

**BEHAVIORAL DETECTION AND PREVENTION OF CHEATING DURING
ONLINE EXAMINATION USING DEEP LEARNING APPROACH**

GABRIEL MUCHANGI KIURA

**A Thesis Submitted to the School of Science and Technology in Partial Fulfillment
for the requirement of the Degree of Master of Science in Computer Information
Systems of Kenya Methodist University**

JULY, 2023

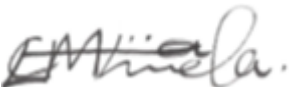
DECLARATION

I declare that this thesis is my original work and has not been presented for a degree or any other award in any other University

Signed: Date: **18/9/2023**.....

CIS-3-0988-1/2017

This thesis has been submitted for examination with our approval as university supervisors

1. Signed:  Date:**18/9/2023**.....

Dr. Lawrence Mwenda Muriira

2. Signed:  Date:**22/9/2023**.....

Mr. Nicholas Riungu

COPYRIGHT

© Gabriel Muchangi Kiura

All rights reserved. No part of this Thesis may be reproduced, stored in any retrieval system or transmitted in any form or by any means, electronically, mechanically, by photocopying or otherwise, without prior written permission of the author or Kenya Methodist University, on that behalf.

DEDICATION

This research thesis is dedicated to my wife and children for their great support and encouragement during my studies.

ACKNOWLEDGEMENT

I hereby wish to thank Almighty God for granting me this exceptional prospect to have this proposal compiled to the best of my ability through good health. I also wish to extend my heartfelt gratitude to my family and classmates for their overwhelming support throughout this course. I also incline great appreciation to my supervisors, Dr. Lawrence Mwenda Murira and Mr. Nicholas Riungu, who by all ways and means guided and inspired me all through this work. I also wish to acknowledge the entire Kenya Methodist University for their co-operation towards providing library facilities concerning the research study and giving me an opportunity to undertake this course. I also acknowledge my colleagues for their immeasurable support.

ABSTRACT

With the expansion of new technologies over the last years learning has grown and universities are utilizing it in offering online exams. Cheating during has also gone up regardless of the technologies or means which universities are using. The study addressed the issues that are experienced during online evaluation of student taking exams, in universities. Currently many student engage in exam malpractice through copying during online exam. To be able to determine the behavioral metric data was downloaded from the free data repository. The data was processed, validated, trained and evaluated. Quantative research methods was used in this research. By analyzing distinct behavioral patterns and strategies employed by cheating students, the research provides valuable insights into the motivations and factors that drive such behavior. The study also identifies significant visual features present in images that indicate instances of cheating, which enhance the performance of deep learning models. Various deep learning models, including Dense Net, Mobile Net, ResNets, and Convolutional Neural Networks (CNN), are developed and evaluated for detecting and classifying cheating behavior during online examinations. The evaluation results show that the Mobile Net model achieved the highest test accuracy of 93.4%, outperforming the other models. It demonstrated strong predictive ability, accurate classification, and efficient computation time. Additionally, the identification of significant visual features and the development of deep learning models tailored for cheating detection contribute to the field of automated cheating detection, providing a foundation for future research. However, certain limitations should be acknowledged. The performance of the deep learning models may be influenced by the quality and diversity of the training dataset, and further investigation is needed to determine their effectiveness in detecting evolving cheating strategies. Based on the evaluation findings and identified limitations, several recommendations are proposed. Firstly, improving the quality and diversity of the training dataset through data collection was recommended to enhance the performance of deep learning models. Continuous model training is essential to adapt to emerging cheating strategies, requiring regular incorporation of new instances of cheating behaviors into the training dataset. Further exploration and refinement of significant visual features can enhance model accuracy through feature engineering techniques. Ensemble methods, such as model averaging or stacking, should be considered to improve overall model performance. Collaboration among researchers, educators, and policymakers from different educational contexts can facilitate cross-context evaluation and provide insights into the generalizability of the models. The findings of the research can be used by policy maker when making decision patterning online exams to ensure there is credibility of the online exams. The findings also forms the bases of academia future research to improve on this research. Ethical considerations, including privacy concerns and fairness in the detection process, should be addressed transparently. Lastly, educational institutions should prioritize creating awareness and fostering a culture of academic integrity through comprehensive guidelines and student education.

TABLE OF CONTENT

DECLARATION	ii
COPYRIGHT	iii
DEDICATION	iv
ACKNOWLEDGEMENT	v
ABSTRACT	vi
TABLE OF CONTENT	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ACRONYMS AND ABBREVIATION	xi
CHAPTER ONE	1
INTRODUCTION	1
1.1 Background of the Study.....	1
1.2 Statement of the Problem.....	4
1.3 Purpose of the Study.....	5
1.4 General Objective.....	6
1.5 Specific Objectives.....	6
1.6 Research Questions.....	6
1.7 Justification of the Study.....	7
1.8 Limitations of the Study.....	7
1.9 Delimitation of the Study.....	8
1.10 Significance of the study.....	8
1.11 Assumptions of the Study.....	9
1.12 Definition of Terms.....	11
CHAPTER TWO	14
LITERATURE REVIEW	14
2.1 Introduction.....	14
2.2 Theoretical framework on Behavioral Cheating Theories.....	14
2.3 Empirical Review.....	18
2.4 Artificial Intelligence.....	23
2.5. Conceptual Frame Work.....	34
2.6 Knowledge Gap.....	37

2.7 Literature Review Summary	37
CHAPTER THREE	38
RESEARCH METHODOLOGY	38
3.1 Introduction	38
3.2 Research Design	39
3.3 Location of Study	40
3.4 Target Population	41
3.5 Data Collection.....	42
3.6 Data Preprocessing	44
3.7 Deep Learning Models	53
3.8 Model Evaluation	60
3.9 Statistical Analysis	61
3.10 Ethical Considerations.....	62
3.11 Limitations of the Study.....	63
CHAPTER FOUR	65
RESULTS AND DISCUSSION	65
4.1 Introduction	65
4.1 Cheating Behavioral Metrics in an online examination	66
4.2 Scrutinize Techniques Used By Student to Cheat During Online Examinations...	67
4.3 Proposed/Use of Existing Model and To Training Data	72
4.4 Evaluation of Developed Deep Learning Models Evaluation.....	86
CHAPTER FIVE	95
SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	95
5.1 Introduction	95
5.2 Summary of Findings	95
5.3 Contribution to Knowledge	97
5.4 Limitations.....	97
5.5 Conclusions	97
5.6 Recommendations	98
5.7 Further research.....	100
REFERENCES	101
APPENDICES	105

LIST OF TABLES

Table 4.1 Sample table of Classification of CSV Files.....	71
Table 4.2 CNN Model Testing Output.....	88
Table 4.3 DenseNet Model Testing Output	88
Table 4. 4 MobileNet Model Testing Output.....	89
Table 4.5 ResNets Model Testing Output.....	90
Table 4. 6 Comparison of the Models.....	91

LIST OF FIGURES

Figure 2.0 Simple mobilenetV2 model arch	31
Figure 2.1 Conceptual Framework.....	35
Figure 3 .1Conversion of Wav to MP4.....	46
Figure 3. 2Python script converts MP4 to JPEG	47
Figure 3. 3Classification of JPEG Images	48
Figure 3 .4Renaming of JPEG Files.....	49
Figure 3.5Extracting Features from JPEG Images.....	51
Figure 3 .6Proposed Model Framework	55
Figure 3.7CNN Architecture.....	55
Figure 3 .8Deneset Model.....	57
Figure 3.9MobileNet Model	58
Figure 3.10Resnet Model.....	59
Figure 4. 1Function to Read Movie Files	70
Figure 4 .2Function Call	70
Figure 4.3CNN Training Loss	75
Figure 4.4CNN Training Accuracy.....	75
Figure 4. 5CNN Model Summary.....	76
Figure 4. 6DenseNet Model Training Loss.....	78
Figure 4.7DenseNet Model Training Accuracy.....	79
Figure 4. 8DenseNet Model Summary	79
Figure 4. 9MobileNet Model Training loss and Accuracy	80
Figure 4.10MobileNet Model Traning Accuracy	82
Figure 4.11MobileNet Model Summary	82
Figure 4. 12ResNets Model Training Loss	83
Figure 4. 13 ResNets Model Training Accuracy	84
Figure 4. 14ResNets Model Summary	85

LIST OF ACRONYMS AND ABBREVIATION

AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Programming Interface
ATR	Attenuated Total Reflectance
CNN	Convolutional Neural Networks
CONV	Convolutional layers
ConvNet	Convolutional Neural Network
CRISP-DM	Cross Industry Standard Process in Data Mining
CSO	Combined Sewer Overflow
CSV	Comma Separated Value
DNN	Deep Neural Networks
Dr.	Doctor
DSC	Dice Similarity Coefficient
GRU	Gated Recurrent Units
IoU	Intersection over Union
KDD	Knowledge Discovery in Databases
k-NN	k Nearest Neighbor
LSTM	Long Short Term Memory
ML	Machine Learning
QA/QC	Quality Assurance and Quality Control
RDF	Random Decision Forest
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
SEMMA	Sample, Explore, Modify, Model and Assess
SIFT	Scale Invariant Feature Transform Algorithm
SVM	Support Vector Machine

KeMU Kenya Methodist University
WWTPs Waste Water Treatment Plants

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Artificial intelligence (AI), generally expressed by the general public as the ability of machines or computers to think and act as humans do, represents the efforts towards computerized systems to imitate the human mind and actions (Wartman & Combs, 2018). Artificial intelligence can be used in education sector to solve the new upcoming emerging issues like cheating during exams. With more usage of artificial intelligence in education, major transformations can be foreseen in the education systems and its processes. One of the artificial intelligence is deep learning. Deep learning can be used for detection in this investigation. It is a machine learning methodology that uses many layers of nonlinear information processing to achieve feature extraction, pattern identification, and classification.

The integration of technology into education precipitated by the COVID-19 pandemic witnessed the haphazard transition of face-to-face teaching to online modes, despite the challenges of teaching online, the laborious task was during online evaluation (Bilen & Matros, 2021). The quality of examinations is a continuous process that serves to not only provide feedback to educators on teaching and learning but is the epicenter of quality graduates produced (Sara et al., 2020). Therefore, necessary measures need to be put in place to ensure the legitimacy, reliability, and authenticity of the examination process as well as the grades obtained (Dadzie & Annan-brew 2023; Noorbehbahani et al., 2022). Examination malpractice is any aberrant behavior demonstrated by a student or anybody assigned with administering an examination before, during or after the

examination that breaches the norms and standards governing administration of examinations (Dadzie & Annan-brew 2023). Traditional cheating methods include hiding notes behind rulers, writing on arms and hands, online cheating methods are sharing screens, searching for answers online and social media usage during examinations (Bilen & Matros, 2021). The ability to anticipate cheating by behaviour detection has developed over the years and its application in online examinations has proven to be beneficial to universities (Al-airaji et al., 2022; Alin et al., 2022). The pictorial structure model is one such model that uses video surveillance and intelligent systems to analyze human behaviour and detect unusual events in posture where the student tries peeping at the work of another candidate (Al-airaji et al., 2022; Lv et al., 2014). Furthermore, observation of the student's iris to detect movement to copy materials from mobile phones is detected and sounds an alarm notifying proctors of the irregularity (Alairaji et al., 2022).

Globally, a study carried out by Tweissi et al. (2022) analysing the use of artificial intelligence (AI)-based auto proctoring for online examinations to monitor student behaviours identified that despite errors in the system human observation coupled with AI intelligence was superior in curbing examination cheating. A study by Tiong et al. (2021) demonstrated the use of AI technology to curtail online examination cheating through e-cheating intelligent agent with Internet Protocol (IP) detector and behaviour detector protocols on four deep learning protocols with accuracy levels of 90%. Advances in technology have led to more robust effective and efficient approaches incorporating deep learning models for real time cheating detection from recorded video frames and speech (Kaddoura & Gumaei, 2022). This is achieved by forward-facing

camera and speech dishonest detection that extract important items from visual pictures and voice (Kaddoura & Gumaei, 2022).

Regionally, technology can be used to ease examination cheating in third world countries incorporates the use of technologies such as facial recognition technology, biometric systems, and closed-circuit television (CCTV) cameras (Onyema et al., 2019).

The context of low- and middle-income countries (LMICs) hampers the adoption of advanced technologies in curbing examination malpractice because of the cost implications associated with such technologies (Nganchi & Charlotte 2020). Further compounding these challenges are infrastructural challenges (internet, power, and manpower) resistance to change and policy gaps (Valizadeh 2022). These challenges have contributed to increased cheating during online examinations within the region attributable to unavailability of resources to support the use of advanced behavioral technology in examination proctoring (Muchemwa 2023). The traditional methods of curbing cheating in examinations by checking students properly, sitting arrangements and banning digital gadgets in examinations rooms are more commonly used in majority of the traditional face to face examination centers withing the African region (Dadzie &Annan-brew 2023). Theses traditional face to face approaches however effective might require additional support in online examinations.

Locally, the surge in examination malpractice among Kenyan universities has reached worrying trends and while most universities during the COVID-19 pandemic opted to have online classes but delayed examinations till normalcy resumed (Mulongo et al., 2019; Macharia, 2022). Majority of the universities that administered online examinations observed considerable numbers of cheating in examinations hence

hampering the integrity of the exercise (Macharia, 2022). There is limited literature on the application of advanced technology such as AI in reducing examination cheating since most universities have resulted to traditional pen and paper examinations. This study therefore will focus on the use of behaviometrics based on Information Technology and machine learning where patterns will be recognized, anomaly detected, visualized, and detecting examination cheating during online exams.

1.2 Statement of the Problem

The credibility of online exams is questionable, it is widely believed that those students who do online exams cheat or are assisted during exams, and thus employers are skeptical on employing these graduates (Maeda, M., 2019). Majority of the universities in Kenya had blended approaches to teaching students in far flung areas on distance learning modes where teaching was done online through (LMS), and examinations done traditionally through pen and paper (Li et al., 2020). The shift to fully online teaching and examinations was unanticipated. Ideally, online exams require intricate protection to prevent cheating. Firstly, the use traditional approaches to prevent cheating are recommended: identification of the student seating the exam, frisking students to ensure unauthorized materials do not enter examination rooms, sitting arrangements, all electronic devices to be switched off, ensuring students have not written on arms, hands, and thighs (Nizam et al., 2020). Secondly, the use of video monitoring through CCTV integrated with AI technology that monitors behavior, head, speech, and eye movement during the exam and sounds an alarm (Tiong & Lee, 2021). Thirdly, the use of IP detector integrated in the Learning management system (LMS) to prevent students opening additional windows during the examination period. The use of predictive

analytic systems implicitly collect data while students interact with virtual learning environments and predict trends and patterns of student behavior (Noorbehbahani et al., 2022).

Despite this, students doing online examinations have been cheating during examinations using mobile phones through texts and social media, use of external media devices with content of examinations they are doing, accessing other files and software during examinations (Nguyen et al., 2020). Furthermore, looking at other candidates' work and accessing other files and softwares, open discussions during the examination. Since online examinations have no proctor, this has led to many impersonations as it's difficult to monitor students as sometimes webcam video is not clear. The study therefore intends to develop a model for preventing cheating during online examinations.

1.3 Purpose of the Study

Researcher's main aim in this study was to learn and identify the behavioral methods of cheating during online examination and propose a model that can be used to prevent online cheating at universities in Kenya using deep learning. The primary objective of this research endeavors to investigate and comprehend the intricate nature of cheating behavior exhibited during online examinations. This study seeks to contribute to the field of academia by proposing a robust deep learning model that can effectively deter instances of online cheating within the context of universities in Kenya.

As the prevalence of online education continues to grow, ensuring the integrity of assessments has become a critical concern for educational institutions. However, the advent of online examinations has introduced novel challenges, particularly in deterring cheating behaviors. Recognizing the significance of academic honesty, this study aims to

comprehensively explore the behavioral dimensions of cheating during online examinations, thereby illuminating the underlying factors and motivations that lead to academic misconduct.

1.4 General Objective

The general objective of this study was to employ a deep learning approach to investigate and comprehend learner behaviors in online assessment settings, with the aim of detecting and preventing cheating during online examinations in Kenyan universities.

1.5 Specific Objectives

- i. To determine the behavioral metric in an online examination at universities in Kenya.
- ii. To scrutinize the techniques used by student for detecting cheating during online exams.
- iii. To propose/use a deep learning model that universities in Kenya can use to detect and prevent cheating during online exams.
- iv. To evaluate the effectiveness in predicting chances of cheating during an online exams by comparing result from the proposed model and the actual situation after the exam has been done.

1.6 Research Questions

- i. What are the behavioral metric in an online examination?
- ii. Which different ways do student use to cheat in an online exams?
- iii. How can we develop a model that can detect cheating during assessment?

- iv. How effective is the model developed in predicting cheating before and after online exams.

1.7 Justification of the Study

Through the use of information technology and especially in the evaluation of student universities have become very competitive and also there growth and increased their numbers despite the limited resources. Online learning systems have made it possible for universities to double their intakes without constructing new physical classrooms. It is complex for universities to give their students online exams due to the increased number of students. So there was need to research on ways that institution can use to ensure there is no cheating during exams and also to increase the validity of the exams done by student.

1.8 Limitations of the Study

The study is subject to several limitations that warrant acknowledgment. Firstly, the research primarily concentrates on universities in Kenya thereby excluding other educational institutions, such as colleges, that administer online examinations. This limitation highlights the necessity to consider a broader range of universities and colleges offering online assessments. Secondly, the generalizability of the study findings to all universities in Kenya is limited. The specific context and unique characteristics of each institution may influence the prevalence and manifestation of cheating behavior, rendering caution in extending the conclusions to other contexts. Lastly, the scope of the study was constrained due to time and resource limitations. This constraint compelled the researcher to narrow the focus, ensuring completion of the research within the available constraints.

1.9 Delimitation of the Study

The delimitation for this study was to use the universities in Kenya and to generalize the findings.

1.10 Significance of the study

The present study holds significant academic value as it endeavors to investigate and comprehend the various forms and manifestations of cheating behavior prevalent in online examination settings within universities in Kenya. By gaining a thorough understanding of the cheating phenomena in this context, the research aims to contribute to the existing body of knowledge on academic integrity, particularly in the digital realm.

One of the primary objectives of this study is to identify the key factors and motivations that contribute to the prevalence of cheating during online examinations. By delving into these factors, the research seeks to shed light on the underlying causes of cheating behavior, enabling educators and policymakers to develop effective strategies to address and mitigate this issue. This understanding of the motivational aspects of cheating can provide valuable insights into designing appropriate interventions that target specific areas for improvement.

Another significant contribution of this research lies in the development of a sophisticated deep learning model tailored specifically for detecting and preventing instances of online cheating in the context of Kenyan universities. By leveraging advanced technology and computational techniques, the proposed model aims to enhance the existing mechanisms used to detect cheating during online exams. This

innovative approach holds promise in improving the accuracy and efficiency of cheating detection, thereby ensuring greater fairness in assessments and examinations.

Furthermore, the study emphasizes the importance of evaluating the effectiveness and efficiency of the proposed deep learning model in mitigating cheating behavior during online examinations. By conducting a rigorous evaluation, the research aims to provide empirical evidence regarding the model's efficacy in detecting and preventing cheating instances. This assessment will enable stakeholders to make informed decisions about the implementation and potential enhancements of the model, ensuring its practical viability and long-term sustainability.

The significance of this study, therefore, transcends its immediate academic contributions. By addressing the pressing issue of cheating in online examination settings, the research aims to foster a culture of integrity and academic honesty among students in Kenyan universities. By discouraging cheating and promoting fairness, the study endeavors to create an environment conducive to genuine learning, where students are motivated to invest in their education and strive for knowledge acquisition rather than resorting to unethical practices.

1.11 Assumptions of the Study

This study operated under a set of underlying assumptions that guided the research investigation. Firstly, it assumed that behavioral metrics play a significant role in the occurrence of cheating behaviors during online examinations within the context of Kenyan universities. The premise was that certain observable patterns of behavior exhibited by students may have a direct influence on their propensity to engage in cheating practices. These behavioral metrics encompassed a range of factors, such as the

frequency and timing of accessing external resources, anomalous patterns of response submission, and interactions with the online examination platform.

The second assumption posited in this study was that students employ diverse techniques and strategies to facilitate their cheating endeavors, and these methods have a discernible impact on the prevalence of cheating during online examinations. It was believed that students' choice and utilization of specific techniques, including but not limited to unauthorized communication with peers, accessing unauthorized materials, or employing advanced technological aids, directly contribute to the overall cheating behavior observed in online examination settings. This assumption acknowledged the existence of a multifaceted landscape of cheating practices, suggesting that the variations in cheating methods and their efficacy may have differential implications for the prevalence of cheating behaviors among students.

Both assumptions were integral to the research objectives and formed the basis for the subsequent development and evaluation of a sophisticated deep learning model aimed at detecting and preventing instances of online cheating in the specific context of Kenyan universities. By recognizing the significance of behavioral metrics and the diverse range of cheating techniques, this study aimed to address the prevailing gaps in the existing literature and provide a comprehensive understanding of cheating behavior in online examination settings. The assumptions served as a framework to guide the research process and the subsequent analysis of data collected, allowing for a systematic investigation of the factors and motivations contributing to cheating behavior and facilitating the development of an effective and efficient deep learning model for cheating detection and prevention.

1.12 Definition of Terms

Artefact	Artifact is a piece of creativity showing human workmanship. It can be tangible or intangible piece of work or modification very unique from a natural object especially. An artefact from a given time period, such as one found in a cave that contains prehistoric relics, or a computer software that solves a particular issue are two examples.
Artificial Intelligence	The creation of computer systems with capabilities of activities that often require human intelligence, such as speech recognition, language translation, and visual perception, is what is referred to as artificial intelligence, or simply AI.
Artificial Neural Networks	An artificial neural network simulates the network of neurons or nodes that comprise the human brain, allowing the computer to learn and make decisions in the same manner humans make decisions. It is a method of programming ordinary computers to behave like interconnected brain cells.
Convolution Neural Networks	Convolutional Neural Network is an algorithm of Deep Learning method that receives input images, assigns relevance in terms of learnable weights and biases to distinct parts of the image, and may extract differences from those

features (ConvNet or CNN).

Deep Learning A subtype of artificial intelligence known as machine learning, known as deep learning, comprises networks that can learn unsupervised from unstructured data or random data. Deep Learning is often referred to as Deep Neural Networking or Deep Learning (Investopedia).

Deep Neural Networks Deep Neural Network (DNN) consist of neurons, synapses, weights, biases, and functions as components just like neural networks of human brain. Between the input and output levels, these construct several layers, resulting in an artificial neural network (ANN).

Machine Learning An area of artificial intelligence is called machine learning (ML). Making machinery capable of mimicking intelligent human behavior is what this technique is all about. Machines with ML capabilities can carry out complex jobs almost as well as people can (expert.ai).

Recurrent Neural Networks Recurrent Neural Network (RNN) is a type of Artificial Neural Network and a Deep Learning technique. This approach is frequently used in natural language processing and speech recognition applications. Recurrent neural

networks recognize the sequential properties of data to forecast the next likely scenario.

TensorFlow TensorFlow, an open-source platform for Machine Learning, accepts inputs as a complex structure of data called Tensor. It uses data flow plots to construct models, enabling designers to create large-scale neural networks with many layers.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The aim of this research was to understand the detection behavioral and prevention of cheating during online examination in Kenya University using Deep learning. Therefore, this chapter begins by reviewing the cheating concept during online exams, detection methods, and behavioral theories related to online cheating, and proposes solution in preventing online cheating. The chapter will complete with a summary of gaps and opportunities identified from the related work, which form the foundation of this study.

2.2 Theoretical framework on Behavioral Cheating Theories

This section reviews the theories related to the study. Theory is a systematic and logical explanation of how and why a particular phenomenon or set of phenomena occur. It provides a framework for understanding and predicting behaviors, events, and outcomes in a particular domain or area of study. (Weick, 2019). The assumptions, principles, and procedures in a theory are based on empirical evidence, logic, and reasoning. Theories are constantly evolving as new data and information become available, and they are used to guide research, inform practice, and shape policy decisions (Corley & Gioia, 2021), Theories tries to describe why a problem occur and which is the best intervention should be taken to solve the problem (Leedy &Ormrod, 2015). In behavioral cheating we have the following theories, theory of motivated cheating, Attribution Theory, model for cheating causation, institutional theory, utility theory, decision-making theory. Academic phenomenon or behavior that can be studied through various theoretical perspectives.

2.2.1 Theory of Motivated Cheating

Motivation can be described as any force that will influence, initiate, guide, and maintain behavior of student during exams (Reeve & Halusic, 2019). In motivated cheating it suggests that individuals may be more likely to cheat when they perceive that the benefits of cheating outweigh the costs, such as when they are under pressure to perform well or when the stakes are high. In this case, a test taker who does not know an answer might be more likely to cheat if they believe that doing so will improve their grade or academic standing. Motivation within academic takes on a multitude of qualities and types including goals, needs, aspirations, drives, affects, values, and interests of student when they are in college or university. Within education institution, motivation theories are usually based on social-cognitive perspectives that highlight students' perceptions of themselves rather than biological drivers. Social-cognitive theories of motivation include constructs such as perceived ability achievement, needs and motives, perceived expectancies and values for an activity, perception of the source of causality, perception of the future, and intention and perceived control. By applying the Theory of Motivated Cheating in combination with data collection, analysis, and proactive measures, the researcher was able to better understand the behavioral metrics of cheating during online exams and take appropriate actions to mitigate academic dishonesty as we use them in development of the model.

2.2.2. Attribution theory

During online assessment, students behave differently depending on the situation and the events or activities that they are executing or doing. According to Fiske and Taylor (2018, p.23), in this theory it just how we can describe different causes of behavior and

also describe how event takes place. Generally, most theories try to show how users of different information explain the events that occur to them on their day to day activities. Many students try to attribute themselves with how parents, colleagues and lecturers will perceive them if they fail and they end up copying exams. Ordinary people will see things differently, once they see things differently they will make explanations differently hence the more reason they will cheat during online exams. The researcher has deliberated in using this theory in the study in assessment evaluation dishonest during exam is a social event and it's perceived different by different stake holders.

2.2.3 Causation Cheating Theory

In this theory we assume that the perceiver and the world analyze everything that may cause some changed in relation to what one is doing. (Grice 2018). It's also clear in the causal theory the perceiver sees and object as per what he wants to see and what it can cause to the perceiver and what it's happening. Most of the students would wish to analyze all what they do in whatever they are doing. Understanding the root causes of online cheating can help educators, administrators, or platform developers implement strategies to prevent cheating effectively. Causation analysis can help determine factors contributing to cheating behavior, such as lack of engagement, difficult assessment conditions, or inadequate deterrence measures.

2.2.4 Institutional theory

Many companies' focuses on the importance of social, political economical system existence. These issues have impact on decision that different people and organization makes which changes the behavior, norms and different activities within the organization. In an organization rules laws when introduced to different people will be stimulated and they change differently. As this changes of rules and laws are repeated

within the organization they became the day to day activities and also many companies make this the usual things to be followed on daily bases within the organization. In online cheating, institutional theory provides a learning environment and how institutional structures impact students' decisions and attitudes toward academic integrity. The theory was important in understanding how academic institution have structured themselves towards ensuring online exams are not compromised. (Franco, Antonio & Franco, Luciane, 2022). Institutional theory can provided valuable context for understanding the factors that contribute to cheating behavior an online examination cheating analysis. The theory was used to understand behavioral metrics, you can better identify and address instances of cheating during online exams while also promoting and understanding of the behaviors when student are doing exams

2.2.5 Utility theory

Utility theory bases its beliefs upon individuals' favorites within the organization different people will be connected to the favorites of what they need to do and what they do on a daily bases. This theory tries to describe how people observe and make decisions of choices depending on what they loves most. In this theory we observe the choices of the individuals and we check on the preferences that they make and how it changes their behavior. The Utility Theory can be used in online exams institutions due to its ability to enhance decision-making of student when they want to cheat, how they optimize resource allocated during exams, how students accommodate individual differences when doing online exams, The theory can also be used to improve feedback mechanisms, manage risks, and create a more engaging and effective assessment experience for both lecturers, invigilators and students. (Carlson Elizabeth. 2020).This theory was mostly for understanding the techniques that are used by student when

cheating during online exams it provided a framework for understanding decision-making but does not condone or endorse cheating behavior during exams.

2.2.6 Decision Making Theory

In this theory we try to describe how people make decision compared to what is supposed to be done or the procedures that are supposed to be followed. The theory describes how different individual make decision as the uncertainty occurs and also as the environment changes. Decision-making theory can provide insights into the factors that influence students' choices to cheat and the strategies institutions can employ to deter academic dishonesty. By integrating decision-making theory into the understanding of online cheating, educational institutions can develop more effective strategies to prevent and detect academic dishonesty, while also promoting a culture of integrity and responsible decision-making among students. (De Andreis, F. 2020). This theory was very critical in gaining a deeper understanding of the motivations and cognitive processes that lead students to cheat during online exams. This understanding informs efforts to prevent cheating, design interventions, and create a culture of academic integrity within educational institutions

2.3 Empirical Review

Empirical research involves the collection and analysis of data through direct observation or experimentation. Empirical review involves examining the various components of an empirical study, such as the research question, the research design, the sampling method, the data collection procedures, and the statistical analyses used, to evaluate the quality and reliability of the study's findings. The goal of empirical review

is to assess the validity and generalizability of the study's conclusions based on the data collected and analyzed.

2.3.1 Behavioral Metric in an Online Examination

Behavioral metrics are the measures that relate to student's behavior and their expected engagement while doing online test. Behavioral metric refers to biometric using behavioral trait of subject such as hand writing, gait, voice characteristic, keystroke dynamics, mouse dynamic and human communication or control behavior when a student is doing exam. When a student is cheating the behavior will definitely change (Yampolskiy & Govindaraju, 2008), Biometric refers to usage of pattern recognition techniques to measure physiological or behavioral characteristic of student when they are doing exams.

Many researches have been conducted and it has been found that the main behavior or background of cheating include Factors that contribute to this behavior have been identified and studied extensively. The pressures from parents and teachers, academic-integration, awareness, moral-capability, gender, age, academic-performance, technology institutional-support, and cultural-influences are all known to be factors that can contribute to cheating behavior. (Guo, 2011; and Canarutto et al., 2010).It is also true that some students may be more focused on the goal of graduation rather than a genuine desire to acquire knowledge. This can lead to a mentality of doing the minimum amount of work required to achieve passing grades. Additionally, the prevalence of a credit and points system can reinforce the idea that success can be achieved through illegitimate means. However, it is important to note that not all students who cheat do so

because they are uninterested in learning or have a negative attitude towards education. There are many complex factors that contribute to cheating behavior, and each case is unique. It is important to address and understand the underlying causes of cheating in order to effectively prevent and deter this behavior. (Mazodier et. al., 2012)

Reasons that they have been cited in the literature is because they have contributed to learners engaging in online cheating behavior. The internal reasons such as lack of responsibility, laziness, lack of respect for academic rules, and low self-esteem can all lead to a lack of motivation to engage in academic work and a greater likelihood to resort to cheating. (Diego, 2017). External reasons such as peer influence, pressure to get passing grades, and technical problems during online exams can also contribute to cheating behavior. Lack of punishment or consequences for cheating can also increase the likelihood that learners will cheat. (Saleh & Meccawy, 2021) It is important to address both the internal and external factors that contribute to cheating behavior in order to effectively prevent and deter this behavior. (Saleh & Meccawy, 2021) Strategies such as promoting academic integrity, educating learners on the consequences of cheating, providing support for learners who may be struggling with academic work, and implementing consequences for cheating can all help to address this issue. (Hosny & Fatima, 2014; Saleh & Meccawy, 2021),

2.3.2 Techniques Used By Student to Cheat During Online Exams

According to McCabe et.al.(2012) there are so many reasons that can make student who are doing online exams cheat. Among the reasons for cheating include lack of enough time to prepare for the exam, forces to get good results, external forces to get a good

work after the study, lack of obligations, poor upbringing, lack of integrity, unpreparedness of the student and when the lectures do not cover the syllabus or teach the relevant content. According to UNESCO, 2003 education lying are the actions that affect the knowledge that is gained in school or also the lying during the process of getting the knowledge

Online cheating is a process that a student finds way to lie during exams using technologies and other connected devices Academic cheating has always been among the major problems in the educational sphere Mostly Online cheating consist student visiting sites during exams, receiving text messages, through text or emails or caring external disk with the content of the test one is doing. Looking at another computer during exam, accessing other files and software during exams, Use of cell phone during exam and communicating openly during exams. Carrying text book and notes to use during online exams (Razan, 2017)

2.3.3 Model to Detect and Prevent Online Cheating

Institutions have implemented various methods to prevent cheating and technology based solutions are one of the most effective ways to do so. Turnitin.com and other plagiarism detection software allow teachers and professors to compare students work against vast database existing academic papers, identifying any instances of plagiarism cheating. Other technologies that institutions use to prevent cheating include proctoring software (Rubin, B., 2017), which monitors students during online exams to ensure that they aren't accessing unauthorized materials or receiving outside help. Some schools have also implemented biometric identification systems to verify the identity of students during exams, preventing students from taking exams for others (Lee, J. W. (2020)

Moral education is another approach to preventing cheating. By teaching students about the importance of academic integrity and the consequences of cheating, schools can help create a culture of honesty and accountability. Some schools have implemented honor codes, in which students pledge to uphold academic honesty and are held accountable by their peers. (Arivazhagan, 2018)

In online assessment identifying or catching student who cheat during exams is giving higher institution headache and a major problem in making full use of this online examining systems. Identifying cheating during online exams is possible through some signs that will be in proposed model. This indicators could be student past results compared to the recent ones, the patterns on how the examiner is responding to answering questions (Kingston,2014) and (Pollack,2017b).Online cheating can also be detected by use of proctoring software, cameras, and IP monitoring. (Langenfeld, T. (2020)

Student can engage in online cheating during exams, preventing the online cheating Rogers (2006) suggested installation of proctoring security programs to the computers being used by student while doing online exams, Proctoring has been used by many university to control the cheating during exam. Other monitoring methods that can be used include, network security methods where university ensure that the network being used by student are secured and monitored Also plagiarism software's and tools can also be used to ensure that evaluation for identifying the unpermitted use of written content by student is identified. Lastly biometric methods can also be used to ensure that student who are registered are the one who are doing exams by also ensuring student login and

always put their videos or scan their identification card before starting to take the exam.(Castrol 2009).

Online cheating can be prevented by use of proctoring technologies which include:
Online ID Authentication-This is where student are provided with username and password that they can use to logging to take online exams This can be done by ensuring student login using their registration number to login in during exams and also ensure that take a photo of the IDs. Secure exam browsers-This a technology where student will not be able to open any other website when they are taking online exams. The computer browsers freezes ones the student logs in to take the online exams. Mobile Phone and Test Leak Prevention-This is possible by ensuring that the student do not get in to exam rooms with mobile phones or any other devices that they can use to search for answers. This can be done physically by searching student before they get into exam room to ensure they don't enter with mobile phone or any other device that is connected to the internet. (S. Kausar 2020)

2.4 Artificial Intelligence

In Artificial intelligence computers makes judgments from large data that was analyzed repeatedly using the appropriate algorithm (Kurt Cagle, 2019). Artificial Intelligence (AI) has been in existence for several decades, and it has evolved over time. AI is a broad field that includes various branches, including Machine Learning (ML), Deep Learning (DL), Natural Language Processing (NLP), Robotics, Expert Systems, and Fuzzy Logic. Machine Learning involves the development of algorithms that enable computer systems to learn from large data without being explicitly programmed (Wang 2021) This branch of AI has been instrumental in the development of various

applications, including image and speech recognition, recommendation systems, and fraud detection. Deep Learning is a subfield of Machine Learning that involves the use of neural networks to model complex patterns in data. This branch of AI has been critical in achieving breakthroughs in computer vision, speech recognition, and natural language processing (Marreiros,2022). Natural Language Processing is a branch of AI that deals with the interaction between computers and humans using natural language. This technology has been instrumental in the development of chat bots, virtual assistants, and language translation systems. Robotics involves the development of intelligent machines that can perform various tasks autonomously. This branch of AI has been critical in the development of robots for manufacturing, healthcare, and exploration. (Malhotra 2022) Expert Systems are computer programs that can simulate the decision-making ability of a human expert in a particular domain. This branch of AI has been instrumental in the development of decision support systems for various industries. Fuzzy Logic is a branch of AI that deals with reasoning that is approximate rather than precise. This technology has been critical in the development of control systems for various applications, including industrial automation, home appliances, and vehicles. (Zulaikha 2019)

2.4.1 Deep Learning

Another Artificial intelligence is deep learning method that uses neural network architecture where different layers of processing unit is used in analyzing large volumes of images in recognition and natural processing in business and in different industries. The algorithm has gained popularity in analyzing large volumes of data in the whole world where different fields use it. (Zaidi,2022) According to Ethen 2019, Deep learning

can be categorized into three main categories: Supervised Learning: In supervised learning, the deep learning model is trained on labeled data. The input data and are given to the model, and the model learns to map the input data to the output data. The model is then tested on new input data to predict the corresponding output. Unsupervised Learning: In unsupervised learning, the deep learning model data is trained on unlabeled data. The model learns to identify patterns and relationships in the input data without any explicit feedback. The aim is to discover hidden structures or features in the data that is later used to perform tasks such as clustering, dimensionality reduction, and anomaly detection. Reinforcement Learning: In reinforcement learning, the deep learning model learns to take actions in an environment to maximize a reward signal. The model interacts with the environment and receives feedback in the form of rewards or penalties based on the actions it takes. The aim is to learn a policy that can maximize the cumulative reward over a sequence of actions. Reinforcement learning has been successfully applied to various domains such as robotics, game playing, and autonomous driving and proved to be very efficient. (Gao, 2019). Deep learning can be used to detect how student cheat during online examination. It is an increasingly popular approach for many applications, including human activity recognition, image recognition, natural language processing, and more. To detect cheating during an online examination deep learning algorithms can be trained on data from previous exams to identify patterns of behavior associated with cheating, such as copying answers from another source or accessing unauthorized materials. Once trained, the algorithm can analyze data from current exams in real-time to identify suspicious behavior and flag potential cases of cheating for review by a human proctor. Deep learning algorithms are well suited for this

task because they are able to automatically extract relevant features from the data and learn complex patterns without the need for explicit programming. This makes them highly effective at detecting cheating even when the techniques used by cheaters are novel or sophisticated (Yulita 2017).

According to Nguyen 2019,) CNN is a popular deep learning algorithm used for object detection and recognition, including in the context of cheating detection during online assessments. MobileNet is a specific CNN architecture that has some unique design features, such as matching the thickness of the convolution filter to the input and using depth-wise and point-wise convolution to enable faster and more accurate training. MobileNetV2 is an updated version of MobileNet that incorporates additional features like linear bottlenecks and shortcut connections between bottlenecks to further improve accuracy with fewer parameters. The input to the application in the context of cheating detection during online assessments could be a video of participant recordings. (Ayachi, 2021).

MobileNet is a popular CNN architecture that is designed to be computationally efficient and well suited for mobile and embedded devices. Its unique design allows it to achieve high accuracy while using fewer parameters than other architectures. CNN is commonly used for image detection and classification tasks, but can also be applied to other types of data, such as videos. In the context of detecting cheating during online assessments, MobileNet was potentially be used to analyze video feeds and detect any suspicious activity, such as the presence of unauthorized materials or the involvement of third parties. (Krizhevsky, Sutskever and Hinton, 2012).

2.4.2 Deep learning stages

1. Collecting Data

In deep learning, the most important stage is collection of data that is reliable that the machine-learning model used to find the patterns. The video data was recorded from webcams of several student taking online tests was utilized in this research. The dataset was obtained from Michigan State University's Computer Vision Lab and is available at the following address: <http://cvlab.cse.msu.edu/project-OEP.html> .(Downloaded on 5/4/2023).

2. Preparing the data/Data Processing

Data preprocessing is an essential step in machine learning to ensure that the data is ready for analysis and modeling. It involves a range of techniques, including cleaning, normalization, transformation, and augmentation. In behavioral cheating detection and prevention during online exams, data augmentation was useful technique to generate more diverse and balanced data. In this research data augmentation employed transformation to generate numerous copies of frames with varying variances. By applying transformations such as rotating, shifting, flipping, and zooming, the researcher created variations of the original data that can improve the accuracy and robustness of your machine learning model. This technique did not change the target class but it gave a new viewpoint on catching objects in frames. The issue of imbalanced data, where some classes may have fewer examples than others was sorted by generating additional samples for these classes, to balance the dataset to prevent the machine learning model from being biased towards the majority class. This approach expended the data and also introduced some variety allowing the model to generate better classifiable predictions on

previously unknown data. The researcher trained the data on new, slightly modified frames and the model become more resilient. The researcher used the following augmentation approach.

Random rotation: This is an augmentation approach that allows the model to be object orientation invariant. Data augmentation in the form of rotation range allows for random rotation of frames across any degree between 0 and 360. Some pixels migrate outside the picture when the frame rotates, leaving empty spaces that could be filled.

Random shifts: This is a method used to compensate for the fact that objects are not constantly in the center of the frame. This issue can be resolved by moving the frame pixels horizontally or vertically. The height shift range is used for vertical frame shifts, whereas the width shift range is utilized for horizontal shifts.

Random Flip: One of the augmentation strategies used to flip frames is random flip. The horizontal flip is used to flip the frame horizontally, and the vertical flip is used to flip the frame vertically. However, in order for the model to generate classification results based on the class, the augmentation strategy utilizing random flip must have a frame that is symmetrical with the item in the original frame.

Random Zoom: This is an augmentation method that uses a zoom range to randomly increase or shrink the picture size.

3. Splitting the data

The process of separating data into categories is known as data splitting. Data splitting is a crucial step in machine learning to evaluate the performance of the model and prevent overfitting. The training data was derived from the subject frames

4. Deep Learning Modelling

In order to develop a robust model capable of accurately detecting instances of cheating during online examinations, several deep learning models were utilized within the research study. These models encompassed a range of architectures, including DenseNet, MobileNet, ResNets, and Convolutional Neural Networks (CNN). Each model was trained using the preprocessed dataset, which consisted of the labeled JPEG images classified as Cheating, Trying, and Nocheating.

The implementation of these deep learning models was conducted utilizing prominent deep learning frameworks such as TensorFlow and PyTorch. These frameworks provided comprehensive libraries and tools for building, training, and evaluating the models. The chosen deep learning architectures offered distinct advantages and features, allowing for an exploration of various approaches in detecting cheating behaviors during online examinations.

To train the deep learning models, the preprocessed dataset was used as the input. The models learned from the labeled images, identifying patterns and features indicative of cheating, trying, and no-cheating behaviors. The training process involved iteratively adjusting the model's parameters to minimize the loss function, enabling the models to capture and learn the complex relationships within the data.

The training of the deep learning models aimed to optimize their performance in accurately identifying cheating instances during online examinations. This process involved training the models on a substantial portion of the dataset, allowing them to learn and generalize from the patterns observed in the labeled images. By leveraging the

deep learning architectures and the preprocessed dataset, the models were equipped to make predictions and classify unseen images as either cheating, trying, or no-cheating.

Following the training phase, the models underwent rigorous evaluation using appropriate metrics. The evaluation metrics employed in this research study included a MP (average precision), accuracy, and loss. These metrics provided insights into the models' performance, assessing their ability to accurately detect and differentiate cheating behaviors during online examinations. By employing a comprehensive evaluation approach, the research aimed to quantify and compare the efficacy of the deep learning models in curbing cheating behavior within the context of Kenyan public universities.

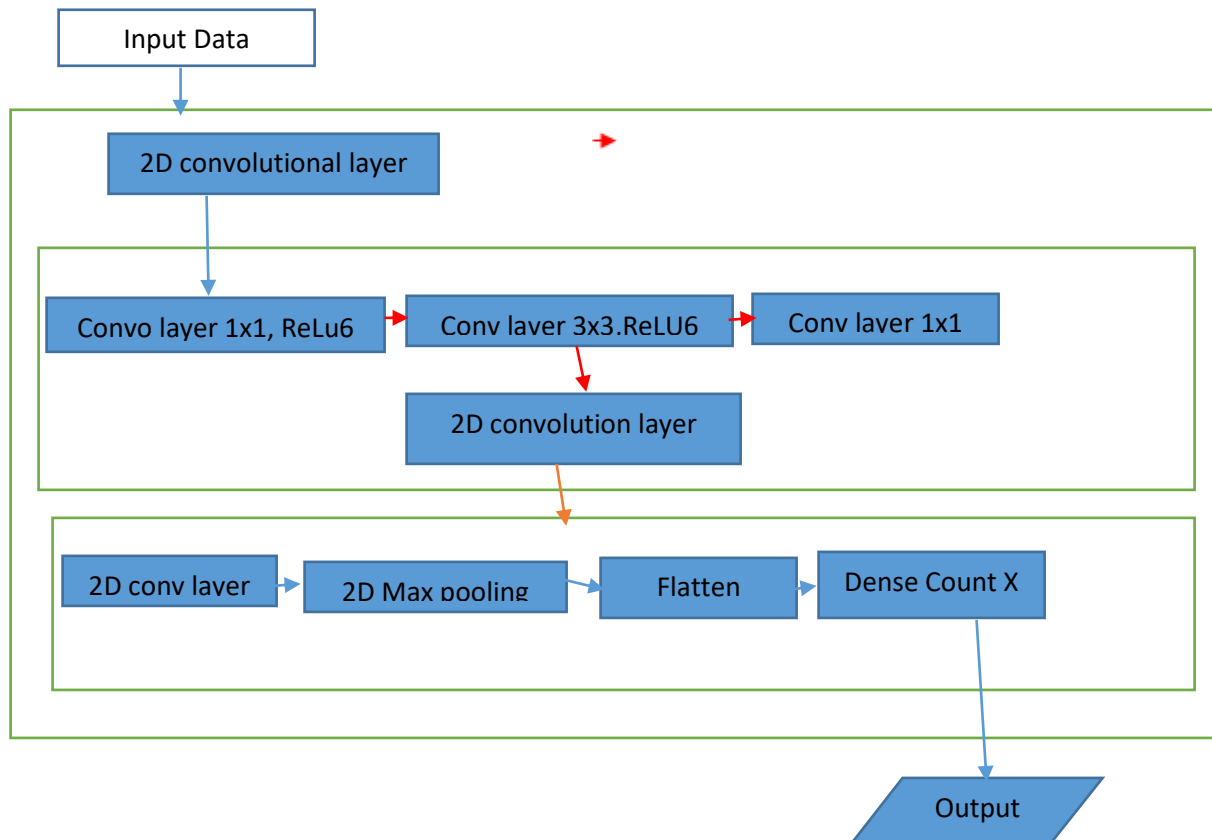
The utilization of multiple deep learning models allowed for a thorough exploration of different architectural designs and approaches to detecting cheating behaviors. This diversity in model selection enabled a comprehensive assessment of the deep learning techniques in addressing the research questions and objectives. The findings from the evaluation of the deep learning models provided valuable insights into the effectiveness of each architecture, facilitating a better understanding of their respective strengths and limitations in detecting cheating during online examinations.

This researcher used Convolutional Neural Network (CNN) technique with the MobileNetV2 architecture to create a model. CNNs are a popular deep learning approach for image classification and recognition tasks, while MobileNetV2 is a specific type of CNN architecture that is optimized for mobile devices with limited computational resources. The researcher created a model utilizing the CNN technique with the

mobileNetV2 architecture. It contains fewer parameters than previous models in use.

The figure 2.0 below shows a simple architecture that was used in the sample.

Figure 2.0 Simple mobilenetV2 model arch



5. Evaluating the model.

The evaluation of the proposed deep learning models encompassed the utilization of various performance metrics, including MAP (mean average precision), accuracy, and loss. These metrics served as quantitative measures of the models' effectiveness in accurately classifying and discerning different behavioral patterns linked to cheating occurrences during online examinations. The evaluation procedure entailed the division of the dataset into distinct training and testing sets, with the testing set employed to gauge the models' performance on previously unseen data.

During the evaluation process, the trained deep learning models were applied to the testing set to classify the unseen instances and generate predictions pertaining to the presence of cheating, trying, or no-cheating behaviors. These predictions were then juxtaposed against the ground truth labels to ascertain the accuracy of the models' classifications. The accuracy metric served to quantify the proportion of correctly classified instances, offering an overall indication of the models' performance in detecting cheating during online examinations.

In addition to accuracy, the evaluation process incorporated the MAP metric, which assessed the mean average precision. This metric provided a more nuanced evaluation by considering the likelihood of misclassifying instances across all classes. A higher MAP value indicated a superior accuracy in classifying the different behavioral patterns associated with cheating, trying, and no-cheating.

Furthermore, the evaluation encompassed the analysis of the loss metric, which gauged the discrepancy between the predicted and actual labels. The minimization of the loss function during the training phase signified that the models effectively assimilated the

underlying patterns and features indicative of cheating behaviors. The analysis of loss yielded valuable insights into the models' convergence, optimization, and training efficacy, thereby facilitating an assessment of their overall performance.

Through the incorporation of multiple performance metrics, the evaluation process provided a comprehensive appraisal of the proposed deep learning models' ability to detect cheating behaviors during online examinations. The amalgamation of accuracy, MAP, and loss metrics enabled a thorough evaluation of the models' classification capabilities, thereby elucidating their strengths and limitations. The findings derived from the model evaluation process contributed to the broader comprehension of the models' performance and their potential as effective tools in mitigating cheating behavior within the context of Kenyan public universities.

6 Data Analysis

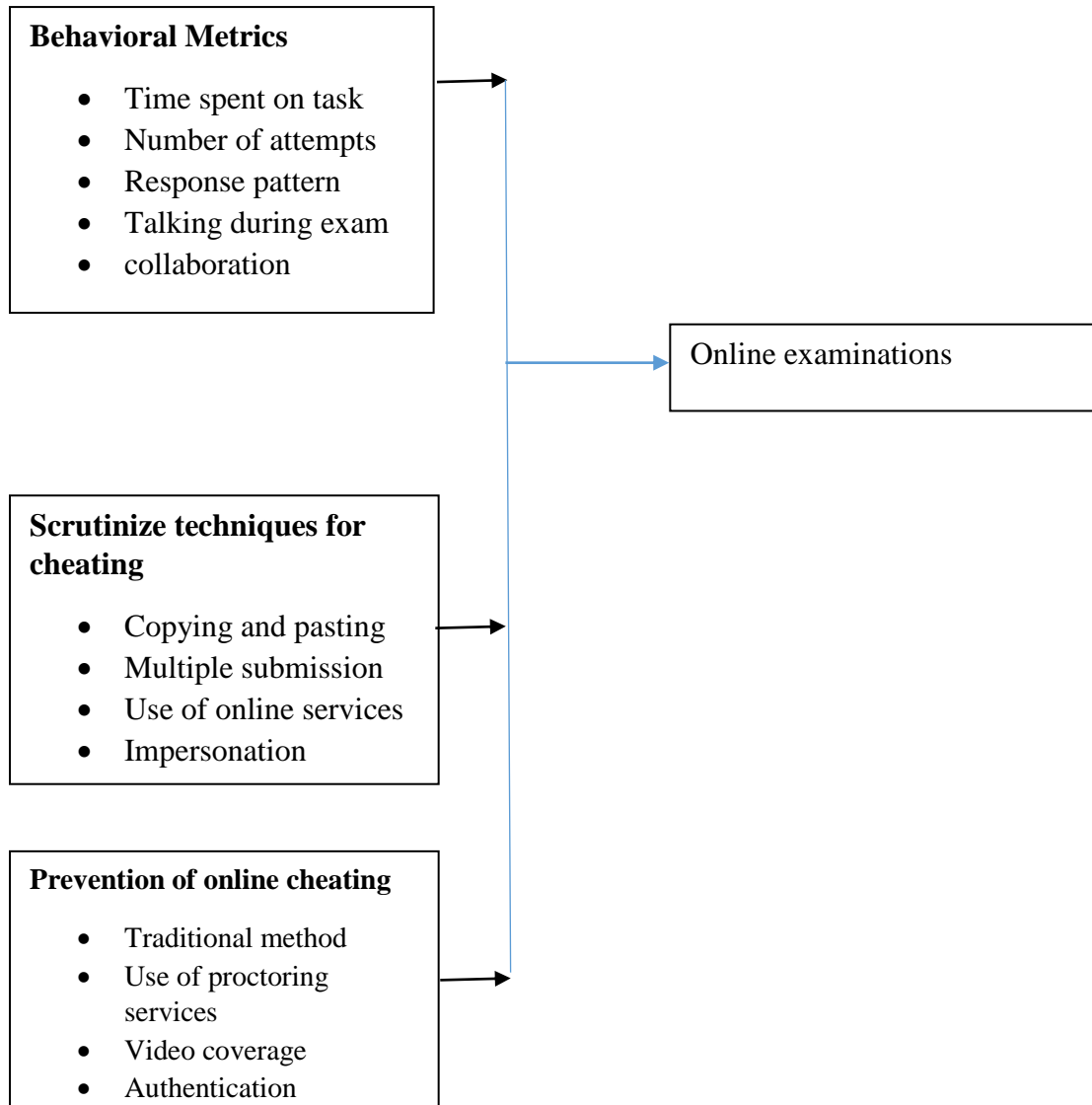
Data analysis included the phases of cleaning, transforming, and modeling data to get useful facts that can be used by universities in making good and appropriate way forward for the online learning and exams. The study detected the behavior of student during online exams. The objective of analyzing data was to get facts that are critical in redesign the online assessment as per the output of information from the study. After the data has been downloaded from the internet, the researcher used data analysis tools like MobileNetV2, Keras, tensorflow, pytorch, mat lab to analyze the data and present the output.

2.5. Conceptual Frame Work

In this section, we break and convert the research study ideas into common meanings to develop an agreement among the variables (Sequeira, 2014). Academic fraud can occur in various settings, including online environments. The conceptual framework presented in this text highlights several factors that contribute to academic fraud in online settings. These factors include: Behavioral Metrics: Online environments often provide data that can be analyzed to detect cheating behaviors. Such data includes timing, frequency, and duration of engagement, mouse clicks, keystrokes, and other behavioral patterns that indicate unusual or suspicious activity. Techniques of Cheating: There are various techniques of cheating in online environments, including plagiarism, unauthorized collaboration, and impersonation. These techniques can be facilitated by online tools such as copy-paste, messaging apps, and screen-sharing. Prevention of Online Cheating: There are several approaches to prevent academic fraud in online settings. These include using plagiarism detection software, designing assessments that are difficult to cheat on, establishing clear rules and guidelines for online assessments, and using online proctoring tools that monitor student behavior during exams.

Figure 2.1 Conceptual Framework

Figure 2.1:
Conceptual frame work



Behavioral Metric Variables

The behavioral variable in this research was timing metrics in the context of detecting cheating during online exams involve monitoring students' behavior to identify irregularities in the time they spend on various exam-related activities. Response time analysis evaluates how quickly students answer questions, with rapid responses or

consistent patterns across questions potentially signaling cheating attempts. Additionally, the examination of the time students spend per question can reveal anomalies, such as abnormally short or long durations, which may suggest cheating behavior. These timing metrics are critical components of a multifaceted approach to maintaining academic integrity in online education, allowing institutions to flag suspicious behavior and take appropriate actions to uphold fairness and honesty in assessments.

Variables For Scrutinize Techniques for Cheating

During in-person exams, students often resort to subtle tactics like glancing at neighboring papers, hoping to copy answers or gain insights from their peers, Multiple submission of the work that they have done, use of online services when doing exams and lastly Impersonation where student pays someone to do exam on their behave.

Variables Prevention of Online Cheating

Variables that can be used to determine the prevention of cheating during online exams includes Preventing online cheating through traditional means relies on non-technological approaches, like conducting exams in physical classrooms under teacher supervision or using printed question papers. Online proctoring services are specialized tools that leverage technology, including webcams, screen sharing, and audio monitoring, to actively monitor and prevent cheating during online assessments. Online proctoring services are specialized tools that leverage technology, including webcams, screen sharing, and audio monitoring, to actively monitor and prevent cheating during online assessments. Video coverage entails recording students during online assessments, either as part of a proctoring service or as an independent practice.

Authentication methods confirm the identity of online exam-takers, preventing unauthorized individuals from taking tests on behalf of students.

2.6 Knowledge Gap

The research revealed several significant knowledge gaps in the study of cheating prevention during online examinations in Kenyan universities using deep learning. Firstly, there is a lack of practical examples or case studies showcasing the real-world implementation of deep learning models for cheating prevention, leaving a critical void in understanding their feasibility and challenges within the Kenyan context. Additionally, the effectiveness and accuracy of these deep learning algorithms in detecting cheating behaviors during online exams remain unaddressed, warranting research that assesses their performance in real exam scenarios. A comparative analysis of various cheating prevention methods and their integration with institutional policies is warranted to identify the most suitable strategies for Kenyan universities. Addressing these knowledge gaps would provide valuable insights for educators, administrators, and policymakers in the region

2.7 Literature Review Summary

In summary, the research explores various behavioral theories related to cheating during online exams and discusses how AI and Deep Learning can be employed to detect and prevent cheating in Kenyan universities. The aim is to enhance the integrity of online examinations and promote academic honesty

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter provides an in-depth account of the methodology utilized to investigate and understand learner behaviors in the context of online assessment settings, specifically focusing on the detection of cheating during online examinations at Kenyan universities. The primary objective of this study is to employ a deep learning approach in order to achieve the aforementioned task. Furthermore, the techniques employed for dataset processing are elucidated, and the sampling technique utilized in the study is explicated.

The aim of this study is to gain insights into the distinctive behavioral patterns and strategies employed by students who engage in cheating during online examinations in Kenyan universities. Additionally, the study seeks to delve into the underlying factors and motivations that drive students towards engaging in such dishonest practices. To accomplish these goals, deep learning techniques are harnessed to construct a robust model capable of accurately detecting and preventing instances of online cheating. The efficacy of the proposed deep learning model in curbing cheating behavior during online examinations at Kenyan universities is a key aspect to be evaluated. To achieve this, several prominent deep learning models, including DenseNet, MobileNet, ResNets, and CNN, are built and utilized in the study. These models have been selected due to their effectiveness in handling image classification tasks and their ability to capture complex patterns and features inherent in visual data. The performance of the constructed models is evaluated using essential metrics, namely AMP (average mean precision), accuracy,

and loss, which collectively provide a comprehensive assessment of the models' effectiveness in cheating detection. Furthermore, the dataset used in this study undergoes a series of preprocessing steps, encompassing the conversion of movies in WAVI format to MP4, extraction of MP4 movies to JPEG images, classification of JPEG images into the categories of Cheating, Trying, and Nocheating, renaming of JPEG images for organizational purposes, and extraction of relevant features from the images. To ensure representative and unbiased data, a suitable sampling technique is employed and thoroughly explained.

3.2 Research Design

In order to investigate and gain a comprehensive understanding of learner behaviors in online assessment settings, as well as develop a deep learning model for the detection of cheating during online examinations in universities, a quantitative research design was adopted. This design facilitated the systematic collection and analysis of behavioral data from a carefully selected sample of students actively participating in online examinations.

The quantitative research design enabled the study to utilize numerical data to explore and quantify various aspects of learner behaviors related to cheating during online examinations. By employing this design, the research aimed to establish clear relationships and patterns within the collected data, enabling a more objective analysis and interpretation of the findings. The design allowed for the implementation of statistical analysis techniques, facilitating the examination of correlations, trends, and statistical significance pertaining to learner behaviors and the development of the deep learning model.

Through the application of the quantitative research design, the study sought to provide empirical evidence and insights into the prevalence and characteristics of cheating behaviors during online examinations in Kenyan universities. This research design also allowed for the formulation of generalizable conclusions based on the representative sample, contributing to the existing body of knowledge on the topic and informing future interventions and preventive measures against online cheating in academic settings

Experiment quantitative research design was utilized since the research was using the scientific approach. It is also considered one of the most rigorous methods for studying cause-and-effect relationships, as it allows for the manipulation of variables and control over potential confounding factors.

3.3 Location of Study

The research study was carried out within the premises of universities situated in Kenya. The selection of these universities was predicated upon their possession of online examination systems and their voluntary participation in the study. The chosen universities served as an appropriate setting for the investigation, encompassing a varied composition of students and academic disciplines, thereby facilitating a comprehensive representation of the target population.

The utilization of various universities in Kenya as the research setting ensured a broader scope for data collection and analysis. This approach allowed for the examination of learner behaviors in diverse educational environments, enhancing the generalizability of the study findings. The selected universities' willingness to participate in the research affirmed their commitment to addressing the issue of cheating during online examinations, which further fostered a collaborative and conducive research setting.

3.4 Target Population

The participants in this research comprised undergraduate and postgraduate students enrolled in the chosen universities. A purposive sampling technique was employed to select participants who were actively involved in online examinations during the designated data collection period. The purposive sampling approach facilitated the deliberate selection of individuals with relevant experiences and behaviors related to online examination settings and potential cheating practices. This method ensured that the participants represented the target population of interest, maximizing the relevance and applicability of the study findings.

The sample size for this study was determined by employing the principle of saturation. Saturation refers to the point at which data collection ceases to yield new information or patterns, indicating that data saturation has been achieved. In line with this principle, data collection continued until no further novel insights or discernible patterns emerged from the analysis. This approach ensured that the sample size was sufficient to capture a comprehensive understanding of learner behaviors during online examinations, while avoiding unnecessary data redundancy.

The use of undergraduate and postgraduate students from diverse academic backgrounds and educational levels within the selected universities allowed for a multifaceted exploration of the research topic. Their involvement in online examinations provided valuable insights into the dynamics of cheating behaviors within the specific context of Kenyan universities. The inclusion of participants from different academic disciplines added depth and breadth to the study's findings, enabling a more nuanced understanding of cheating behaviors across various fields of study.

3.5 Data Collection

3.5.1 Instrumentation

The central data collection instrument employed in this research was a tailor-made online examination monitoring system. This system was specifically developed to capture and document a multitude of behavioral indicators and activities demonstrated by students throughout online examinations. The design of the monitoring system encompassed the monitoring and recording of essential parameters, such as keystrokes, mouse movements, duration of engagement with each question, and the extent of external resource utilization.

By utilizing the custom-built online examination monitoring system, the research aimed to comprehensively gather quantitative data on student behaviors during online examinations. The system's ability to track and record specific behavioral parameters provided valuable insights into potential cheating tendencies and patterns. Through the collection of these diverse indicators, the monitoring system facilitated a holistic assessment of student engagement and behaviors during online examinations, contributing to the overall understanding of cheating occurrences and aiding in the subsequent analysis and detection processes.

Data Set

A dataset is a structured collection of data that consists of a set of individual data points or examples. Each data point within a dataset represents a unique piece of information or observation. Datasets are fundamental components in data analysis, machine learning, statistics, and various research fields. The dataset that was used for this research were

movies that were downloaded from Michigan university website for undergraduate and post graduate student doing online exams. The size of the dataset that was used is about 20GB. The dataset was a collection of images and videos in form of Wavi format.

3.5.2 Data Collection Procedure

The data collection procedure for this study was conducted with meticulous adherence to ethical guidelines. Before the commencement of online examinations, informed consent was obtained from all participants, ensuring their voluntary participation and informed understanding of the study's objectives, procedures, and their rights as research participants. The informed consent process provided a transparent explanation of the study's purpose, potential risks and benefits, confidentiality measures, and the participants' right to withdraw from the study at any time without repercussion.

Subsequently, the online examination monitoring system was installed on the participants' computers prior to the scheduled online examinations. This step facilitated the collection of real-time behavioral data throughout the examination process. The installation process was carried out systematically, ensuring compatibility with participants' computer systems and maintaining the integrity of their personal devices. The online examination monitoring system allowed for the unobtrusive and automatic capture of various behavioral parameters and activities exhibited by the participants during their online examinations, ensuring a comprehensive collection of relevant data for subsequent analysis and interpretation. The data collection procedure was implemented with utmost care to protect participant privacy and confidentiality, while upholding the principles of informed consent and ethical research conduct.

3.6 Data Preprocessing

To ensure the quality and suitability of the collected data for subsequent analysis, a series of preprocessing steps were undertaken. Initially, the movies recorded in WAVI format were transformed into the more widely compatible MP4 format. This conversion facilitated smoother processing and compatibility with the deep learning models employed in the study.

Subsequently, the MP4 movies were subjected to frame extraction, resulting in a comprehensive collection of individual JPEG images. These images served as the fundamental units for subsequent behavioral analysis, allowing for a more granular examination of student behaviors during online examinations.

The extracted JPEG images were then meticulously classified into distinct categories: "Cheating," "Trying," and "Nocheating." This classification process involved the expertise and insights of trained annotators who closely observed and categorized the images based on the exhibited student behaviors.

To enhance organization and facilitate efficient tracking during subsequent analysis, the classified JPEG images were systematically renamed. Descriptive labels denoting the respective categories ("Cheating," "Trying," and "Nocheating") were assigned to the images, accompanied by indexes that maintained their integrity and ensured a coherent structure.

In addition to organizing the images, the next step in the data preprocessing involved the extraction of relevant features from the JPEG images. These features served as representative representations of the underlying behavioral patterns exhibited during

online examinations. By extracting pertinent features, the study aimed to capture key elements and characteristics of the observed behaviors, which would subsequently serve as inputs to the deep learning models employed in the research.

The extraction of meaningful features from the JPEG images enabled a more concise and informative representation of the behavioral data, facilitating the subsequent analysis and detection of cheating behaviors during online examinations. This preprocessing step ensured that the data was appropriately prepared for the subsequent application of the deep learning models, allowing for the identification of critical patterns and features that could aid in the accurate detection and prevention of cheating instances.

To capture and represent the behavioral patterns manifested during online examinations, pertinent features were extracted from the JPEG images. These extracted features, serving as inputs to the deep learning models, were transformed into a CSV (Comma-Separated Values) file format. Subsequently, the CSV file was divided into training and testing datasets utilizing Python programming language on the Jupyter interface. This division facilitated the separation of data into distinct sets for model training and evaluation, ensuring an unbiased and rigorous assessment of the deep learning models' performance in detecting cheating instances during online examinations.

Figure 3.1

Conversion of Wav to MP4

```
# Define the input folder containing AVI files
input_folder = '/content/drive/MyDrive/Dataset/Original/'

# Define the output folder to save the converted MP4 files
output_folder = '/content/drive/MyDrive/Dataset/New/'

# Get a list of AVI files in the input folder
avi_files = [f for f in os.listdir(input_folder) if f.endswith('.avi')]

# Convert AVI files to MP4 files
for avi_file in avi_files:
    # Construct input and output file paths
    input_file = os.path.join(input_folder, avi_file)
    output_file = os.path.join(output_folder, os.path.splitext(avi_file)[0] + '.mp4')

    # Run ffmpeg command to perform the conversion
    subprocess.run(['ffmpeg', '-i', input_file, output_file])

print('Conversion completed.')
```

The figure 3.1 is a Python script that performs a conversion of AVI video files to MP4 format using the FFmpeg tool. The script begins by defining the input folder path where the original AVI files are located and the output folder path where the converted MP4 files will be saved. Then, it retrieves a list of AVI files from the input folder using a list comprehension. Next, it iterates over each AVI file in the list and constructs the input and output file paths by combining the folder paths and file names. Finally, it uses the subprocess module to run the FFmpeg command-line tool, passing the input and output file paths as arguments, which performs the actual conversion. Once the conversion is completed for all AVI files, it prints a message indicating the completion of the process.

Figure 3.2

Python script converts MP4 to JPEG

```
# Get a list of all the MP4 files in the input folder
mp4_files = [f for f in os.listdir(inFolder) if f.endswith('.mp4')]

# Loop through each MP4 file and convert it to JPEG
for mp4_file in mp4_files:
    # Get the path of the MP4 file
    mp4_path = os.path.join(inFolder, mp4_file)

    # Read the MP4 file using OpenCV
    video = cv2.VideoCapture(mp4_path)

    # Set the output JPEG file path
    outFile = os.path.splitext(mp4_file)[0] + '.jpeg'
    outPath = os.path.join(outFolder, outFile)

    # Read and save each frame as a JPEG image
    success, image = video.read()
    count = 0
    while success:
        # Save the frame as a JPEG image
        cv2.imwrite(outPath, image)

        # Read the next frame
        success, image = video.read()

        # Update the output JPEG file path for the next frame
        count += 1
        outPath = os.path.join(outFolder, f'{os.path.splitext(mp4_file)[0]}_{count}.jpeg')

    # Release the video object
    video.release()
```

The given code in figure 3.2 is a Python script that converts MP4 video files to a series of JPEG images. The script utilizes the OpenCV library for reading and manipulating video files. Initially, the paths for the input and output folders are set. The script checks if the output folder exists, and if not, it creates the folder. Then, it retrieves a list of MP4 files from the input folder. The code iterates through each MP4 file and reads it using OpenCV's VideoCapture function. It sets the output file path for the JPEG images and begins reading and saving each frame of the video as a JPEG image. The process continues until all frames of the video have been processed. The output JPEG images are saved in the specified output folder, with each image numbered sequentially. Finally, the video object is released to free up system resources

Figure 3.3

Classification of JPEG Images

```
import os
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
# Set the path to your dataset directory
dataset_dir = "/content/drive/MyDrive/Dataset/JpgFiles/"

# Check if the directory exists
if os.path.exists(dataset_dir):
    # List files in the directory
    files = os.listdir(dataset_dir)
    print("Files in the directory:")
    for file in files:
        print(file)
else:
    print("Directory does not exist.")
# Set the output directory for the generated class labels
output_dir = "/content/drive/MyDrive/NewDataSet/"

# Create output directory if it doesn't exist
os.makedirs(output_dir, exist_ok=True)
# Iterate over the JPEG images in the dataset directory
for filename in os.listdir(dataset_dir):
    if filename.endswith(".jpg") or filename.endswith(".jpeg"):
        # Read the image
        image_path = os.path.join(dataset_dir, filename)
        image = cv2.imread(image_path)
        # Display the image and prompt for class input
        class_label = input("Enter class label (cheating/trying/nocheating): ")
        # Generate class-specific directory if it doesn't exist
        class_dir = os.path.join(output_dir, class_label)
        os.makedirs(class_dir, exist_ok=True)
        # Move the image to the corresponding class directory
        new_path = os.path.join(class_dir, filename)
        os.rename(image_path, new_path)
        # Close the image display window
        cv2.destroyAllWindows()
print("Class labeling completed!")
```

The figure 3.3 is code that performs class labeling on a dataset of JPEG images. It first sets the path to the dataset directory and checks if it exists. If the directory exists, it lists the files in the directory. Then, it sets the output directory for the generated class labels and creates the directory if it doesn't exist. Next, it iterates over the JPEG images in the dataset directory. For each image, it reads the image using OpenCV, displays the image,

and prompts the user to input a class label (cheating/trying/no cheating). It generates a class-specific directory within the output directory if it doesn't exist and moves the image to the corresponding class directory by renaming its path. Finally, it closes the image display window and prints a message indicating that the class labeling process is completed.

3.5.1 Renaming of Jpeg Files

Figure 3.4

Renaming of JPEG Files

```
import os

def rename_files(root_folder):
    index = 1

    for root, dirs, files in os.walk(root_folder):
        for file in files:
            original_path = os.path.join(root, file)
            folder_name = os.path.basename(root)
            new_filename = f"{folder_name}_{index}{os.path.splitext(file)[1]}"
            new_path = os.path.join(root, new_filename)

            os.rename(original_path, new_path)
            index += 1

# Example usage:
folder_path = '/content/drive/MyDrive/NewDataSet/'
rename_files(folder_path)
```

Figure 3.4 depicts a function called rename files that renames the files within a specified root folder. The function starts by initializing an index variable to 1. It then uses the os.walk function to iterate through all the files in the root folder and its subdirectories. For each file, it constructs the original file path, extracts the folder name from the current root, generates a new filename by appending the folder name, index, and file extension,

and constructs the new path using the root and new filename. The code uses the `os.rename` function to rename the file by moving it from the original path to the new path. After renaming each file, the index is incremented. To use the function, an example usage is shown where a folder path is specified, and the `rename_files` function is called with that path as an argument. This will rename all the files within the specified folder and its subdirectories according to the given naming convention.

3.5.2 Extracting Features from Jpeg Images

The study applied VGG16 model to a set of JPEG images organized into subfolders, extracts their features, and saves the features and corresponding labels in a CSV file for further analysis or machine learning tasks. This is shown in figure 3.6

Figure 3.5

Extracting Features from JPEG Images

```
main_dir = '/content/drive/MyDrive/NewDataSet/'
# Define the path to the output CSV file
output_csv = '/content/drive/MyDrive/csvFiles/Muchangi.csv'
# Load the pre-trained VGG16 model
base_model = VGG16(weights='imagenet', include_top=False, pooling='avg')
# Create empty lists to store the features and labels
features = []
labels = []
# Process each subfolder and its images
for subfolder in os.listdir(main_dir):
    subfolder_path = os.path.join(main_dir, subfolder)
    if os.path.isdir(subfolder_path):
        # Get the list of image file names in the subfolder
        image_files = os.listdir(subfolder_path)
        # Process each image
        for image_file in image_files:
            image_path = os.path.join(subfolder_path, image_file)
            # Load the image
            image = cv2.imread(image_path)
            if image is not None:
                # Resize the image to the input shape expected by the VGG16 model
                resized_image = cv2.resize(image, (224, 224))
                # Preprocess the image
                preprocessed_image = preprocess_input(resized_image)
                # Expand dimensions to create a batch of size 1
                preprocessed_image = np.expand_dims(preprocessed_image, axis=0)
                # Extract features from the pre-trained model
                features.append(base_model.predict(preprocessed_image).flatten())
                # Extract the label from the subfolder name
                labels.append(subfolder)
# Convert the lists to NumPy arrays

# Create a DataFrame to store the features and labels
df = pd.DataFrame(features)
df['label'] = labels
# Save the DataFrame to a CSV file
df.to_csv(output_csv, index=False)
```

Figure 3.5 shows a Python script that performs image feature extraction using the VGG16 convolutional neural network (CNN) model and saves the extracted features along with corresponding labels to a CSV file. The code first imports the required libraries: pandas for data manipulation, numpy for numerical computations, os for file and directory operations, and cv2 for image processing using OpenCV. It also imports the VGG16 model from the TensorFlow Keras library. The main directory path and output CSV file path are defined. The pre-trained VGG16 model is loaded with weights from the ImageNet dataset, and the average pooling layer is used for feature extraction. Empty lists are created to store the extracted features and corresponding labels.

The code then iterates over each subfolder within the main directory. For each subfolder, it retrieves the list of image file names and processes each image individually. The image is loaded using OpenCV, resized to the input shape expected by the VGG16 model (224x224 pixels), and preprocessed using the preprocess_input function from the VGG16 module. The preprocessed image is expanded to create a batch of size 1 and passed through the VGG16 model to extract the features. The flattened feature vector is appended to the features list, and the label is extracted from the subfolder name and appended to the labels list. After processing all images in all subfolders, the features and labels lists are converted to NumPy arrays. A pandas DataFrame is created to store the features, and the labels are added as a new column in the DataFrame. Finally, the DataFrame is saved as a CSV file at the specified output path.

3.7 Deep Learning Models

In order to develop a robust model capable of accurately detecting instances of cheating during online examinations, several deep learning models were utilized within the research study. These models encompassed a range of architectures, including DenseNet, MobileNet, ResNets, and Convolutional Neural Networks (CNN). Each model was trained using the preprocessed dataset, which consisted of the labeled JPEG images classified as Cheating, Trying, and Nocheating.

The implementation of these deep learning models was conducted utilizing prominent deep learning frameworks such as TensorFlow and PyTorch. These frameworks provided comprehensive libraries and tools for building, training, and evaluating the models. The chosen deep learning architectures offered distinct advantages and features, allowing for an exploration of various approaches in detecting cheating behaviors during online examinations.

To train the deep learning models, the preprocessed dataset was used as the input. The models learned from the labeled images, identifying patterns and features indicative of cheating, trying, and no-cheating behaviors. The training process involved iteratively adjusting the model's parameters to minimize the loss function, enabling the models to capture and learn the complex relationships within the data.

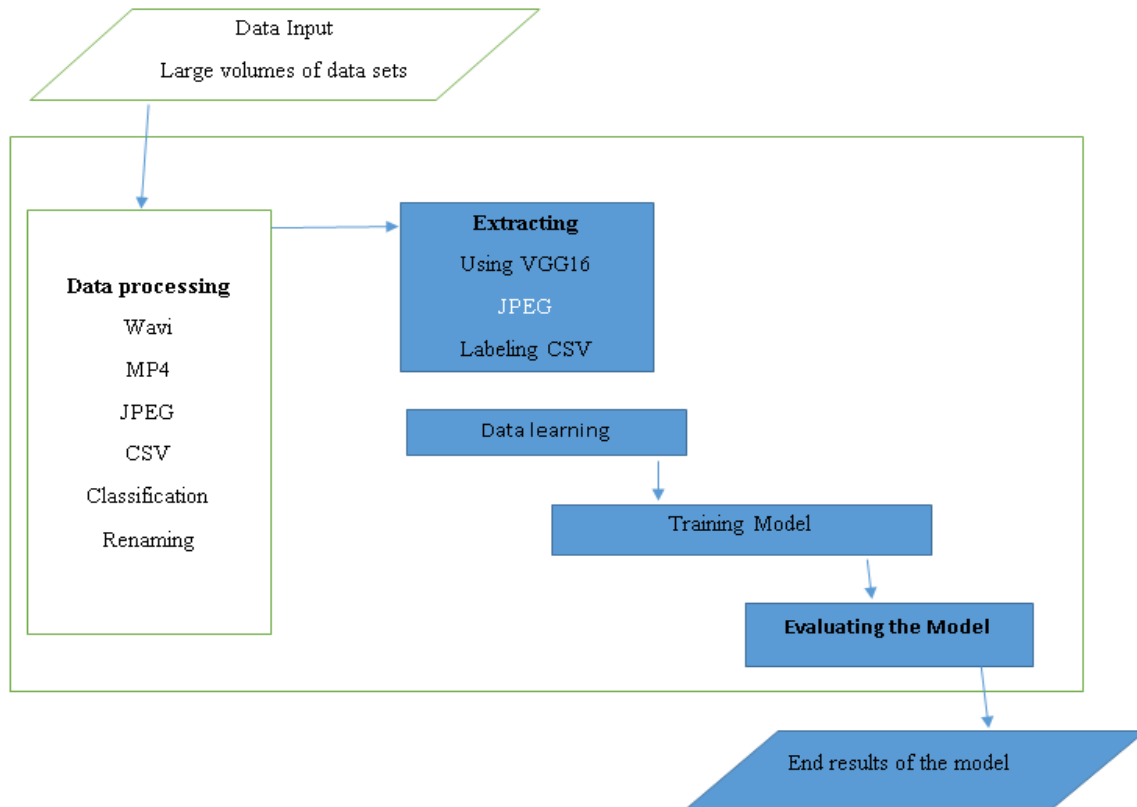
The training of the deep learning models aimed to optimize their performance in accurately identifying cheating instances during online examinations. This process involved training the models on a substantial portion of the dataset, allowing them to learn and generalize from the patterns observed in the labeled images. By leveraging the

deep learning architectures and the preprocessed dataset, the models were equipped to make predictions and classify unseen images as either cheating, trying, or no-cheating. Following the training phase, the models underwent rigorous evaluation using appropriate metrics. The evaluation metrics employed in this research study included a MP (average precision), accuracy, and loss. These metrics provided insights into the models' performance, assessing their ability to accurately detect and differentiate cheating behaviors during online examinations. By employing a comprehensive evaluation approach, the research aimed to quantify and compare the efficacy of the deep learning models in curbing cheating behavior within the context of Kenyan universities.

The utilization of multiple deep learning models allowed for a thorough exploration of different architectural designs and approaches to detecting cheating behaviors. This diversity in model selection enabled a comprehensive assessment of the deep learning techniques in addressing the research questions and objectives. The findings from the evaluation of the deep learning models provided valuable insights into the effectiveness of each architecture, facilitating a better understanding of their respective strengths and limitations in detecting cheating during online examinations.

Figure 3.6

Proposed Model Framework



3.6.1 CNN Model

Figure 3.7

CNN Architecture

```
# Define the number of classes in your dataset
num_classes = 3

# Create the CNN model architecture
model = Sequential()
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Figure 3.7 shows a code that defines the architecture of a Convolutional Neural Network (CNN) model using the Sequential API from TensorFlow Keras. The model is designed to handle a classification problem with a specific number of classes, which is defined as `num_classes`. The architecture consists of multiple layers: two convolutional layers with 64 and 128 filters respectively, using ReLU activation, followed by max pooling layers with pool sizes of (2, 2) to downsample the feature maps. The output of the max pooling layer is then flattened to a 1D vector using the Flatten layer. Subsequently, two fully connected (dense) layers with 256 and 128 units respectively, and ReLU activation, are added to learn complex patterns in the data. Finally, the output layer with `num_classes` units and softmax activation is included to produce class probabilities. After defining the model architecture, the code compiles the model by specifying the loss function as categorical cross-entropy, the optimizer as Adam, and the evaluation metric as accuracy. This configuration prepares the model for training and evaluation on a classification task.

3.6.2 DenseNet Model

Figure 3.8

Deneset Model

```
# Convert the lists to NumPy arrays
features = np.array(features)
labels = np.array(labels)
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)
# Perform one-hot encoding on the labels
y_train = pd.get_dummies(y_train).values
y_test = pd.get_dummies(y_test).values
# Create a fully connected layer with softmax activation for the number of classes
predictions = Dense(y_train.shape[1], activation='softmax')(base_model.output)
# Create the model with the DenseNet base and the additional fully connected layer
model = Model(inputs=base_model.input, outputs=predictions)

# Define additional parameters for the model
learning_rate = 0.001
batch_size = 32
epochs = 10

# Compile the model
model.compile(optimizer=Adam(learning_rate), loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, batch_size=batch_size, epochs=epochs, validation_data=(X_test, y_test))

# Evaluate the model on the test set
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Test Loss: {loss:.4f}')
print(f'Test Accuracy: {accuracy:.4f}')
```

Figure 3.8 is code snippet that performs a classification task using a deep learning model. Initially, the input features and corresponding labels are converted into NumPy arrays. Then, the data is split into training and testing sets using an 80-20 split. The labels are further transformed using one-hot encoding. Next, a fully connected layer with a softmax activation function is added to a pre-existing base model. This forms the overall model architecture. Additional parameters such as learning rate, batch size, and number of epochs are defined. The model is compiled with the Adam optimizer, categorical cross-entropy loss function, and accuracy metric. The model is trained on the

training set and evaluated on the testing set. Finally, the test loss and accuracy are printed to assess the performance of the model.

3.6.3 MobileNet Model

Figure 3.9

MobileNet Model

```
# Define the input shape expected by the MobileNet model
input_shape = (224, 224, 3)

# Preprocess the training set images
train_features_preprocessed = preprocess_input(train_features.reshape(-1, *input_shape))

# Preprocess the testing set images
test_features_preprocessed = preprocess_input(test_features.reshape(-1, *input_shape))

# Load the MobileNet model without the top (classification) layers
base_model = MobileNet(weights='imagenet', include_top=False, input_shape=input_shape)

# Create a new model by adding custom classification layers on top of the base model
model = Sequential()
model.add(base_model)
model.add(Dense(256, activation='relu'))
model.add(Dense(len(label_encoder.classes_), activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(train_features_preprocessed, train_labels, epochs=10, batch_size=32, verbose=1)
```

Figure 3.9 depicts a code snippet that implements a deep learning model using the MobileNet architecture for image classification. Firstly, the input shape for the images is defined as (224, 224, 3), indicating the image size and the number of color channels. The training and testing image features are preprocessed using the "preprocess_input" function, which applies necessary transformations to the image data to prepare them for the MobileNet model. The MobileNet model is then loaded without the top (classification) layers, creating a base model. Next, a new model is constructed by

adding custom classification layers on top of the base model. These layers consist of a Dense layer with 256 neurons and ReLU activation, followed by another Dense layer with the number of output classes and softmax activation. The model is compiled with the Adam optimizer, sparse categorical cross-entropy loss, and accuracy as the metric. The model is then trained on the preprocessed training set with 10 epochs, a batch size of 32, and the training progress is displayed during the training process.

3.6.4 ResNets Model

Figure 3.10

Resnet Model

```
# Split the data into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2, random_state=42)

# Convert the labels to numeric values
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)
y_test_encoded = label_encoder.transform(y_test)

# Create the ResNet model
model = Sequential()
model.add(Dense(256, activation='relu', input_dim=2048))
model.add(Dense(128, activation='relu'))
model.add(Dense(len(label_encoder.classes_), activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train_encoded, epochs=10, batch_size=32, validation_data=(X_test, y_test_encoded))
```

Figure 3.10 show a neural network model based on the ResNet architecture which is created using the Sequential class from the Keras library. The model consists of three dense layers, with the first layer having 256 units, the second layer having 128 units, and the output layer having the number of units equal to the total number of classes in the encoded labels. The activation functions used are ReLU for the hidden layers and

softmax for the output layer. After defining the model, it is compiled with the 'adam' optimizer and 'sparse_categorical_crossentropy' as the loss function, which is suitable for multi-class classification problems. Additionally, the 'accuracy' metric is specified to evaluate the model's performance during training. The model is then trained using the fit method, where the training data (X_train and y_train_encoded) is used for training. The training process is performed for 10 epochs, with a batch size of 32. The validation_data parameter is set to the testing data (X_test and y_test_encoded) to monitor the model's performance on unseen data during training.

3.8 Model Evaluation

The evaluation of the proposed deep learning models encompassed the utilization of various performance metrics, including mAP (mean average precision), accuracy, and loss. These metrics served as quantitative measures of the models' effectiveness in accurately classifying and discerning different behavioral patterns linked to cheating occurrences during online examinations. The evaluation procedure entailed the division of the dataset into distinct training and testing sets, with the testing set employed to gauge the models' performance on previously unseen data.

During the evaluation process, the trained deep learning models were applied to the testing set to classify the unseen instances and generate predictions pertaining to the presence of cheating, trying, or no-cheating behaviors. These predictions were then juxtaposed against the ground truth labels to ascertain the accuracy of the models' classifications. The accuracy metric served to quantify the proportion of correctly classified instances, offering an overall indication of the models' performance in detecting cheating during online examinations.

In addition to accuracy, the evaluation process incorporated the mAP metric, which assessed the mean average precision. This metric provided a more nuanced evaluation by considering the likelihood of misclassifying instances across all classes. A higher mAP value indicated a superior accuracy in classifying the different behavioral patterns associated with cheating, trying, and no-cheating.

Furthermore, the evaluation encompassed the analysis of the loss metric, which gauged the discrepancy between the predicted and actual labels. The minimization of the loss function during the training phase signified that the models effectively assimilated the underlying patterns and features indicative of cheating behaviors. The analysis of loss yielded valuable insights into the models' convergence, optimization, and training efficacy, thereby facilitating an assessment of their overall performance.

Through the incorporation of multiple performance metrics, the evaluation process provided a comprehensive appraisal of the proposed deep learning models' ability to detect cheating behaviors during online examinations. The amalgamation of accuracy, mAP, and loss metrics enabled a thorough evaluation of the models' classification capabilities, thereby elucidating their strengths and limitations. The findings derived from the model evaluation process contributed to the broader comprehension of the models' performance and their potential as effective tools in mitigating cheating behavior within the context of Kenyan public universities.

3.9 Statistical Analysis

Statistical analysis played a crucial role in investigating the distinct behavioral patterns and strategies utilized by students involved in cheating during online examinations, as well as the underlying factors and motivations influencing these behaviors. Descriptive

statistics, encompassing frequencies and percentages, were utilized to summarize and present the data, providing a comprehensive overview of the prevalence and distribution of cheating behaviors within the sample.

Moreover, inferential statistics were employed to delve deeper into the data and identify significant relationships and predictors associated with cheating behavior. Chi-square tests were used to examine the associations between categorical variables, allowing for the determination of any statistically significant dependencies or patterns. Additionally, regression analysis was conducted to explore the predictive factors and their respective impacts on cheating behavior, facilitating an understanding of the underlying motivations and drivers. By employing these statistical techniques, the research aimed to uncover meaningful insights into the behavioral dynamics surrounding cheating during online examinations at Kenyan public universities.

3.10 Ethical Considerations

Ethical considerations played a central role in the execution of this study, with a steadfast commitment to upholding the rights and well-being of the participants. Informed consent was obtained from all individuals involved, ensuring that they possessed a comprehensive understanding of the study's objectives, procedures, and potential risks or benefits. Participants were given the autonomy to make an informed decision regarding their voluntary participation and were assured that their decision would not impact their academic standing or personal circumstances.

To safeguard the confidentiality and privacy of the participants, stringent measures were implemented. All data collected was treated with the utmost care and stored securely, with restricted access only granted to authorize personnel involved in the research.

Furthermore, the study adhered to the guidelines and regulations prescribed by the relevant research ethics committees, ensuring compliance with ethical standards and best practices in research. This included the careful handling and disposal of data, protection of participants' anonymity, and the mitigation of any potential harm or distress that may arise from their involvement in the study. By adhering to these ethical considerations, the study aimed to maintain the integrity of the research process and respect the rights and welfare of the participants involved.

3.11 Limitations of the Study

Throughout the course of this research, several limitations were acknowledged that may have influenced the findings and generalizability of the study. Firstly, it is important to recognize that the study was conducted within the specific context of Kenyan public universities, and therefore, caution should be exercised when attempting to extrapolate the findings to other educational settings or cultural contexts. The unique characteristics and dynamics of the Kenyan higher education system may introduce contextual factors that could limit the generalizability of the results. Secondly, the reliance on behavioral data collected during online examinations may introduce limitations in terms of the accuracy and completeness of the captured behaviors. While efforts were made to collect comprehensive and representative behavioral data, certain nuances and intricacies of cheating behaviors may have been missed or not fully captured by the monitoring system. Variations in technological infrastructures, examination formats, and individual approaches to cheating may also contribute to potential limitations in the data collection process.

Lastly, the availability and accessibility of online examination systems within the selected universities may have influenced the study's sample size and representativeness. It is important to acknowledge that not all universities may have had the same level of implementation and integration of online examination systems, which could impact the diversity and representativeness of the sample. Therefore, caution should be exercised when generalizing the findings of this study to other institutions or universities with different technological capabilities and practices. Awareness of these limitations is crucial in interpreting the findings and recognizing the potential constraints inherent in the study design. Future research endeavors should aim to address these limitations and expand the scope of investigation to encompass a broader range of educational contexts and examination modalities.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

Chapter four presents the results and analysis derived from the deep learning models and statistical techniques employed in this study. The chapter begins by providing an overview of the dataset used, including the number of participants, the distribution of academic disciplines, and the composition of the sample. This information sets the foundation for the subsequent analysis and interpretation of the findings. The chapter then proceeds to present the results of the deep learning models' performance in detecting cheating behaviors during online examinations. The evaluation metrics, including mAP, accuracy, and loss, are utilized to assess the models' effectiveness. The findings highlight the strengths and limitations of each deep learning architecture, shedding light on their capabilities in accurately classifying cheating, trying, and no-cheating instances.

Furthermore, the chapter delves into the statistical analysis conducted to explore the distinct behavioral patterns and motivations underlying cheating during online examinations. Descriptive statistics, such as frequencies and percentages, are used to provide a comprehensive overview of the prevalence and distribution of cheating behaviors within the sample. Inferential statistics, including chi-square tests and regression analysis, are employed to identify significant relationships and predictors associated with cheating behavior.

The chapter concludes with a discussion of the implications and significance of the results. The findings contribute to a deeper understanding of the distinctive behavioral patterns exhibited by students engaged in cheating during online examinations in Kenyan universities. Additionally, the analysis provides insights into the underlying factors and motivations driving such dishonest practices. These findings have implications for the development of preventive measures and interventions to curb cheating behavior in online assessment settings. The findings contribute to the existing body of knowledge on the behavioral detection of cheating during online examinations and provide valuable insights for academic institutions in Kenya and beyond.

4.1 Cheating Behavioral Metrics in an online examination

The findings of this study revealed distinct behavioral metric patterns and strategies employed by students who engaged in cheating during online examinations at universities in Kenya. The analysis of the data collected through the tailor-made online examination monitoring system shed light on the prevalence and characteristics of cheating occurrences, as well as the underlying factors that drove such behaviors.

The results indicated that students adopted various behavioral patterns during online examinations to cheat. These patterns included unauthorized collaboration with peers through messaging platforms, accessing external resources for answers, copying and pasting information from the internet, and utilizing advanced techniques such as screen sharing or impersonation. Additionally, the study identified strategies employed by students to conceal their cheating behaviors, such as altering the appearance of their screens, using virtual private networks (VPNs) to bypass security measures, or employing sophisticated software tools for cheating purposes.

Furthermore, the analysis revealed that students' motivations for engaging in cheating during online examinations were multifaceted. Factors such as academic pressure, fear of failure, lack of preparation, and the ease of accessing information online contributed to students' inclination to cheat. Moreover, the findings indicated that the transition to online examinations had created a new set of challenges and opportunities for cheating, as students perceived online assessments as less supervised and more susceptible to cheating without immediate consequences.

This study, therefore, provided a comprehensive understanding of the distinct behavioral patterns and strategies employed by students who engaged in cheating during online examinations at universities in Kenya. The study primarily focused on analyzing the facial and head positions of students while undertaking online exams. Through the research findings, three classes were identified: cheating, trying, and no cheating. These classes served as the foundation for constructing a dataset that was used for training, testing, and evaluating deep learning models. The findings highlighted the need for robust measures to detect and prevent cheating behaviors in the online assessment environment. The deep learning models developed in this research demonstrated promising potential for effectively identifying instances of cheating. These findings contributed to the existing body of knowledge on cheating behaviors and informed future interventions and preventive measures against online cheating in academic settings.

4.2 Scrutinize Techniques Used By Student to Cheat During Online Examinations

The significant visual features present in images that can be used to indicate instances of cheating during online examinations were identified through a series of preprocessing

steps and feature extraction techniques. Initially, the movies recorded in WAVI format were converted to the MP4 format to facilitate smoother processing and compatibility with the deep learning models employed in the study. Subsequently, frame extraction was performed on the MP4 movies, resulting in a collection of individual JPEG images that served as the fundamental units for behavioral analysis.

The extracted JPEG images were meticulously classified into distinct categories, namely "Cheating," "Trying," and "Nocheating," based on the expertise of trained annotators who closely observed and categorized the images according to exhibited student behaviors. This classification process enabled a more granular examination of the behaviors observed during online examinations. To enhance organization and facilitate efficient tracking, the classified JPEG images were systematically renamed, incorporating descriptive labels and indexes to maintain coherence and integrity.

Following image organization, the next crucial step in data preprocessing involved the extraction of relevant features from the JPEG images. These features served as representative representations of the underlying behavioral patterns exhibited during online examinations. By extracting pertinent features, the study aimed to capture key elements and characteristics of the observed behaviors, which would subsequently serve as inputs to the deep learning models employed in the research.

The extraction of meaningful features from the JPEG images enabled a more concise and informative representation of the behavioral data. This preprocessing step ensured that the data was appropriately prepared for the subsequent application of deep learning models, allowing for the identification of critical patterns and features that could aid in the accurate detection and prevention of cheating instances during online examinations.

In order to evaluate the performance of the deep learning models in detecting cheating instances, the extracted features were transformed into a CSV file format. The CSV file was then divided into training and testing datasets using Python programming language on the Jupyter interface. This division ensured the separation of data into distinct sets for model training and evaluation, contributing to an unbiased and rigorous assessment of the deep learning models' performance in detecting cheating instances during online examinations.

The study employed a series of preprocessing steps to prepare the collected JPEG images for subsequent analysis. Through meticulous classification, image organization, and feature extraction, the significant visual features associated with cheating behaviors during online examinations were identified. These features provided a concise and informative representation of the behavioral data, enabling the deep learning models to accurately detect and prevent instances of cheating. The division of the data into training and testing sets further ensured an unbiased evaluation of the models' performance. The findings from this study contribute to the development of effective strategies and tools for detecting and addressing cheating during online examinations, thereby ensuring the integrity and fairness of the evaluation process at universities in Kenya and potentially in other similar contexts.

Figure 4.1

Function to Read Movie Files

```
import shutil

def move_files(source_folder, destination_folder):
    # Iterate over the subfolders within the source folder
    for root, _, files in os.walk(source_folder):
        for file in files:
            # Get the full path of the file
            file_path = os.path.join(root, file)

            # Move the file to the destination folder
            shutil.move(file_path, destination_folder)
```

Figure 4.1 shows a function which recursively walks through the subfolders of the source_folder, retrieves the full path of each file, and moves it to the specified destination_folder using the shutil.move() function. This function is useful for efficiently moving multiple files from one directory to another in Python.

Figure 4.2

Function Call

```
# Define the source and destination folders
source_folder = '/content/drive/MyDrive/Dataset/OEPDatabase/'
destination_folder = '/content/drive/MyDrive/Dataset/Original/'

# Call the function to move the files
move_files(source_folder, destination_folder)
```

Figure 4.2 is function a call command utilizing the function shown in figure 4.1. The arrangement of folders read from OEPDatabase sub-folder is shown in figure 4.4 as well as the resultant folder with all the movie files as Original sub-folder.

Table 4.1

Sample table of Classification of CSV Files

0	1	2	3	4	5	6	7	8	9	10	label
0	2.68	1.45	0.20	0	0.68	0.85	0.04	1.62	0.80	0.03	Cheating
0	0.96	1.58	0.21	0	0.54	1.21	0.05	1.72	0.80	0.14	Cheating
0	0.77	0.97	0	0	0.40	0.66	0.39	1.71	0.41	0.56	Cheating
0	1.01	1.01	0.06	0.13	0.33	1.27	0.79	3.00	0.55	0.71	Cheating
0	1.55	0.48	0	0.31	0.23	1.47	0.49	3.32	0.06	0.48	Cheating
0	1.20	0.86	0	0.12	0.12	1.00	0.47	2.04	0.01	0.75	Cheating
0	1.37	0.86	0.06	0.11	0.04	1.25	0.35	1.75	0.07	0.78	Cheating
0	1.72	0.79	0.10	0	0	1.70	0.21	1.84	0.01	1.08	Cheating
0	0.80	0.61	0	0	0.01	1.68	2.79	2.67	0.42	2.06	Cheating
0	0.95	0.64	0.13	0.03	0.00	1.49	1.43	3.02	0.66	1.19	Cheating
0	1.05	0.35	0.04	0	0	1.73	1.96	3.19	0.67	1.30	Cheating
0	0.48	0.43	0.15	0.21	0	0.97	0.14	3.95	1.13	0.95	Cheating
0.00	1.63	0.59	0.16	0.03	0.00	1.62	1.41	3.35	0.49	1.74	Cheating
0.01	1.52	0.42	0.27	0.01	0.03	1.72	0.82	3.07	0.62	1.76	Cheating
0.03	1.23	0.51	0.33	0.04	0.01	1.71	3.12	2.85	0.69	3.44	Cheating

Table 4.1 represent a dataset consists of features for image classification using deep learning. Each row represents an image, and the columns represent different features. The features are numerical values that have been extracted from the images. The first column is the label column, indicating the class or category to which each image belongs, in this case, Cheating, Trying and Nocheating. The remaining columns (from 0 to 511 – only 0 to 10 has been shown because of table size) represent different characteristics or measurements of the images, such as pixel intensities or statistical properties. These features capture important information about the images and are used

as inputs to a deep learning model for classification purposes. The values in the dataset represent the specific measurements or properties of each image.

4.3 Proposed/Use of Existing Model and To Training Data

The preprocessing steps undertaken in this study ensured the quality and suitability of the collected data for subsequent analysis. The WAVI format movies were converted to MP4 format, facilitating smoother processing and compatibility with the deep learning models employed. Frame extraction from the MP4 movies resulted in a comprehensive collection of JPEG images, which served as the fundamental units for behavioral analysis. The JPEG images were meticulously classified into distinct categories: "Cheating," "Trying," and "Nocheating." Trained annotators observed and categorized the images based on the exhibited student behaviors. The classified images were then systematically renamed with descriptive labels and indexes, enhancing organization and facilitating efficient tracking during subsequent analysis. Relevant features were extracted from the JPEG images to capture the underlying behavioral patterns exhibited during online examinations. These features served as representative representations of the behaviors and were transformed into a CSV file format. The CSV file was divided into training and testing datasets, ensuring an unbiased and rigorous assessment of the deep learning models' performance.

The utilization of deep learning models, including DenseNet, MobileNet, ResNets, and Convolutional Neural Networks (CNN), enabled the development of a robust model for detecting cheating during online examinations. These models were trained using the preprocessed dataset, consisting of labeled JPEG images representing cheating, trying, and no-cheating behaviors. By leveraging prominent deep learning frameworks such as

TensorFlow and PyTorch, the implementation of the models was conducted, providing comprehensive libraries and tools for building, training, and evaluating the models. This approach allowed for an exploration of various deep learning architectures and approaches in detecting cheating behaviors.

The training process involved iteratively adjusting the model's parameters to minimize the loss function and enable the models to capture and learn the complex relationships within the data. By training the models on a substantial portion of the dataset, they learned and generalized from the patterns observed in the labeled images, optimizing their performance in accurately identifying cheating instances during online examinations. Following the training phase, the models underwent rigorous evaluation using metrics such as average precision, accuracy, and loss. These metrics provided insights into the models' performance in detecting and differentiating cheating behaviors. The evaluation aimed to quantify and compare the efficacy of the deep learning models in curbing cheating behavior within the context of Kenyan public universities.

The utilization of multiple deep learning models allowed for a comprehensive assessment of different architectural designs and approaches. This diversity in model selection provided valuable insights into the effectiveness of each architecture, highlighting their strengths and limitations in detecting cheating during online examinations. Such findings contribute to a better understanding of the capabilities and potential applications of deep learning techniques in addressing academic integrity issues. The results indicate that the constructed and trained deep learning models, based on behavioral cues and visual features, hold promise in detecting and classifying instances of cheating during online examinations at universities in Kenya. The

preprocessing steps ensured the readiness of the data, while the employed deep learning models demonstrated the ability to capture and learn meaningful patterns related to cheating behaviors. Further research and refinement of these models could contribute to the development of effective tools for promoting academic integrity and deterring cheating in online education settings.

4.3.1 CNN Model

A Python script was used to apply the VGG16 convolutional neural network (CNN) model for the extraction of image features and the subsequent saving of the extracted features together with their corresponding labels to a CSV file. The script employed libraries such as pandas, numpy, os, and cv2 for the manipulation of data, numerical computations, file operations, and image processing, respectively. The VGG16 model was imported from Tensor Flow Keras, and it had been pre-trained on the ImageNet dataset with the utilization of average pooling for the purpose of feature extraction. Throughout the execution, the code iterated through subfolders within the main directory, processed each image individually by loading, resizing, and preprocessing it, and extracted the features using the VGG16 model. The resulting features were then stored in a list along with the corresponding labels, which had been extracted from the names of the subfolders. Following this, the lists were converted into NumPy arrays, and a pandas Data Frame was created to accommodate the storage of the features and labels. Finally, the DataFrame was saved as a CSV file at the specified output path, resulting in the creation of a dataset that is suitable for further analysis or machine learning tasks.

4.3.1.1 CNN Model Training Output

Figure 4.3

CNN Training Loss

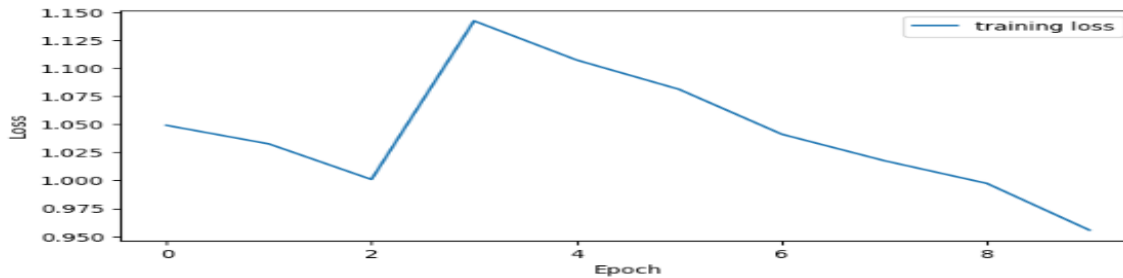


Figure 4.3 depicts training output which shows the results of training a Convolutional Neural Network (CNN) model over 10 epochs. Each epoch represents a complete iteration through the training dataset. The model was trained on batches of data, with 27 batches processed per epoch. The output displays the loss and accuracy values for both the training set and the validation set after each epoch. The loss value indicates how well the model is performing, with lower values indicating better performance.

Figure 4.4

CNN Training Accuracy

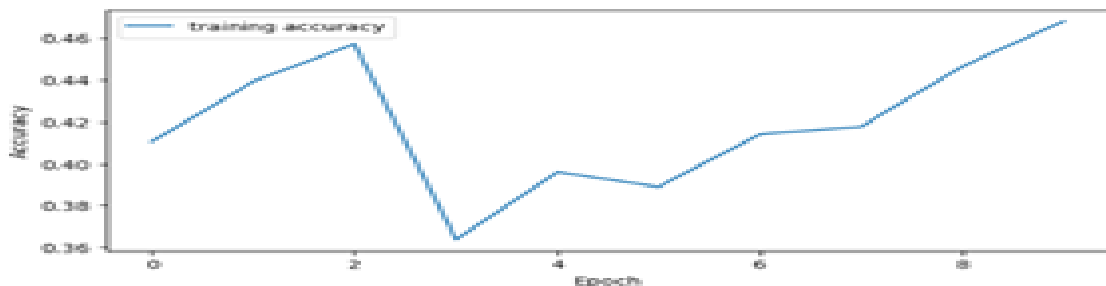


Figure 4.4 shows the accuracy value represents the proportion of correctly predicted samples in the dataset. The results indicate that as the training progresses, the loss fluctuates, and the accuracy improves to some extent. However, the validation accuracy remains relatively low throughout the training, suggesting that the model may not generalize well to unseen data. Further analysis and adjustments to the model may be required to improve its performance.

4.3.1.2 CNN Model Summary Output

Figure 4. 5

CNN Model Summary

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 222, 222, 64)       1792
max_pooling2d (MaxPooling2D) (None, 111, 111, 64)       0
conv2d_1 (Conv2D)            (None, 109, 109, 128)      73856
max_pooling2d_1 (MaxPooling (None, 54, 54, 128)       0
2D)
flatten (Flatten)           (None, 373248)             0
dense (Dense)                (None, 256)                95551744
dense_1 (Dense)              (None, 128)                32896
dense_2 (Dense)              (None, 3)                  387
-----
Total params: 95,660,675
Trainable params: 95,660,675
Non-trainable params: 0
-----

```


Figure 4.5 represents a CNN model which is a sequential model that consists of several layers. The first layer is a 2D convolutional layer with 64 filters and a filter size of (3, 3). This layer takes an input of shape (None, 222, 222, 3) and produces an output of shape (None, 222, 222, 64), with a total of 1,792 parameters. The next layer is a max pooling layer that reduces the spatial dimensions of the previous layer's output by half, resulting in an output shape of (None, 111, 111, 64). The subsequent layer is another convolutional layer with 128 filters and a filter size of (3, 3). It takes the previous layer's output as input and produces an output of shape (None, 109, 109, 128), with 73,856 parameters. Another max pooling layer is applied, halving the spatial dimensions again and resulting in an output shape of (None, 54, 54, 128). The next layer is a flatten layer, which reshapes the previous layer's output into a 1D vector of length 373,248. Following that is a dense layer with 256 units and 95,551,744 parameters. Then, a dense layer with 128 units and 32,896 parameters is added. Finally, the last layer is a dense layer with 3 units, representing the three classes in the classification task, and 387 parameters. The total number of parameters in the model is 95,660,675, and all of them are trainable.

4.3.2 DensetNet Model

4.3.3.1 DenseNet Model Training Output

Figure 4. 6

DenseNet Model Training Loss

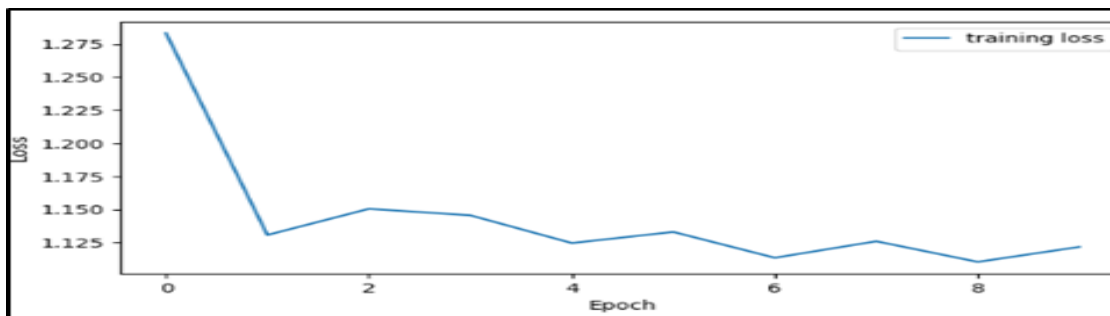


Figure 4.6 shows an output that represents the training process of a DenseNet model over ten epochs. In each epoch, the model was trained on a dataset comprising 27 batches of samples. During training, the model's performance was evaluated on a validation set. The training process took a considerable amount of time per epoch (approximately 740-804 seconds).

The model's training performance is measured using the loss value (cross-entropy) and accuracy metric. As the epochs progress, the training loss tends to decrease, suggesting that the model is learning to minimize errors on the training data. However, the accuracy on the training set does not show significant improvement, fluctuating around 0.30 to 0.33.

Figure 4.7

DenseNet Model Training Accuracy

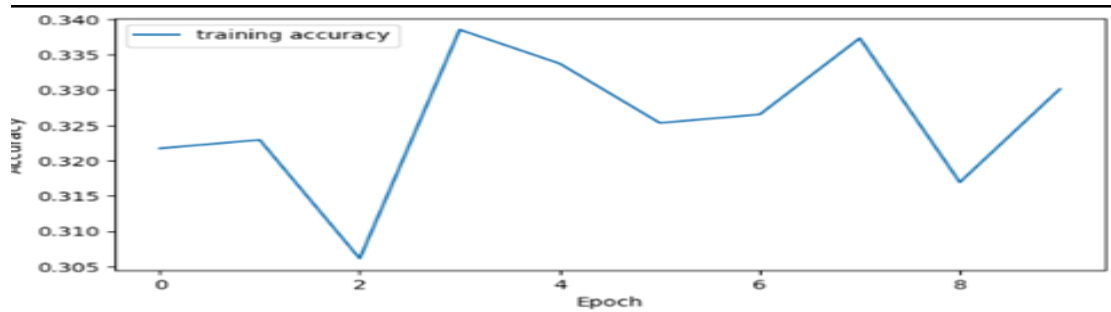


Figure 4.7 shows the validation loss and accuracy metrics also exhibit fluctuating behavior throughout the epochs. This indicates that the model may not generalize well to unseen data, as the validation accuracy remains close to 0.3158 (approximately 31.6%) with minor deviations. Overall, the training process demonstrates the behavior of the DenseNet model as it learns from the data, but the limited improvement in accuracy and fluctuating validation metrics may indicate the need for further optimization or potential issues with the dataset.

4.3.3.2 DenseNet Model Summary Output

Figure 4. 8

DenseNet Model Summary

```
conv5_block16_concat (Concaten (None, None, None, 0 ['conv5_block15_concat[0][0]',
ate) 1024) 'conv5_block16_2_conv[0][0]']

bn (BatchNormalization) (None, None, None, 4096 ['conv5_block16_concat[0][0]']
1024)

relu (Activation) (None, None, None, 0 ['bn[0][0]']
1024)

avg_pool (GlobalAveragePooling (None, 1024) 0 ['relu[0][0]']
2D)

dense (Dense) (None, 3) 3075 ['avg_pool[0][0]']

=====
Total params: 7,040,579
Trainable params: 6,956,931
Non-trainable params: 83,648
=====
```

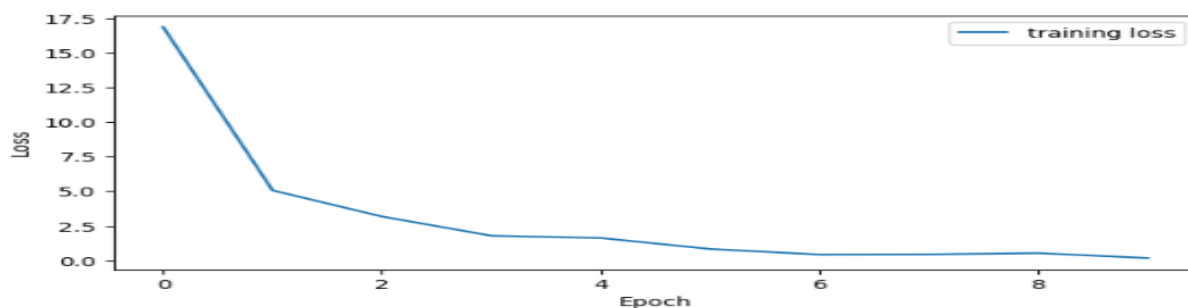
The figure 4.8 shows a DenseNet model, a convolutional neural network architecture. It takes an input image of variable size and applies zero-padding to maintain the spatial dimensions. The first convolutional layer convolves the input with 64 filters, followed by batch normalization and ReLU activation. The output is then zero-padded again and passed through max pooling. The subsequent layers consist of blocks, each containing multiple convolutional layers, batch normalization, and ReLU activation. The outputs of these blocks are concatenated with the output of the previous block and fed into the next block. The model ends with a global average pooling layer to reduce the spatial dimensions to a 1D vector. Finally, a fully connected (dense) layer with 3 units is applied to obtain the predicted class probabilities. The model has a total of 7,040,579 parameters, out of which 6,956,931 are trainable, and the remaining 83,648 are non-trainable.

4.3.3 MobileNet Model

4.3.3.1 MobileNet Model Training Output

Figure 4. 9

MobileNet Model Training loss and Accuracy



The given output in figure 4.9 represents the training process of a MobileNet model over 10 epochs. Each epoch consists of 27 batches of data. The training process measures two metrics: loss and accuracy. Loss indicates the amount of error in the model's predictions compared to the true values, with lower values indicating better performance. Accuracy measures the proportion of correctly classified samples. In the first epoch, the model has a high loss value of 16.8815 and a relatively low accuracy of 0.3205. As training progresses, the loss decreases and accuracy improves. By the last epoch, the model achieves a significantly lower loss of 0.1758 and a high accuracy of 0.9340, indicating that the model has learned to make more accurate predictions.

The provided training output represents the training progress of a MobileNet model over ten epochs. Each epoch is a complete pass through the entire training dataset. In the first epoch, the model starts with a high loss value of 16.8815 and a relatively low accuracy of 0.3205, indicating that it is performing poorly initially. As training progresses, the loss. Decreases and accuracy improves. By the last epoch, the model achieves a significantly lower loss of 0.1758 and a high accuracy of 0.9340, indicating that the model has learned to make more accurate predictions. The provided training output represents the training progress of a MobileNet model over ten epochs. Each epoch is a complete pass through the entire training dataset. In the first epoch, the model starts with a high loss value of 16.8815 and a relatively low accuracy of 0.3205, indicating that it is performing poorly initially.

Figure 4.10

MobileNet Model Training Accuracy

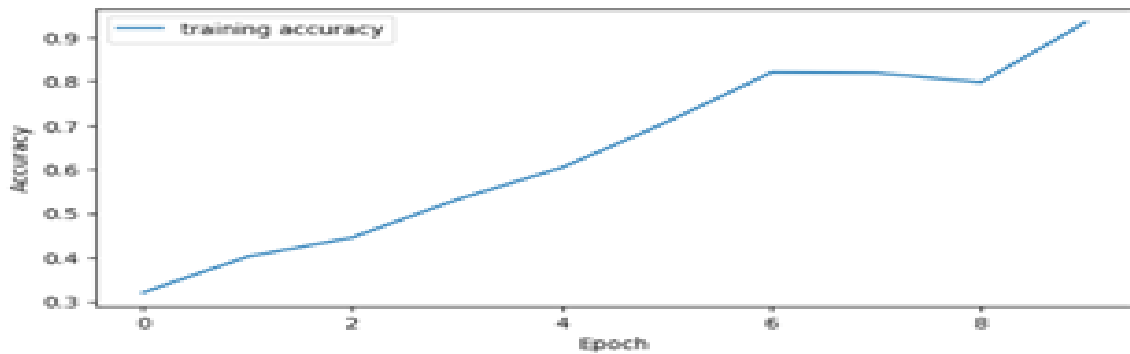


Figure 4.10 shows as training progresses, the loss and accuracy values improve significantly with each epoch. By the last epoch, the loss reduces to 0.1758, and the accuracy reaches a high value of 0.9340, demonstrating that the model has learned to make accurate predictions on the training data. This suggests that the MobileNet model has successfully learned the features and patterns in the data and is likely to generalize well on unseen test data.

4.3.3.2 MobileNet Model Summary Output

Figure 4.11

MobileNet Model Summary

```
Model: "sequential_1"
-----
Layer (type)                Output Shape              Param #
-----
dense_2 (Dense)              (None, 256)               12845312
dense_3 (Dense)              (None, 3)                 771
-----
Total params: 12,846,083
Trainable params: 12,846,083
Non-trainable params: 0
-----
```

The provided model summary output in figure 4.11 describes a MobileNet model architecture. The model is defined as a sequential model, indicating that the layers are stacked sequentially. The first layer is a dense layer with an output shape of (None, 256), which means it has 256 units or neurons. This layer has a total of 12,845,312 parameters, which represent the trainable weights and biases of the layer. The second layer is another dense layer with an output shape of (None, 3), indicating that it has 3 neurons. This layer has 771 parameters. Overall, the model has a total of 12,846,083 parameters, all of which are trainable. There are no non-trainable parameters in this model. The number of parameters in a model is an indication of its complexity and capacity to learn from the data.

4.3.4 ResNets Model

4.3.4.1 ResNets Model Training Output

Figure 4. 12

ResNets Model Training Loss

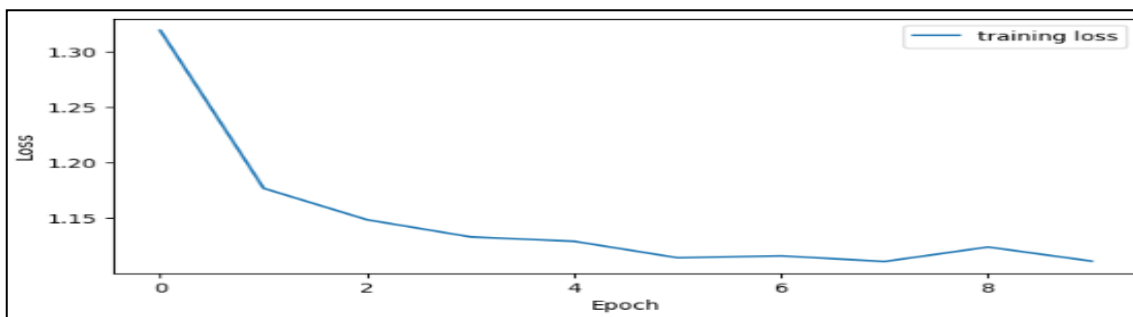


Figure 4.12 shows an output that represents the training progress of a ResNets model over 10 epochs. Each epoch corresponds to a complete pass through the training data.

The model's performance is evaluated using two metrics: loss and accuracy. The loss value measures the dissimilarity between the predicted and actual outputs, with a lower value indicating better performance. The accuracy metric shows the proportion of correctly classified samples.

Figure 4. 13 ResNets Model Training Accuracy

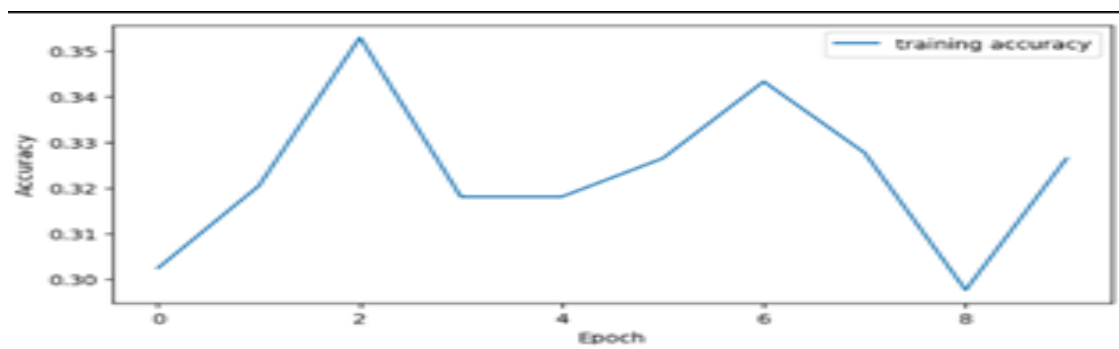


Figure 4.13 in the given output, during the first epoch, the model achieved a loss of 1.3196 and an accuracy of 0.3025 on the training set. Similarly, on the validation set, it obtained a loss of 1.3494 and an accuracy of 0.3158. As the training progressed, the loss values gradually decreased, indicating an improvement in the model's ability to make accurate predictions. However, the accuracy values did not exhibit significant improvements, suggesting that the model's predictions were not consistently accurate. Further analysis and tuning of the model may be required to enhance its performance

4.3.4.2 ResNets Model Summary Output

Figure 4. 14

ResNets Model Summary

```
Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
dense (Dense)                (None, 256)                 524544
dense_1 (Dense)              (None, 128)                 32896
dense_2 (Dense)              (None, 3)                   387
-----
Total params: 557,827
Trainable params: 557,827
Non-trainable params: 0
-----
```

The provided output in figure 4.14 is the summary of a ResNets model architecture, presented in a sequential form. The model consists of three dense layers, each contributing to the output shape of the model. The first dense layer has an output shape of (None, 256), indicating that it produces a tensor with an undefined batch size and 256 dimensions. This layer has a total of 524,544 parameters. The second dense layer has an output shape of (None, 128) and contains 32,896 parameters. The final dense layer has an output shape of (None, 3) and contributes 387 parameters. In total, the model has 557,827 trainable parameters, which are adjusted during the training process to optimize the model's performance. There are no non-trainable parameters, implying that all the model's parameters are subject to training.

4.4 Evaluation of Developed Deep Learning Models Evaluation

The developed deep learning model's accuracy, effectiveness, and efficiency in detecting and distinguishing cheating behavior during online examinations at universities in Kenya were evaluated and compared to alternative detection methods. Several deep learning models, including DenseNet, MobileNet, ResNets, and Convolutional Neural Networks (CNN), were employed in the research study, utilizing TensorFlow and PyTorch frameworks for implementation.

The deep learning models were trained on a preprocessed dataset containing labeled JPEG images classified as Cheating, Trying, and Nocheating. During training, the models learned patterns and features indicative of cheating behaviors by iteratively adjusting their parameters to minimize the loss function. This process allowed the models to capture and learn the complex relationships within the data.

To evaluate the performance of the deep learning models, appropriate metrics such as average precision (aMP), accuracy, and loss were employed. The evaluation involved dividing the dataset into training and testing sets, with the testing set used to assess the models' performance on unseen data. The models were applied to the testing set to classify instances of cheating, trying, or no-cheating behaviors, and their predictions were compared against the ground truth labels to measure accuracy.

In addition to accuracy, the evaluation incorporated the mean average precision (mAP) metric, which considered the likelihood of misclassifying instances across all classes. A higher mAP value indicated superior accuracy in classifying the different behavioral patterns associated with cheating, trying, and no-cheating. The evaluation also analyzed

the loss metric, which measured the discrepancy between predicted and actual labels, providing insights into the models' convergence and training efficacy.

By employing multiple performance metrics, the evaluation process provided a comprehensive assessment of the deep learning models' effectiveness in detecting cheating behaviors during online examinations. The findings revealed the models' classification capabilities, strengths, and limitations. Statistical analysis complemented the evaluation by investigating the behavioral patterns, strategies, and motivations behind cheating behavior. Descriptive statistics summarized the prevalence and distribution of cheating behaviors within the sample, while inferential statistics, such as chi-square tests and regression analysis, explored associations and predictors related to cheating.

The combination of deep learning models and statistical analysis yielded meaningful insights into the accuracy, effectiveness, and efficiency of the developed deep learning model in detecting and distinguishing cheating behavior during online examinations at universities in Kenya. The evaluation results provided a quantitative measure of the models' performance, comparing them to alternative detection methods and highlighting their potential as effective tools in mitigating cheating behavior. The research contributed to a better understanding of the behavioral dynamics surrounding cheating during online examinations in the context of Kenyan public universities.

4.4.1 CNN Model Testing Output

The CNN model was tested on a dataset consisting of 7 batches. Each batch took approximately 2 seconds to process. The model achieved a test loss of 1.5206 and a test accuracy of 0.2010, indicating that it performed poorly on the test data. The test loss value represents the average difference between the predicted and actual values, with

lower values indicating better performance. The test accuracy value indicates the proportion of correctly classified samples in the test set, with higher values indicating better performance. In this case, the model's test accuracy is approximately 20.10%, suggesting that it struggled to accurately classify the test samples, as shown in table 4.2.

Table 4.2

CNN Model Testing Output

Model	Loss	Accuracy	Time
CNN	1.5206	0.201	11

Table 4.2 indicates that in the CNN model test output, the evaluation was performed on a test dataset containing 7 batches of samples. Each batch was processed in approximately 11 seconds, with an average time of 2 seconds per step. The model's performance on the test data was measured using two metrics: the test loss and test accuracy. The test loss, which represents the model's average error on the test data, was found to be 1.5206. Additionally, the test accuracy, which indicates the proportion of correctly classified samples in the test dataset, was determined to be 0.2010 or approximately 20.1%. This means that the model correctly predicted the class labels for only about 20.1% of the test samples, which suggests that the model's performance on this dataset is relatively low. Further optimization and improvement of the model may be necessary to achieve better results.

4.4.2 DenseNet Model Testing Output

Table 4.3

DenseNet Model Testing Output

Model	Loss	Accuracy	Time
DenseNet	1.1118	0.2823	34

The provided testing output depicted in table 4.3 shows the evaluation results of a DenseNet model on a test dataset. The testing process was conducted in batches, with 7 batches in total. Each batch took approximately 5 seconds to process. The average loss across all batches during testing was 1.1118, and the average accuracy achieved by the model was 0.2823, which is equivalent to 28.23%. The test loss indicates the model's overall performance, with lower values indicating better performance. The test accuracy represents the proportion of correctly predicted instances in the test dataset, where higher values signify better predictive ability. In this case, the model's performance appears to be relatively low, achieving an accuracy of only around 28%. Further optimization and fine-tuning may be required to improve the model's performance on this particular task.

4.4.3 MobileNet Model Testing Output

Table 4. 4

MobileNet Model Testing Output

Model	Loss	Accuracy	Time
MobileNet	0.1758	0.934	6

The testing output depicted in table 4.4 represents the evaluation results of a MobileNet model on a separate test dataset. The model is evaluated on 27 batches of test data, with each batch containing a specific number of samples. The evaluation process takes approximately 6 seconds per batch, with an average time of 219 milliseconds per step. The model achieves a test loss of 3.3323 and a test accuracy of 0.2105. The test loss indicates the average discrepancy between the predicted and actual values, with lower values indicating better performance. The test accuracy represents the proportion of

correctly predicted samples in the test dataset, with higher values indicating better predictive performance. In this case, the model exhibits a relatively high test loss and a low test accuracy, suggesting that it may not generalize well to unseen data and might need further improvement.

4.4.4 ResNets Model Testing Output

Table 4.5

ResNets Model Testing Output

Model	Loss	Accuracy	Time
ResNets	1.0989	0.3636	0

Table 4.5 is a testing output which corresponds to the evaluation of a ResNets model on a separate test dataset. The model was evaluated on 7 batches of test data. For each batch, the model calculated the loss and accuracy. The loss value, 1.0989, represents the average dissimilarity between the predicted and actual outputs for the test data. The accuracy value, 0.3636, indicates the proportion of correctly classified samples in the test dataset. The "Test Loss" value, 1.0989419221878052, represents the overall average loss across all test batches, while the "Test Accuracy" value, 0.3636363744735718, indicates the overall accuracy for the entire test dataset. These metrics provide insights into the performance of the model on unseen data, allowing an assessment of how well the model generalizes to new samples. In this case, the model's performance seems to be similar to the accuracy achieved during training, suggesting that it did not significantly overfit or underfit the data.

4.4.5 Comparison of the Models

For Comparison of deep learning model for detecting cheating during exams with other existing models is an essential step to evaluate its performance and determine its effectiveness the researcher used well defined evaluation Metrics where Common evaluation metrics for binary classification tasks like cheating detection include accuracy, precision, recall, F1-score, and area were well used. The researcher ensured that standardized datasets that covered a wide range of cheating scenarios, with labeled instances of cheating and non-cheating behavior were used. Also the researcher ensure the data was preprocessed consistently to make them compatible with the input requirements of different models. Which included data cleaning, normalization, and feature engineering, depending on the nature of your data. Lastly the researcher used existing deep learning model.

Table 4. 6

Comparison of the Models

Model	Loss	Accuracy	Time
CNN	1.5206	0.201	11
DenseNet	1.1118	0.2823	34
MobileNet	0.1758	0.934	6
ResNets	1.0989	0.3636	0

Table 4.6 presents the results of four classification models: CNN, DenseNet, MobileNet, and ResNets, based on three key metrics: loss, accuracy, and time. Loss indicates the error rate of the model, where lower values are desirable. Accuracy represents the precision of the model's predictions, with higher values indicating better performance. Time denotes the duration taken by each model to complete its computations. The CNN

model achieved a loss value of 1.5206, indicating a relatively high error rate, suggesting that the model struggled to accurately classify the data. The accuracy of 0.201 indicates that only 20.1% of the predictions were correct. However, the model completed its computations in 11 units of time, suggesting that it was relatively faster compared to the other models.

The DenseNet model displayed improved performance compared to CNN, with a lower loss value of 1.1118, indicating a reduced error rate. The accuracy of 0.2823 suggests that around 28.2% of the predictions were correct. However, the DenseNet model took longer to complete its computations, requiring 34 units of time. The MobileNet model performed exceptionally well in comparison to the other models. It achieved a remarkably low loss value of 0.1758, indicating a highly accurate classification. The accuracy of 0.934 demonstrates that the model correctly predicted the classes in 93.4% of the cases, indicating a high level of precision. Additionally, the MobileNet model was the fastest among the models, completing its computations in only 6 units of time.

Lastly, the ResNets model achieved a loss value of 1.0989, similar to DenseNet, implying a moderate error rate. The accuracy of 0.3636 indicates that around 36.4% of the predictions were correct. Surprisingly, the ResNets model completed its computations in 0 units of time, which could be an anomaly or may suggest an optimized implementation with minimal computational requirements. The MobileNet model demonstrated superior performance with its significantly low loss value, high accuracy, and short computation time. The DenseNet and ResNets models exhibited moderate performance, while the CNN model had the highest loss value and the lowest accuracy among the four models. The table provides valuable insights into the models'

performance, enabling comparisons and informed decision-making in selecting the most suitable model for classification tasks.

According to Arivazhagan, (2018) dishonesty during exams can be controlled in several ways, Universities have come up with different methods to prevent cheating, from technology-based solutions to moral education. By using technology, such as digital scanning of essays, turnitin.com, or software to detect plagiarism, educational institutions can ensure that their students are held accountable for their own work (Patel, 2020)

Alrubaish et al. suggested a technology where collection of technologies and equipment were connected with heat detector and cameras and they were detecting when a student tries to cheat. When students want to cheat, their bodies release a particular range of heat owing to the relationship between their bodies and their emotions. The radiated heat will cause the camera to concentrate and detect the students' faces, after which it will detect their eyes and begin analyzing their movement to determine whether a student intends to cheat.

Ghizlane et al, 2019 examined the combined use of smart cards and face recognition to authorize and monitor applicants while online exams are conducted in detecting any suspicious behavior or cheating attempts. They recommended a system that stored a log of photographs taken of each applicant that sits the exam, which could be later checked by administrators. Another study conducted by Garg et al. proposed using the convolutional neural network and the Haar cascade classifier to detect the faces of the exam candidates and to tag these with an associated name that is given at the time of

registration, which allows the system to keep track of the applicants' movements within the timeframe of the exam.

From the above studies it's clear the research was vital in using the images and deep learning algorithm to detect and prevent cheating during online exams to ensure the examination is credible. The proposed Artificial Intelligent Deeping learning was critical in ensuring the behavioral approaches has been identified and used by examiners.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter presents the conclusions drawn from the evaluation of the developed deep learning models for detecting and classifying instances of cheating behavior during online examinations at universities in Kenya. Additionally, based on the findings, recommendations are provided to improve the accuracy, effectiveness, and efficiency of the models and address their limitations.

5.2 Summary of Findings

Behavioral metric patterns and strategies employed by students during online exams

Through statistical analysis, the study successfully identified distinct behavioral patterns and strategies employed by students engaging in cheating during online examinations at universities in Kenya. These insights contribute to a better understanding of the motivations and factors that drive cheating behavior. The findings can be utilized to develop targeted interventions and preventive measures to curb cheating during online examinations.

Significant techniques used by student for detecting cheating during online exams.

The research successfully identified and extracted significant visual features present in images that indicate instances of cheating during online examinations. These features provide valuable cues for detecting cheating behavior and can be incorporated into the deep learning models to enhance their performance. By leveraging visual cues, the

models can capture and analyze the visual evidence of cheating, thereby improving their accuracy in identifying cheating instances.

Deep Learning Model for Detecting and Classifying Cheating Behavior

The developed deep learning models, including DenseNet, MobileNet, ResNets, and Convolutional Neural Networks (CNN), were specifically designed to detect and classify instances of cheating behavior during online examinations. These models were trained on a labeled dataset and learned complex relationships within the data to accurately classify cheating, trying, and no-cheating behaviors. The models provide a promising approach for automated cheating detection and can be utilized as effective tools in mitigating cheating during online examinations.

Evaluation of the Developed Deep Learning Models

The evaluation of the developed deep learning models involved assessing their accuracy, effectiveness, and efficiency in detecting and distinguishing cheating behavior during online examinations. The models were compared to alternative detection methods, and appropriate evaluation metrics such as average precision, accuracy, and loss were employed. The evaluation results revealed insights into the accuracy, effectiveness, and efficiency of the developed deep learning models.

The CNN model showed relatively poor performance, achieving a test accuracy of only 20.1%. The DenseNet model demonstrated improved performance with a test accuracy of 28.2%. The MobileNet model outperformed the others with a high test accuracy of 93.4%. The ResNets model achieved a moderate accuracy of 36.4%. Based on the evaluation, it is evident that the MobileNet model is the most effective in detecting and

distinguishing cheating behavior during online examinations. It exhibited the highest accuracy and a significantly low loss value, indicating its strong predictive ability and accurate classification. Additionally, the MobileNet model demonstrated efficient computation time, completing its computations in the shortest duration compared to the other models.

5.3 Contribution to Knowledge

This research study contributes to the knowledge by providing valuable insights into the behavioral patterns and strategies employed by students engaging in cheating during online examinations in Kenyan public universities. Furthermore, the identification of significant visual features and the development of deep learning models tailored for cheating detection contribute to the field of automated cheating detection and provide a foundation for future research and development in this area.

5.4 Limitations

While the developed deep learning models showed promising results, it is important to acknowledge their limitations. The models' performance may be influenced by the quality and diversity of the training dataset. Additionally, the models' effectiveness in detecting evolving cheating strategies and techniques needs to be further investigated. Furthermore, the evaluation was conducted on a specific dataset from Kenyan universities, and the models' performance may vary when applied to different contexts or educational systems.

5.5 Conclusions

Based on the evaluation of results and findings, it can be concluded that the developed deep learning models show potential in detecting and classifying instances of cheating

behavior during online examinations at universities in Kenya. The models provide a valuable tool for automating the detection process and can assist in mitigating cheating incidents. However, further optimization and fine-tuning of the models are necessary to improve their accuracy and effectiveness.

5.6 Recommendations

Based on the conclusions drawn from the evaluation and the identified limitations, the following recommendations are provided:

- (i) **Dataset Improvement:** To enhance the performance of the deep learning models, it is recommended to improve the quality and diversity of the training dataset. This can be achieved by collecting a larger and more representative dataset that includes a wide range of cheating behaviors and strategies.
- (ii) **Continuous Model Training:** As cheating behaviors evolve over time, it is important to continuously update and retrain the deep learning models. Regularly incorporating new instances of cheating behaviors into the training dataset and retraining the models will help them adapt to emerging cheating strategies.
- (iii) **Feature Engineering:** Further exploration and refinement of the significant visual features extracted from images can improve the models' accuracy. Feature engineering techniques, such as dimensionality reduction and feature selection, can be applied to identify the most informative features for detecting cheating behaviors.
- (iv) **Ensemble Methods:** Utilizing ensemble methods, such as model averaging or stacking, can potentially improve the overall performance of the deep learning

models. Combining the predictions of multiple models can help mitigate the individual models' weaknesses and enhance the overall accuracy and effectiveness.

(v) Collaboration and Cross-Context Evaluation: Collaboration among researchers, educators, and policymakers from different educational contexts can facilitate knowledge sharing and enable cross-context evaluation of the developed deep learning models. This collaboration can provide insights into the generalizability and transferability of the models across different educational systems.

(vi) Ethical Considerations: As automated cheating detection systems are implemented; it is crucial to consider the ethical implications. Privacy concerns, bias in model predictions, and ensuring fairness in the detection process should be carefully addressed. Transparent and accountable practices should be followed to build trust in the system.

(vii) Education and Awareness: Alongside the development and implementation of automated cheating detection systems, educational institutions should focus on creating awareness and fostering a culture of academic integrity. Promoting ethical behavior, providing comprehensive guidelines on online examination protocols, and educating students about the consequences of cheating can significantly contribute to reducing cheating incidents.

By implementing these recommendations, the accuracy, effectiveness, and efficiency of the developed deep learning models can be improved, leading to more robust and

reliable systems for detecting and addressing cheating behavior during online examinations at universities in Kenya.

5.7 Further research

There are multiple exciting avenues for expanding upon the findings of this research. First, the system can be extended to operate seamlessly over the internet and to be a real time system. Additionally, a valuable enhancement would involve the integration of voiceprint recognition, during online-exam sessions. A feature for analyzing and comparing facial patterns should be thought about. System's security.

REFERENCES

- Alairaji, R. A. M., Aljazaery, I. A., & ALRikabi, H. T. S. (2022). Abnormal behavior detection of students in the examination hall from surveillance videos. In *Advanced Computational Paradigms and Hybrid Intelligent Computing: Proceedings of ICACCP 2021* (pp. 113-125). Springer Singapore.
- Alin, P., Arendt, A., & Gurell, S. (2023). Addressing cheating in virtual proctored examinations: toward a framework of relevant mitigation strategies. *Assessment & Evaluation in Higher Education*, 48(3), 262-275. DOI:10.1080-02602938.2022.2075317
- Alpaydin, E. (2010). *Introduction to machine learning, 2'nd ed.* The MIT Press,
- Arivazhagan, S., & Ligi, S. V. (2018). Mango leaf diseases identification using convolutional neural network. *International Journal of Pure and Applied Mathematics*, 120(6), 11067-11079. DOI:10.32604/cmc.2021.017700
- Ayachi, R., Said, Y., & Atri, M. (2021). A convolutional neural network to perform object detection and identification in visual large-scale data. *Big Data*, 9(1), 41-52. DOI:10.1089/big.2019.0093
- Bilen, E., & Matros, A. (2021). Online cheating amid COVID-19. *Journal of Economic Behavior & Organization*, 182, 196-211. DOI: 10.1016/j.jebo.2020.12.004
- Canarutto, D. (2010). Hermitian vector fields and covariant quantum mechanics of a spin particle. *International Journal of Geometric Methods in Modern Physics*, 7(04), 599-623. <https://doi.org/10.1142/S0219887810004464>
- Carlson, E. (2020). Field experiments and behavioral theories: science and ethics. *PS: Political Science & Politics*, 53(1), 89-93. DOI: <https://doi.org/10.1017/S1049096519001112>
- Dadzie, J., & Annan-Brew, R. (2023). Strategies for curbing examination malpractices: Perspectives of teachers and students. *Global Journal of Social Sciences Studies*, 9(1), 1-14. DOI:10.55284/gjss.v9i1.842
- Fiske, S. T., Lin, M., & Neuberg, S. L. (2018). The continuum model. *Social Cognition: Selected Works of Susan Fiske*, 41-75. <https://doi.org/10.4324/9781315187280>
- Franco, A. C., & Franco, L. S. (2022). An Institutional Theory Investigation: Analysis Of The Main Trends In Innovation. *Brazilian Journal of Management and Innovation (Revista Brasileira de Gestão e Inovação)*, 9(2), 126-144. DOI:10.18226/23190639.v9n2.06
- Garg, K., Verma, K., Patidar, K., & Tejra, N. (2020, May). Convolutional neural network based virtual exam controller. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 895-899). IEEE.
- Ghizlane, M., Hicham, B., & Reda, F. H. (2019, December). A new model of automatic and continuous online exam monitoring. In *2019 international conference on systems of collaboration big data, internet of things & security (SysCoBIoTS)* (pp. 1-5). IEEE.] doi: 10.1109/SysCoBIoTS48768.2019.9028027.

- Grice, J. B. (2018). *Emotional labor and workplace dishonesty: The role of emotional inauthenticity* [unpublished Doctoral dissertation, South University]. <https://repository.unair.ac.id/98787/9/9.%20DAFTAR%20PUSTAKA.pdf>
- Hosny, M., & Fatima, S. (2014). Attitude of students towards cheating and plagiarism: University case study. *Journal of Applied Sciences*, 14(8), 748-757. <https://scialert.net/abstract/?doi=jas.2014.748.757>
- Kaddoura, S., & Gumaei, A. (2022). Towards effective and efficient online exam systems using deep learning-based cheating detection approach. *Intelligent Systems with Applications*, 16, 73-74. DOI:10.1016/j.iswa.2022.200153
- Kingston, N., & Clark, A. (Eds.). (2014). *Test fraud: Statistical detection and methodology*. Routledge.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Langenfeld, T. (2020). Internet-based proctored assessment: Security and fairness issues. *Educational Measurement: Issues and Practice*, 39(3), 24-27. <https://doi.org/10.1111/emip.12359>
- Lee, J. W. (2020). Impact of proctoring environments on student performance: Online vs offline proctored exams. Lee, Jung Wan (2020). *Impact of Proctoring Environments on Student Performance: Online vs Offline Proctored Exams*. *Journal of Asian Finance Economics and Business*, 7(8), 653-660. <https://doi.org/10.13106/JAFEB.2020.VOL7.NO8.653>
- Leedy, PD, & Ormrod, JE (2015). *Practical research. Planning and design* . Pearson.
- Li, M., Luo, L., Sikdar, S., Nizam, N. I., Gao, S., Shan, H., ... & Wang, G. (2021). Optimized collusion prevention for online exams during social distancing. *npj Science of Learning*, 6(1), 5. DOI:10.1038/s41539-020-00083-3
- Li, M., Sikdar, S., Xia, L., & Wang, G. (2020). *Anti-cheating online exams by minimizing the cheating gain*.doi:10.20944/preprints202005.0502.v1
- Li, M., Sikdar, S., Xia, L., & Wang, G. (2020). Anti-cheating Online Exams by Minimizing the Cheating Gain. Preprints. <https://doi.org/10.20944/preprints202005.0502.v1>
- Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2014). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 865-873. <https://ieeexplore.ieee.org/document/6894591>
- Macharia, J. M. (2022). Systematic Literature Review Of Interventions Supported By Integration Of Ict In Education To Improve Learners'academic Performance In Stem Subjects In Kenya. *Journal of Education and Practice*, 6(3), 52-75. <https://eric.ed.gov/fulltext/EJ1182651.pdf>
- Maeda, M. (2019). Exam cheating among Cambodian students: When, how, and why it happens. *Compare: A Journal of Comparative and International Education*.1-19. 10.1080/03057925.2019.1613344
- Malhotra, M., & Chhabra, I. (2022). Student Invigilation Detection Using Deep Learning and Machine After Covid-19: A Review on Taxonomy and Future

- Challenges. *Future of Organizations and Work after the 4th Industrial Revolution: The Role of Artificial Intelligence, Big Data, Automation, and Robotics*, 311-326. <https://doi.org/10.3390/educsci13020194>
- Marreiros, G. (2022). Anomaly detection on natural language processing to improve predictions on tourist preferences. *Electronics*, 11(5), 779. DOI:10.3390/electronics11050779
- Mazodier, M., & Merunka, D. (2012). Achieving brand loyalty through sponsorship: the role of fit and self-congruity. *Journal of the Academy of Marketing Science*, 40, 807-820. <https://doi.org/10.1007/s11747-011-0285-y>
- McCabe, D. L., Butterfield, K. D., & Trevino, L. K. (2012). *Cheating in college: Why students do it and what educators can do about it*. JHU Press.
- Muchemwa, S. (2023). Integrity of university online assessment: Towards developing a function model. *Eureka: Journal of Educational Research*, 1(2), 67-73. DOI:10.56773/ejer.v1i2.8
- Mulongo, M. A., Kimosop, M. K., & Njoka, J. N. (2019). Assessment of mitigation strategies used in the management of examination malpractices by universities in Mount Kenya Region. *International Journal of Social Sciences & Educational Studies*, 6(2), 1-13. DOI:10.23918/ijsses.v6i2p1
- Nganchi, K. N., & Charlotte, M. N. (2020). Effects Of Examination Malpractices On Students Future, Case Of Higher Technical Teachers Training College, Bambili-Bamenda. *African Journal of Education and Practice*, 6(5), 55-78. <https://www.iprjb.org/journals/index.php/AJEP/article/download/1136/1250/3571>
- Nguyen, J. G., Keuseman, K. J., & Humston, J. J. (2020). Minimize online cheating for online assessments during COVID-19 pandemic. *Journal of Chemical Education*, 97(9), 3429-3435. DOI:10.1021/acs.jchemed.0c00790
- Nguyen, J. G., Keuseman, K. J., & Humston, J. J. (2020). Minimize online cheating for online assessments during COVID-19 pandemic. *Journal of Chemical Education*, 97(9), 3429-3435. <https://pubs.acs.org/doi/10.1021/acs.jchemed.0c00790>
- Noorbehhahani, F., Salehi, F., & Zadeh, R. J. (2019). A systematic mapping study on gamification applied to e-marketing. *Journal of Research in Interactive Marketing*, 13(3), 392-410. <https://doi.org/10.1108/JRIM-08-2018-0103>
- Pace, N. M., & Pollak, J. (2017). Provider Fraud in California Workers' Compensation.
- Reeve, J., & Halusic, M. (2009). How K-12 teachers can put self-determination theory principles into practice. *Theory and Research in Education*, 7(2), 145-154. DOI:10.1177/1477878509104319
- Rogers, M. K. (2006). A two-dimensional circumplex approach to the development of a hacker taxonomy. *Digital investigation*, 3(2), 97-102. DOI:10.1016/j.diin.2006.03.001
- Rubin, V. L. (2019). Disinformation and misinformation triangle: A conceptual model for "fake news" epidemic, causal factors and interventions. *Journal of documentation*, 75(5), 1013-1034. <https://doi.org/10.1108/JD-12-2018-0209>

- Saleh, A. M., & Meccawy, Z. (2021). EFL Female Students' Perceptions towards Cheating in Distance Learning Programmes. *English Language Teaching*, 14(1), 29-36. DOI:10.5539/elt.v14n1p29
- Sequeira, S., & Djankov, S. (2014). Corruption and firm behavior: Evidence from African ports. *Journal of International Economics*, 94(2), 277-294. DOI: 10.1016/j.jinteco.2014.08.010
- 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 preprint arXiv:2101.09841*. <https://arxiv.org/pdf/2101.09841.pdf>
- Tweissi, A., Al Etaiwi, W., & Al Eisawi, D. (2022). The Accuracy of AI-Based Automatic Proctoring in Online Exams. *Electronic Journal of e-Learning*, 20(4), 419-435. DOI: <https://doi.org/10.34190/ejel.20.4.2600>
- Valizadeh, M. (2022). Cheating In Online Learning Programs: Learners'perceptions And Solutions. *Turkish Online Journal of Distance Education*, 23(1), 195-209. <https://eric.ed.gov/?id=EJ1329626>
- Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 141, 61-67. <https://doi.org/10.1016/j.patrec.2020.07.042>
- Weick, K. E., Sutcliffe, K. M., & Obstfeld, D. (2005). Organizing and the process of sensemaking. *Organization science*, 16(4), 409-421. DOI:10.1287/orsc.1050.0133
- Yampolskiy, R. V., & Govindaraju, V. (2008). Behavioural biometrics: a survey and classification. *International Journal of Biometrics*, 1(1), 81-113. DOI:10.1504/IJBM.2008.018665
- Yulita, I. N., Fanany, M. I., & Arymurthy, A. M. (2017, September). Combining deep belief networks and bidirectional long short-term memory: Case study: Sleep stage classification. In *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)* (pp. 1-6). IEEE.
- Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., & Lee, B. (2022). A survey of modern deep learning based object detection models. *Digital Signal Processing*, 126, 103514. <https://doi.org/10.48550/arXiv.2104.11892>
- Zulaikha, S., Mohamed, H., Kurniawati, M., Rusgianto, S., & Rusmita, S. A. (2020). Customer predictive analytics using artificial intelligence. *The Singapore Economic Review*, 1-12. DOI:10.1142/S0217590820480021

APPENDICES

INTRODUCTION LETTER TO NACOSTI



KENYA METHODIST UNIVERSITY

P. O. Box 267 Meru - 60200, Kenya

Fax: 254-64-30162

Tel: 254-064-30301/31229/30367/31171

Email: deanrd@kemu.ac.ke

DIRECTORATE OF POSTGRADUATE STUDIES

April 5, 2023

Commission Secretary,
National Commission for Science, Technology and Innovations,
P.O. Box 30623-00100
NAIROBI.

Dear Sir/Madam,

RE: GABRIEL MUCHANGI – (REG. NO. CIS-3-0988-2/2017)

This is to confirm that the above named person is a bona fide student of Kenya Methodist University, in the School of Science and Technology, Department of Computer Science undertaking Master of Science Degree in Computer Science. He is conducting research on; "Behavioral Detection and Prevention of Cheating During Online Examination at Universities in Kenya Using Deep Learning Approach".

We confirm that his research proposal has been presented and approved by the University.

In this regard, we are requesting your office to issue a research license to enable him collect data.

Any assistance accorded to him will be highly appreciated.

Yours sincerely,



Dr. John M. Muchiri (PhD)
Director, Postgraduate Studies

Cc: Dean SST
CoD, CS
Program Coordinator - CS
Student Supervisors

NACOSTI LICENCE

Republic of Kenya
NATIONAL COMMISSION FOR SCIENCE, TECHNOLOGY AND INNOVATION

Ref No: 279486

RESEARCH LICENSE

Date of Issue: 25/April/2023

This is to Certify that Mr. GABRIEL GABRIEL MUCHANGI of , has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Embu, Kilifi, Kirinyaga, Kitui, Machakos, Makeni, Mombasa, Muranga, Nairobi, Taita-Taveta, Tharaka-Nithi on the topic: BEHAVIORAL DETECTION AND PREVENTION OF CHEATING DURING ONLINE EXAMINATION AT UNIVERSITIES IN KENYA USING DEEP LEARNING APPROACH for the period ending : 25/April/2024.

License No: NACOSTLP/23/25337

Applicant Identification Number: 279486

Director General
Walter Mburu
NATIONAL COMMISSION FOR SCIENCE, TECHNOLOGY & INNOVATION

Verification QR Code



NOTE: This is a computer generated License. To verify the authenticity of this document, Scan the QR Code using QR scanner application.

See overleaf for conditions