is used. The overall accuracy for CNN for face classification it gives 86.3% accuracy.

To combine detection and classification model we use harmonic mean to calculate the overall accuracy. We use this equation. (OA=overall accuracy) (fd= face detection) (fc= face classification) (A=accuracy)

$$OA = \frac{2}{\left(\frac{1}{A(fd)}\right) + \left(\frac{1}{A(fc)}\right)}$$

Table 15: All models accuracy

Technique Name	Overall Accuracy %
Faster RCNN with Bi-GRU	86.63
SSD MobileNet with Bi-GRU	84.35
Faster RCNN with CNN-LSTM	67.94
YOLOv5 with CNN	89.41
YOLOv5 with Bi-GRU	86.18

Finally, Table 15 shows that among all the techniques, YOLOv5 with CNN is selected as it gives the best overall performance which is 89.41%. All the cheating types (chatting from books/notes, chatting with another person in the room/another face detected, using mobile phone) categorized as cheating and if it indicates not cheating it will classify it to not cheating.

Additionally, in Table 16 we perform t-test to compare the accuracy for each fold between the combined models. It concludes that YOLOv5 with CNN is significantly better than other 4 model combinations (Faster RCNN with Bi-GRU, Faster RCNN with CNN-LSTM, SSD MobileNet with Bi-GRU and YOLOv5 with Bi-GRU).

Table 16: Statistical Analysis using t-test for combined model

Model	t-value	p-value	Significance at $p < 0.5$
YOLOv5 with CNN and Faster RCNN with Bi-GRU	4.71707	.000754	The result is significant at $p < .05$
YOLOv5 with CNN and Faster RCNN with CNN-LSTM	52.93166	$< .00001$	The result is significant at $p < .05$
YOLOv5 with CNN and SSD MobileNet with Bi-GRU	9.31318	$< .00001$	The result is significant at $p < .05$
YOLOv5 with CNN and YOLOv5 with Bi-GRU	4.91389	.000587	The result is significant at $p < .05$

Figure 21 shows the testing images result of YOLOv5 with CNN for “cheating” or “not cheating” detection. Green colour indicates not cheating, while blue colour indicates cheating. And all classify it correctly based on the activity.



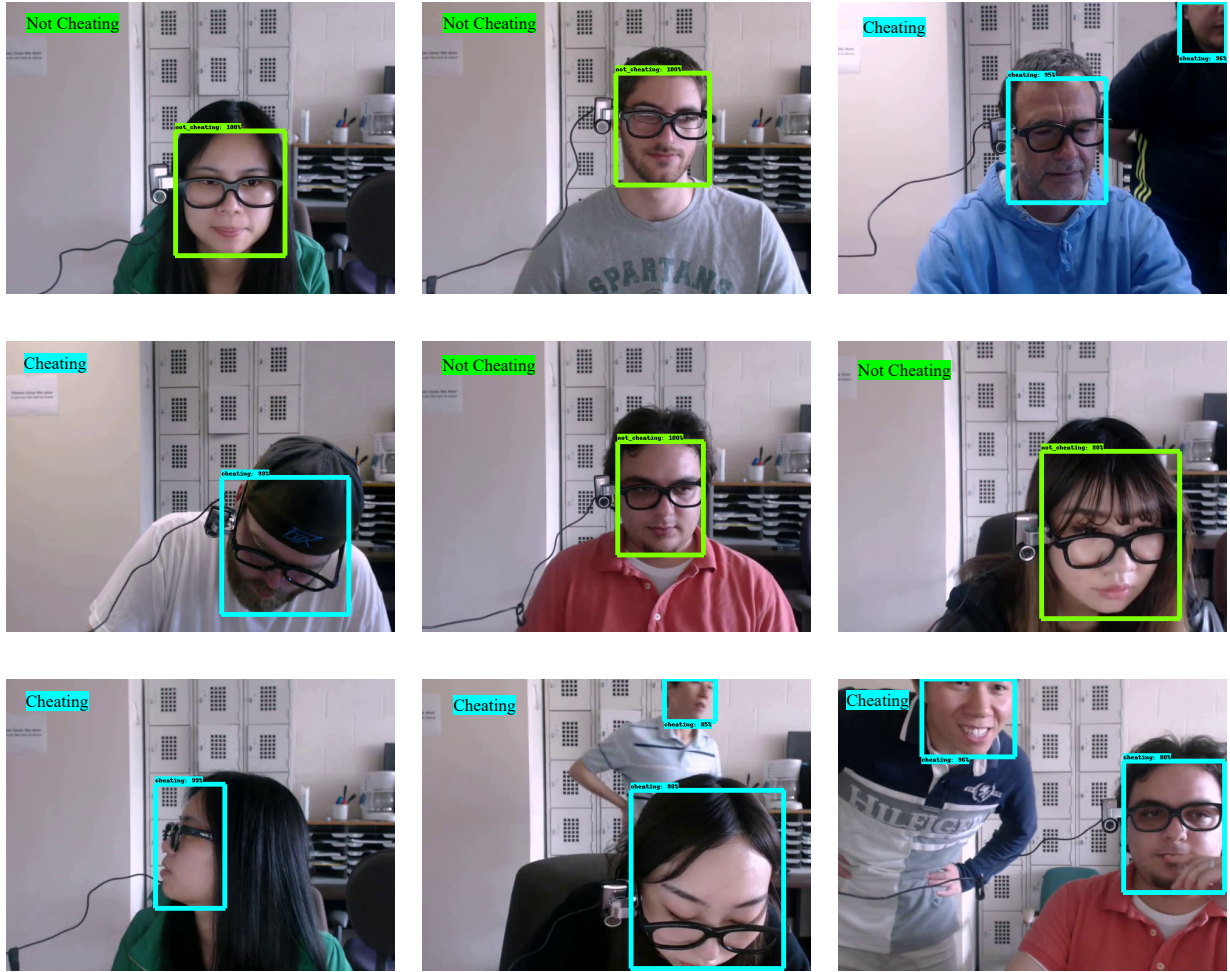


Figure 21: Predicted Cheating/Not Cheating faces using YOLOv5 + CNN

## 5.5 Parameters Tunning

In this section, we find the best accuracy for each model by training with different parameters for object detection and classification models. We used different batch size to be trained with each model, we choose the parameter because we expected to achieve an improved result. Batch size is the quantity of samples handled before a model update. In addition, we planned to train the model each time for other parameters to check if the accuracy will be improved, but due time constrains we kept up with the batch sizes which it gave us good, improved accuracy compared to the original result.

### 5.5.1 Faster RCNN

Table 17 shows training with different batch sizes till we get the best accuracy of Faster-RCNN. Maximum batch size is 8 due to GPU limitation. The highest accuracy

shows at batch size 8 with 94.01%. Moreover, Figure 22 shows accuracy vs batch size for Faster-RCNN model and accuracy increased over the batch sizes increase.

Table 17: Faster-RCNN parameters tuning

Batch Size	Steps	Image Size	Precision	Recall	Accuracy
2	2500	640	0.989	0.854	0.9166
4	2500	640	0.99	0.882	0.9329
6	2500	640	0.99	0.893	0.9390
8	2500	640	0.99	0.895	0.9401

Accuracy vs. Batch Size

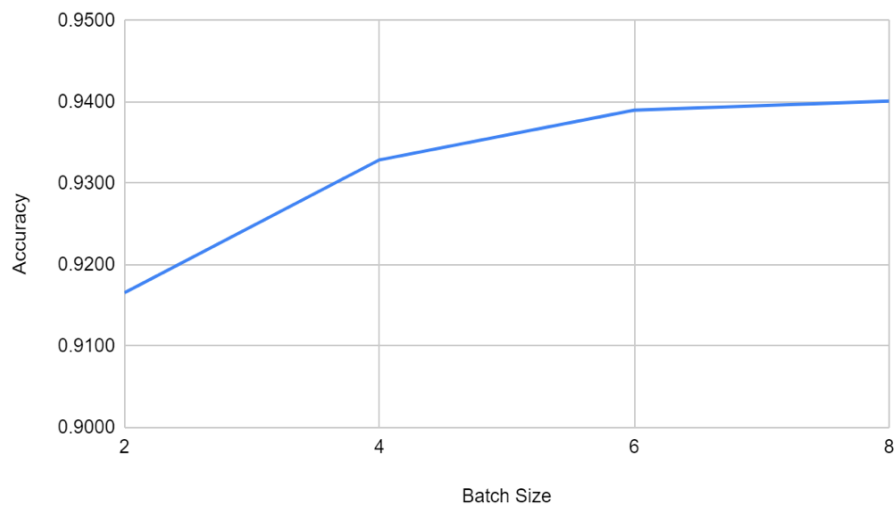


Figure 22: Accuracy Vs Batch Size for Faster RCNN

### 5.5.2 SSD MobileNet

Table 18 presents training with different parameters till we reach the best accuracy of SSD MobileNet. The highest accuracy shows at batch size 8 with 89.98%. Also, Figure 23 shows accuracy vs batch size for SSD MobileNet model. The graph it is fluctuated which it shows us on some batch sizes increase and decrease.

Table 18: SSD MobileNet parameters tuning

Batch Size	Steps	Image Size	Precision	Recall	Accuracy
2	2500	640	0.972	0.769	0.8587
4	2500	640	0.98	0.825	0.8958
6	2500	640	0.988	0.771	0.8661
8	2500	640	0.988	0.823	0.8980

Accuracy vs. Batch Size

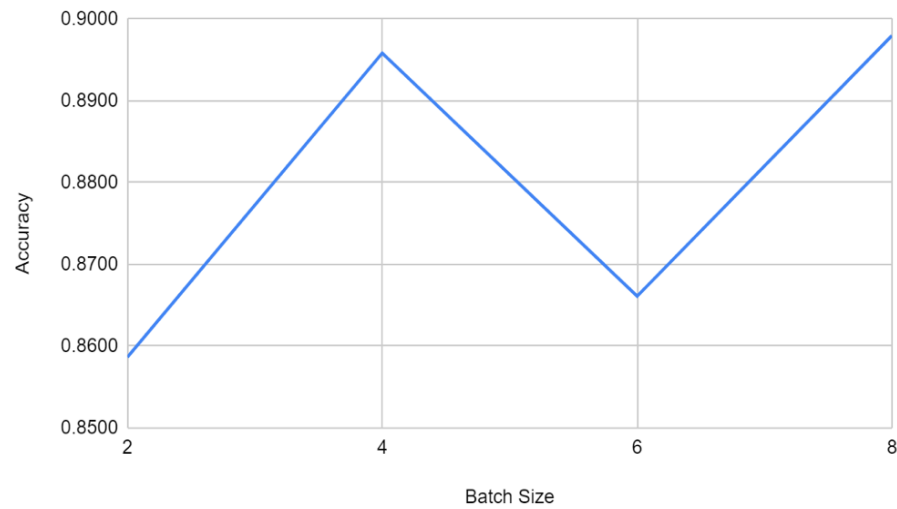


Figure 23: Accuracy Vs Batch Size for SSD MobileNet

### 5.5.3 YOLOv5

Table 19 shows different parameters training to get the best accuracy. The highest accuracy display at batch size 2 with 96.47%, which is the highest among all detection models. Moreover, Figure 24 shows accuracy vs batch size for YOLOv5 model. The graph below it decreased over the batch sizes, sometimes lower batch sizes gave higher accuracy; because for lower batch size, the training time is higher than the training time for higher batch size.

Table 19: YOLOv5 parameters tuning

Batch Size	Epochs	Image Size	Precision	Recall	Accuracy
2	2	640	0.982	0.948	0.9647
4	2	640	0.993	0.931	0.9610
8	2	640	0.986	0.934	0.9593
16	2	640	0.956	0.949	0.9525

Accuracy vs. Batch Size

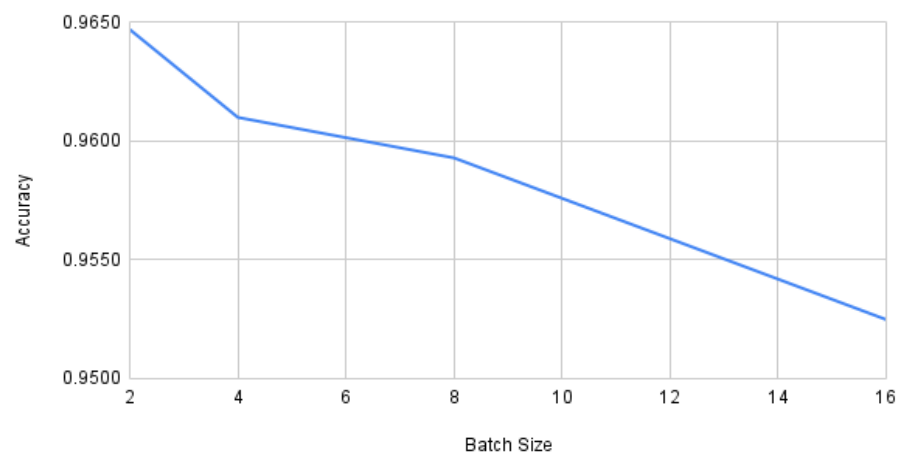


Figure 24: Accuracy Vs Batch Size for YOLOv5

#### 5.5.4 Bi-GRU

Table 20 display training with different parameters to get the best accuracy of Bi-GRU. The highest accuracy shows at batch size 8 with 82.83%. Figure 25 shows accuracy vs batch size for Bi-GRU model.

Table 20: Bi-GRU parameters tuning

Batch Size	Epochs	Optimizers	Accuracy
2	5	Adam	0.8175
4	5	Adam	0.8058
8	5	Adam	0.8283
16	5	Adam	0.8183

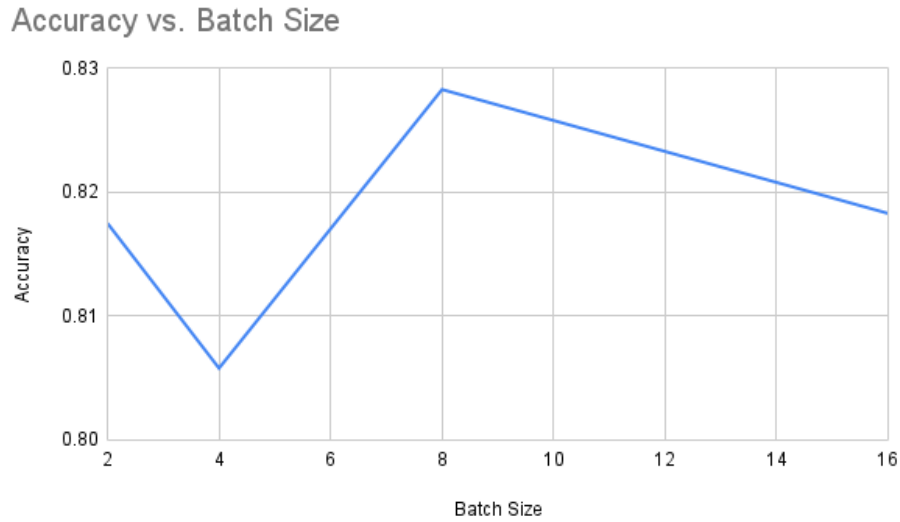


Figure 25: Accuracy Vs Batch Size for Bi-GRU

#### 5.5.5 CNN-LSTM

To get the best accuracy on the CNN-LSTM, Table 21 presents training with different batch sizes (2,4,8,16). The highest accuracy shows at batch size 16 with 53.72%, which the lowest among all the models due to the input data. In addition, Figure 26 shows accuracy vs batch size for CNN-LSTM model.

Table 21: CNN-LSTM parameters tuning

Batch Size	Epochs	Optimizers	Accuracy
2	5	Adam	0.5329
4	5	Adam	0.5278
8	5	Adam	0.5335
16	5	Adam	0.5372

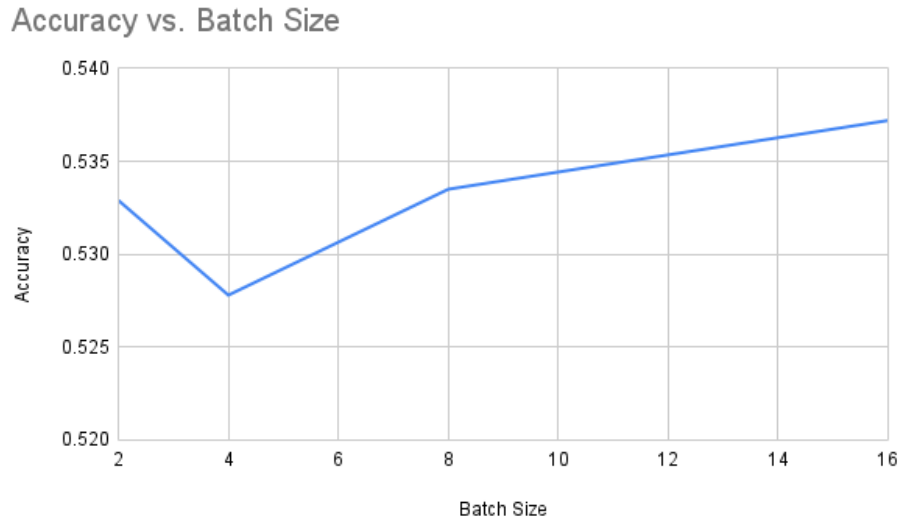


Figure 26: Accuracy Vs Batch Size for CNN-LSTM

### 5.5.6 CNN

Table 22 display different parameter training to get the highest accuracy. The highest accuracy shows at batch size 16 with 89.96%, the highest among classification models. Besides, Figure 27 shows accuracy vs batch size for CNN model.

Table 22: CNN parameters tuning

Batch Size	Epochs	Optimizers	Accuracy
2	5	Adam	0.8333
4	5	Adam	0.8842
8	5	Adam	0.8525
16	5	Adam	0.8996

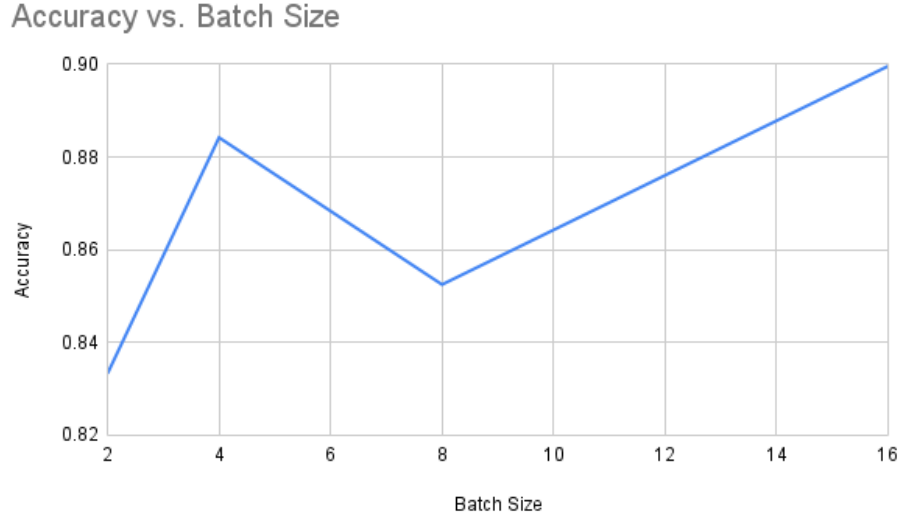


Figure 27: Accuracy Vs Batch Size for CNN

#### 5.5.7 Best Accuracy for All Models

To get improved accuracy for the combined model, we choose best parameter for the batch size accuracy for all models and calculate overall accuracy by using harmonic mean. Table 23 display the improved result for the combined model when we train it with different batch sizes. The combination of YOLOv5 and CNN shows the best accuracy with 93.6%, because YOLOv5 is the latest technology, and it works well with our dataset in terms of detection, and CNN model is performed well for the classification of images.

$$OA = \frac{2}{\left(\frac{1}{A(fd)}\right) + \left(\frac{1}{A(fc)}\right)}$$

Table 23: Improved accuracy

Technique Name	Overall Accuracy %
Faster RCNN with Bi-GRU	88.06
SSD MobileNet with Bi-GRU	86.17
Faster RCNN with CNN-LSTM	68.37
YOLOv5 with CNN	93.1
YOLOv5 with Bi-GRU	89.60

## 5.6 Runtime Performance

For runtime performance, 36 random images are selected to calculate inference time (testing). Table 24 reported for each combined model the total runtime for 36 images and the total runtime for each image/second. Among all the techniques, YOLOv5 with CNN gives the best runtime performance (0.422 sec/image). While Faster-RCNN and Bi-GRU it gave the slowest runtime compared to other combined models.

Table 24: Runtime performance

Technique Name	Total Runtime (sec) for 36 images	Runtime Performance (Inference Time (sec) for each image)
Faster RCNN with Bi-GRU	35.28	0.98
SSD MobileNet with Bi-GRU	26.0019	0.7223
Faster RCNN with CNN-LSTM	25.066	0.6963
YOLOv5 with CNN	15.2	0.422
YOLOv5 with Bi-GRU	18.6	0.5167

In Table 25, we record each training time for each model. Google Colab used to train the models. The table reported for each combined model the runtime performance per fold and total 5-fold training. In summary, YOLOv5 and CNN it performs faster compared to other combined models. However, SSD MobileNet and Bi-GRU it shows the slowest in terms of training.



Table 25: Training time for each algorithm

Technique Name	Model Name	Runtime Performance (Per Fold Training)	Runtime Performance (5-Fold Training)
Faster RCNN with Bi-GRU	Faster RCNN	27 Min 8 Sec	2 Hrs 15 Min 40 Sec
	Bi-GRU	11 Min 33 Sec	57 Min 45 Sec
SSD MobileNet with Bi-GRU	SSD MobileNet	32 Min 32 Sec	2 Hrs 42 Min 40 Sec
	Bi-GRU	11 Min 33 Sec	57 Min 45 Sec
Faster RCNN with CNN-LSTM	Faster RCNN	27 Min 8 Sec	2 Hrs 15 Min 40 Sec
	CNN-LSTM	3 Min 18 Sec	16 Min 30 Sec
YOLOv5 with CNN	YOLOv5	21 Min 15 Sec	1 Hr 46 Min 15 Sec
	CNN	1 Min 54 Sec	9 Min 30 Sec
YOLOv5 with Bi-GRU	YOLOv5	21 Min 15 Sec	1 Hr 46 Min 15 Sec
	Bi-GRU	11 Min 33 Sec	57 Min 45 Sec

## 5.7 Summary and Analysis of Result

In summary, we apply 5-fold cross validation for all the models. For object detection model we used (Faster RCNN, SSD MobileNet and YOLOv5), for the classification model we used (Bi-GRU, CNN-LSTM and CNN). We expected high result from YOLOv5 model because it is the latest technology for the object detection. In addition, we used same parameter for all detection and for all classification models to have fair comparison between the models. The result shows when we combine YOLOv5 and CNN we achieved 89.41% accuracy. However, with training all models with different batch sizes the overall accuracy for the combined model for each combination increased.; the improved result shows that YOLOv5 and CNN gave us the highest accuracy among all with 93.1%. The accuracy increased over the batch sizes.

On the other hand, the runtime performance for the training for each combination it shows that YOLOv5 and CNN had the best training time. YOLOv5 training time was 1 Hr 46 Min 15 Sec for 5-Fold training and CNN training time was 9 Min 30 Sec, the fastest compared to others. For the runtime performance for the testing, we tested 36 images for each combination. YOLOv5 and CNN gave us the best runtime with 0.422 sec/image.

We conclude from the result that the combination of YOLOv5 and CNN model gives the best performances in terms of parameter tuning and runtime.

## Chapter 6: Conclusion

Online education is a brand-new, fascinating possibility that is growing in popularity among both students and educational institutions. E-learning offers distinctive potential in the current context, but it also poses distinctive obstacles. Academic cheating in the sense of cheating, that students seek to do via a variety of means, is the main cause for worry in online exams. As a result, it is the obligation of educational institutions to put more effective systems in place to identify academically dishonest behaviour.

We proposed different model combinations for object detection and classification to produce the most accurate combined model. For object detection models we used (Faster-RCNN, SSD MobileNet and YOLOv5), and for the classification models we used (Bi-GRU, CNN-LSTM and CNN). Our experimental research and findings show that the suggested technique can effectively handle the difficulties associated with cheating at online exams and in avoiding such uncharacteristic behaviour of students. Different techniques were applied to get the best model performance among all. After several experiments, YOLOv5 and CNN were the best combined models for detection and classification with the accuracy of 93.1% in terms of parameter tuning and runtime performance.

Our proposed techniques have some limitations. We used dataset with 24 subjects, if we train our model with larger dataset, the result will be much better. Also, this technique used rely on only frames on detecting cheating, however if we used temporal type of classification model, we could utilize the video in better way. And we did the parameter tuning only for the batch size, there are other parameters we can improve and test.

In the future, we would go further to analyse the sound of the student's video and face movement such as (eyes, mouth, etc.) to detect cheating and it will be identifying student's unusual behaviour more accurately. Also, we will train our dataset in more techniques to find best model performance, and to extend our dataset to be larger.

## References

- Al Khafaji, Y. A., & El Abbadi, N. K. (2022). Traffic signs detection and recognition using a combination of YOLO and CNN. 2022 *Iraqi International Conference on Communication and Information Technologies (IICCIT)*, Basrah, Iraq, 328–334. <https://doi.org/10.1109/IICCIT55816.2022.10010598>
- Al\_ airaji, R. M., Aljazaery, I. A., Alrikabi, H. Th. Salim., & Alaidi, A. H. M. (2022). Automated cheating detection based on video surveillance in the examination classes. *International Journal of Interactive Mobile Technologies (IJIM)*, 16(08), 124–137. <https://doi.org/10.3991/ijim.v16i08.30157>
- Alnassar, F., Blackwell, T., Homayounvala, E., & Yee-king, M. (2021). How well a student performed? A machine learning approach to classify students' performance on virtual learning environment. 2021 *2nd International Conference on Intelligent Engineering and Management (ICIEM)*, United Kingdom, 1–6. <https://doi.org/10.1109/ICIEM51511.2021.9445286>
- Asep, H. S. G., & Bandung, Y. (2019, July). A design of continuous user verification for online exam proctoring on M-Learning. 2019 *International Conference on Electrical Engineering and Informatics (ICEEI)*. Bandung, Indonesia, 284–289. <https://doi.org/10.1109/iceei47359.2019.8988786>
- Atoum, Y., Chen, L., Liu, A. X., Hsu, S. D. H., & Liu, X. (2017a). Automated online exam proctoring. *IEEE Transactions on Multimedia*, 19(7), 1609–1624. <https://doi.org/10.1109/tmm.2017.2656064>
- Atoum, Y., Liu, Y., Jourabloo, A., & Liu, X. (2017b). Face anti-spoofing using patch and depth-based CNNs. 2017 *IEEE International Joint Conference on Biometrics (IJCB)*, Denver, CO, USA. 319–328. <https://doi.org/10.1109/BTAS.2017.8272713>
- Bilen, E., & Matros, A. (2021). Online cheating amid COVID-19. *Journal of Economic Behavior & Organization*, 182(C), 196–211. <https://ideas.repec.org/a/eee/jeborg/v182y2021icp196-211.html>
- Chiang, F., Zhu, D., & Yu, W. (2022). A systematic review of academic dishonesty in online learning environments. *Journal of Computer Assisted Learning*, 38(4), 907–928. <https://doi.org/10.1111/jcal.12656>
- Chotikakamthorn, N., & Tassanaprasert, S. (2020, October 7). Affordable proctoring method for Ad-hoc Off-campus exams. *Proceedings of the 21st Annual Conference on Information Technology Education* 36(6), 1589–1602. <https://doi.org/10.1145/3368308.3415421>

- Chuang, C. Y., Craig, S. D., & Femiani, J. (2017). Detecting probable cheating during online assessments based on time delay and head pose. *Higher Education Research & Development*, 36(6), 1123–1137.  
<https://doi.org/10.1080/07294360.2017.1303456>
- Cote, M., Jean, F., Albu, A. B., & Capson, D. (2016, March). Video summarization for remote invigilation of online exams. *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, New York, USA, 1-9.  
<https://doi.org/10.1109/wacv.2016.7477704>
- Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3(1).  
<https://doi.org/10.1017/atsip.2013.9>
- Dhilipan, J., Vijayalakshmi, N., Suriya, S., & Christopher, A. (2021). Prediction of Students Performance using Machine learning. *IOP Conference Series: Materials Science and Engineering*, 1055(1), 012122. <https://doi.org/10.1088/1757-899x/1055/1/012122>
- Dilini, N., Senaratne, A., Yasarathna, T., Warnajith, N., & Seneviratne, L. (2021, December 1). Cheating Detection in Browser-based Online Exams through Eye Gaze Tracking. *2021 6th International Conference on Information Technology Research (ICITR)*, Moratuwa, Siri Lnaka, 1-8.  
<https://doi.org/10.1109/icitr54349.2021.9657277>
- Duhaim, A. M., Al-mamory, S. O., & Mahdi, M. S. (2022). Cheating detection in online exams during Covid-19 Pandemic using data mining techniques. *Webology*, 19(1), 341–366. <https://doi.org/10.14704/web/v19i1/web19026>
- El Kohli, S., Jannaj, Y., Maanan, M., & Rhinane, H. (2022). Deep learning: New approach for detecting scholar exams fraud. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVI-4/W3-2021, 103–107. <https://doi.org/10.5194/isprs-archives-xlvi-4-w3-2021-103-2022>
- Hasri, A., Supar, R., Azman, N. D. N., Sharip, H., & Yamin, L. S. M. (2022). Students' attitudes and behavior towards academic dishonesty during online learning. *International Academic Symposium of Social Science 2022*, 82(1), 36.  
<https://doi.org/10.3390/proceedings2022082036>
- Houssein, E. H., Abohashima, Z., Elhoseny, M., & Mohamed, W. M. (2022). Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision. *Expert Systems with Applications*, 194, 116512.  
<https://doi.org/10.1016/j.eswa.2022.116512>

- Hu, S., Jia, X., & Fu, Y. (2018). Research on abnormal behavior detection of online examination based on image information. *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, 02, 88–91. <https://doi.org/10.1109/IHMSC.2018.10127>
- Hussein, F., Al-Ahmad, A., El-Salhi, S., Alshdaifat, E., & Al-Hami, M. (2022). Advances in contextual action recognition: Automatic cheating detection using machine learning techniques. *Data*, 7(9), 122. <https://doi.org/10.3390/data7090122>
- Hylton, K., Levy, Y., & Dringus, L. P. (2016). Utilizing webcam-based proctoring to deter misconduct in online exams. *Computers & Education*, 92-93, 53–63. <https://doi.org/10.1016/j.compedu.2015.10.002>
- Indi, C. S., Pritham, V., Acharya, V., & Prakasha, K. (2021). Detection of malpractice in E-exams by head pose and gaze estimation. *International Journal of Emerging Technologies in Learning (IJET)*, 16(08), 47. <https://doi.org/10.3991/ijet.v16i08.15995>
- J S, A., Kumaran, H. S., U, S., Rajesh, K. P. B. V., & R, L. (2021). Deep learning based approach for facilitating online proctoring using transfer learning. *2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, Chennai, India, 306–312. <https://doi.org/10.1109/ICCCSP52374.2021.9465530>
- Jalali, K., & Noorbehbahani, F. (2017). An automatic method for cheating detection in online exams by processing the students Webcam images. *3rd Conference on Electrical and Computer Engineering Technology (E-Tech 2017)*, Tehran, Iran. <https://t.ly/dviks>
- Kamalov, F., Sulieman, H., & Santandreu Calonge, D. (2021). Machine learning based approach to exam cheating detection. *PLOS ONE*, 16(8), e0254340. <https://doi.org/10.1371/journal.pone.0254340>
- Kasliwal, G. (2015). *Cheating detection in online examinations* [Masters Thesis, San Jose State University]. <https://doi.org/10.31979/etd.y292-cddh>
- Ketab, S. S., Clarke, N. L., & Dowland, P. S. (2017). A robust e-Invigilation System employing multimodal biometric authentication. *International Journal of Information and Education Technology*, 7(11), 796–802. <https://doi.org/10.18178/ijiet.2017.7.11.975>

- Khan, A. R., Saba, T., Khan, M. Z., Fati, S. M., & Khan, M. U. G. (2022). Classification of human's activities from gesture recognition in live videos using deep learning. *Concurrency and Computation: Practice and Experience*, 34(10), e6825. <https://doi.org/10.1002/cpe.6825>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90. <https://doi.org/10.1145/3065386>
- Li, H., Xu, M., Wang, Y., Wei, H., & Qu, H. (2021). A Visual Analytics Approach to Facilitate the Proctoring of Online Exams. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, New York, USA, 1–17. <https://doi.org/10.1145/3411764.3445294>
- Li, S., & Liu, T. (2021). Performance prediction for higher education students using deep learning. *Complexity*, 2021, 1–10. <https://doi.org/10.1155/2021/9958203>
- Li, X., Ma, X., Xiao, F., Xiao, C., Wang, F., & Zhang, S. (2022). Time-series production forecasting method based on the integration of Bidirectional Gated Recurrent Unit (Bi-GRU) network and Sparrow Search Algorithm (SSA). *Journal of Petroleum Science and Engineering*, 208, 109309. <https://doi.org/10.1016/j.petrol.2021.109309>
- Liu, F., Chen, Z., & Wang, J. (2018). Video image target monitoring based on RNN-LSTM. *Multimedia Tools and Applications*, 78(4), 4527–4544. <https://doi.org/10.1007/s11042-018-6058-6>
- Malhotra, M., & Chhabra, I. (2022). Student invigilation detection using deep learning and machine after Covid-19: A review on taxonomy and future challenges. In *Future of organizations and work after the 4th industrial revolution* (pp. 311–326). Springer. [https://doi.org/10.1007/978-3-030-99000-8\\_17](https://doi.org/10.1007/978-3-030-99000-8_17)
- Malik, A. A., Hassan, M., Rizwan, M., Mushtaque, I., Lak, T. A., & Hussain, M. (2023). Impact of academic cheating and perceived online learning effectiveness on academic performance during the COVID-19 pandemic among Pakistani students. *Frontiers in Psychology*, 14. <https://doi.org/10.3389/fpsyg.2023.1124095>
- Masud, M. M., Hayawi, K., Mathew, S. S., Michael, T., & El Barachi, M. (2022). Smart online exam proctoring assist for cheating detection. In *Advanced data mining and applications* (pp. 118–132). Springer, Cham. [https://doi.org/10.1007/978-3-030-95405-5\\_9](https://doi.org/10.1007/978-3-030-95405-5_9)

- Noorbehbahani, F., Fanian, A., Mousavi, R., & Hasannejad, H. (2015). An incremental intrusion detection system using a new semi-supervised stream classification method. *International Journal of Communication Systems*, 30(4), e3002. <https://doi.org/10.1002/dac.3002>
- Özgen, A. C., Öztürk, M. U., & Bayraktar, U. (2021). Cheating detection pipeline for online interviews and exams. *ArXiv:2106.14483 [Cs]*, 21(6), 14–31. <https://arxiv.org/abs/2106.14483>
- Patel, I., & Patel, S. (2020). An optimized deep learning model for flower classification using NAS-FPN and faster R- CNN. *International Journal of Scientific & Technology Research*, 9(3), 2277–8616. <http://www.ijstr.org/final-print/mar2020/An-Optimized-Deep-Learning-Model-For-Flower-Classification-Using-Nas-fpn-And-Faster-R-cnn.pdf>
- Poddar, D., Nagori, S., Mathew, M., Maji, D., & Garud, H. (2019). Deep learning based parking spot detection and classification in Fish-Eye images. *2019 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, Bangalore, India, 1–5. <https://doi.org/10.1109/conecct47791.2019.9012933>
- Prathish, S., & Bijlani, K. (2016). An intelligent system for online exam monitoring. *2016 International Conference on Information Science (ICIS)*, Kochi, India, 138–143. <https://doi.org/10.1109/infosci.2016.7845315>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *CV Foundation*. [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/redmon\\_you\\_only\\_look\\_cvpr\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/redmon_you_only_look_cvpr_2016_paper.pdf)
- Sharma, N., Sharma, R., & Jindal, N. (2021). Machine learning and deep learning applications-A vision. *Global Transitions Proceedings*, 2(1), 24–28. <https://doi.org/10.1016/j.gltp.2021.01.004>
- Shdaifat, A. M., Obeidallah, R. A., Ghazal, G., Abu Sarhan, A., & Abu Spetan, N. R. (2020). A proposed Iris recognition model for authentication in mobile exams. *International Journal of Emerging Technologies in Learning (IJET)*, 15(12), 205. <https://doi.org/10.3991/ijet.v15i12.13741>
- Tiong, L. C. O., & Lee, H. J. (2021). E-cheating prevention measures: Detection of cheating at online examinations using deep learning approach -- A case study. *ArXiv:2101.09841 [Cs]*. <https://arxiv.org/abs/2101.09841>



- Turani, A. A., Alkhateeb, J. H., & Alsewari, A. A. (2020). Students online exam proctoring: A case study using 360 degree security cameras. *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, Bangladesh, 1–5. <https://doi.org/10.1109/etcce51779.2020.9350872>
- Wan, Z., Li, X., Xia, B., & Luo, Z. (2021). Recognition of cheating behavior in examination room based on deep learning. *2021 International Conference on Computer Engineering and Application (ICCEA)*, Kunming, China, 204–208. <https://doi.org/10.1109/iccea53728.2021.00048>
- Wang, Y., Wang, J., Zhang, W., Zhan, Y., Guo, S., Zheng, Q., & Wang, X. (2021). A survey on deploying mobile deep learning applications: A systemic and technical perspective. *Digital Communications and Networks*, 8(1), 1–17. <https://doi.org/10.1016/j.dcan.2021.06.001>



جامعة الإمارات العربية المتحدة  
United Arab Emirates University



## UAE UNIVERSITY MASTER THESIS NO. 2023:19

The suggested solution used a public dataset contains recorded videos of the student in real time exam. The goal is to analyze the videos using various deep learning methods to find best/accurate combinations of models for face detection and face classification, to detect if cheating occurred or not.

**Aysha Alkalbani** received her Master of Science in Information Technology Management from the Department of Information Systems and Security, College of Information Technology at UAE University, UAE. She received her Bachelor of Science in Information Security from the College of Information Technology, UAEU University.

[www.uaeu.ac.ae](http://www.uaeu.ac.ae)



عمادة المكتبات  
Libraries Deanship

جامعة الإمارات العربية المتحدة  
United Arab Emirates University

