# Using CNN to Detect the Use of Mobile Phone in Examination Rooms

**Reuben Moyo**
University of Dar es Salaam, Tanzania
reubencmoyo@gmail.com

**Stanley Ndebvu**
Mzuzu University, Malawi
stanndebvu@yahoo.com

**Michael Zimba**
Malawi University of Science & Technology, Malawi
mgmzimba@gmail.com

**Jimmy Mbelwa**
University of Dar es Salaam, Tanzania
jimmymbelwa@gmail.com

## Abstract

Exams play a crucial role in the learning process, and academic institutions invest significant resources to ensure their integrity by preventing cheating by students or facilitators. Unfortunately, cheating has become rampant in exam environments, compromising their integrity. The traditional method of relying on invigilators to monitor every student is impractical and ineffective. It is necessary to record exam sessions to monitor students for suspicious activities to address this challenge. However, these recordings are often too lengthy for invigilators to analyse effectively, and fatigue may cause them to miss significant details. To expand the coverage, invigilators can use fixed overhead or wearable cameras. This paper introduces a model that automates the analysis videos to effectively detect the use of mobile phones as during exams. We used Convolutional Neural Networks (CNN) object detection techniques to identify mobile phones. The experimental results show that model achieved a 98.9% accuracy, a recall of 0.795, an F-measure of 0.697, and an average precision of 0.783. This detection system is essential in preventing cheating and promoting academic integrity, fairness, and quality education for institutions.

## 1 Introduction

The use of advanced Information and Communication Technology (ICT) in education has significantly impacted the field's growth. ICT is utilized in teaching, learning, and exam administration [1], among other areas. Exams are a crucial part of a student's learning, and academic institutions put in a lot of effort and resources to ensure the integrity of the exam. Despite the presence of a proctor, students may still cheat cautiously to avoid harsh consequences[2]-[4]. Using security cameras and object detection techniques, we can record videos and analyse suspicious use of prohibited materials in exam sessions [5]. These videos can contain information that even invigilators may miss it. However, going through each video manually is time-consuming and arduous. Automating the analysis and evaluation of the videos and highlighting any suspicious activities would be helpful. Most modern lecture halls have cameras installed in strategic locations for security purposes. Similarly, these cameras can record and analyse any unusual or suspicious activity during exams, along with wearable cameras if necessary. This paper discusses a framework that automatically analyses pre-recorded exam videos and detects prohibited materials like the use of phones during an examination. The use of mobile phones is not allowed in the examination rooms; hence, it is considered a prohibited material.

5th Deep Learning Indaba Conference (DLI 2023).

Cheating in exams has become widespread, with many students sneaking on mobile phones to look up answers on the Internet. Although screening students when entering an exam hall is standard practice, invigilators sometimes fail to detect these devices [3]. It is essential to detect them during the exam to prevent cheating effectively. This is why an object detection model using Object Detection techniques is necessary [15],[19].

In computer vision, image classification involves assigning a single label to an image, placing it into classes or categories. It seeks to answer the question, "What is in this image?" Object detection, on the other hand, is concerned with identifying and locating multiple objects within an image. It is an extension of image classification by answering, "What objects are in this image?" and "Where are they located?" [13]. The decision to use one method over the other depends on the specific application and whether the goal is to identify specific objects and their placement in an image or categorize the entire image into predetermined classes. Detecting objects is one of the most difficult challenges in Computer Vision (CV). It requires identifying and localizing objects within a particular scene whilst labelling different objects and drawing bounding boxes around them [10],[22]. Deep Neural Networks (DNNs), especially Convolutional Neural Networks (CNNs), have greatly improved the training of models for OD [9],[22], [27]. The Fast/Faster R-CNN frameworks have played a crucial role in OD by providing flexibility, robustness, training, and inference time [24],[25]. To identify the location of an object in an image and draw a bounding box around its extent, we utilise object localization techniques. Instance Segmentation requires precise detection of objects in an image. Each Region of Interest (RoI) is predicted, and a segmentation mask is drawn. Faster R-CNN has two stages; the first is a Region Proposal Network (RPN) that suggests potential object bounding boxes. The second stage involves feature extraction using the RoI Pool for each candidate box. Then bounding box regression and classification are performed. Mask R-CNN is a two-stage object detector that follows the same procedure as Faster R-CNN, with an identical RPN stage first, and then it outputs a binary mask for each RoI along with class prediction and bounding box offset [28], [20].

The MMDetection library, presented by Chen et al. [21], is a Python-based open-source library that offers top-notch object detection techniques with a high performance, memory efficient, a fast training speed and high accuracy. The MaskRCNN benchmark includes Mask R-CNN [20], RetinaNet, Faster R-CNN [25], RPN, Fast R-CNN [24], and a specially designed Python code. This combination of state-of-the-art algorithms into a single framework significantly increases performance and accuracy, as MMDetection Model Zoo and Baselines demonstrated. MMDetection supports VOC-style and COCO datasets and is particularly noteworthy for its fast-training capability. To train an object detector using MMDetection, the main units, such as a data pipeline, an iterative pipeline, and a model, must be defined [33]. This research follows a similar approach to train a custom object detector.

Our research aims to address the limitations of current fraud detection systems by exploring the effectiveness of advanced CNN models such as the MMDetection Toolbox, Mask R-CNN, and Openpose library in detecting suspicious activities without requiring a physical proctor. After careful consideration, we used the MMDetection library for phone detection due to its range of models, including Mask R-CNN, Faster R-CNN, and RetinaNet.

### 1.1 Objectives

This study intended to analyse pre-recorded videos of an exam session to detect the presence of prohibited objects like mobile phones in an exam to assist a physical invigilator in continuously monitoring the students.

## 2 Methodology

### 2.1 Dataset creation

Since no existing dataset depicted people using phones in an exam room, we had to create one. To do this, we recorded videos of a simulated exam room setting. Sixteen students were randomly chosen to participate in a video recording experiment. We arranged the students in four rows and columns to create a realistic exam environment. The experiment consisted of ten trials, with the first six allowing phone usage during each trial. Two trials were used as a control, where all students followed strict exam rules. In the final four trials, participants engaged in random activities. A custom

| Parameter | Settings |
|---|---|
| The number of Epochs | 50 |
| Number of Classes | 2 |
| Pre-trained Backbone | Resnet50 |
| Configuration Model | faster-rcnn-r50-fpn _1x |
| Learning Rate | 0.01 |
| Model Type | Faster RCNN |
| Number of filters | 18 |

Table 1: Summary of parameter settings used during the fine-tuning of the custom network

dataset was created by sampling frames from the recorded videos. To obtain a dataset large enough to train a neural network, We collected additional images from the web and successfully labelled and annotated them. The frames were resized to 224 x 224, and we created a dataset of 10,000 images. Approximately 6,600 images had a phone on it. The dataset comprised smartphones and second generation phones. Generating a custom dataset is a time-consuming process requiring much effort. Nonetheless, we found that utilizing pre-trained models trained on datasets with comparable classes to ours was advantageous. We utilized an adaptable open-source image and video labeller tool, OpenLabeling [21], to aid us with fast and flexible image labelling and dataset annotation. The resultant dataset includes 7,000 images for training, 1,500 for testing, and 1,500 for validation. The validation dataset helps evaluate the model's quality, prevents over- and under-fitting, and assists in selecting the best model for unseen data.

### 2.2 Design and Evaluation

To detect the presence of cell phones in the exam room, we employed an existing MMDetection library with Faster R-CNN. This CNN framework was pre-trained on the state-of-the-art COCO and Pascal VOC-Style dataset. However, the pre-trained network performed poorly on the videos because of the angle of capture for the phone and the small size of the phone object. It proved challenging to recognise a phone in the student's hands. Secondly, the benchmark datasets did not consist of adequate classes of mobile phones. Hence, fine-tuning the CNN network and creating our custom data set was vital. Much as the students may hide their mobile phones, the model should be able to detect it at the slightest chance that the camera captures the phone. First, images used for training, testing, and validation were re-sized to 224 * 244 pixels to fine-tune the network. We use this custom data set to train our fine-tuned network. The last three network layers with the FC, Softmax, and classification output layers were replaced. The FC layer was set to have one (1) class as in the training data set. The learning rate factor of the FC layer was increased to train the network faster. Then, We set the training options, including a learning rate of 0.01, 50 epochs, and validation data on a single GPU system. During the training of the network, we used the Adam optimizer because it uses fewer memory requirements and is efficient as compared to stochastic gradient descent [19], [23], [8]. Lastly, we classified the testing and the validation images using the fine-tuned network and calculated the classification accuracy.

To train the network on a custom data set, we created a pipeline with eight operations, including LoadImageFromFile, LoadAnnotations, Resize, RandomFlip, Normalize, Pad, DefaultFormatBundle, and Collect. These operations handle data loading, pre-processing, formatting, and test-time augmentation, as described in [33]. We modified the Neck of the network by adding a Feature Pyramid Network (FPN) layer in the CONV layer. Next, we adjusted the RPN Head by removing some classification layers to produce bounding boxes for the cell phone class only. For bounding box RoI extraction, we used the singleRoIExtractor type. Finally, we fine-tuned the model on a single GPU machine based on the parameters listed in Table 1.

## 3 Results and Discussion

During the experiment, the goal was to identify any mobile phones present in the exam room. While the model could detect visible phones, it did not catch all phones in a single frame. The accuracy rate

(a) Training results for accuracy and total loss

(b) Training results for classification loss, bounding box loss

Figure 1: Training results for classification loss, bounding box loss, accuracy and total loss for the fine-tuned model

| Class | Ground Truths | Detections | Recall | Avg Precision |
|---|---|---|---|---|
| Cellphone | 19 | 800 | 0.368 | 0.057 |
| Mean Avg Precision | | | | 0.057 |

Table 2: Results of Mean Average Precision (mAP) of the generic MMDetection when run on the custom dataset

of the detection model was 98.9%. However, the model faced obstacles when trying to detect phones firmly held by the students, leading to decreased accuracy.

**Scenario 1: Use of MMDetection model:** To detect phones in pre-recorded videos, we utilized an unmodified Faster R-CNN Network from the MMDetection toolbox. In figures (a) and (b) of Figure 1, we showcase some of the results obtained using the generic MMDetection models on the data. However, detecting a phone proved challenging for the generic MMDetection model when the phone was partially concealed or held in hand. The model was trained to recognize phones in full view, but it struggled to detect phones from different angles. As a result, we encountered difficulties when attempting to detect and classify phones that were partially visible or hidden under other objects.

Additionally, there were some difficulties with misclassifying objects, such as pens being identified as cell phones, and all objects in the 21 classes of the VOC dataset being classified. Furthermore, unwanted objects like people, chairs, and books were also categorized. The bounding box classification loss on the object was high, with a low mAP of 0.057, and a recall of only 0.368 from 19 ground truths, with 800 misclassified detections. This experiment was not intended to assess the performance of MMDetection, but rather to determine if it could be utilized for our cell phone detection problem.

**Scenario 2: Use of Fine-tuned CNN Network:** To overcome the challenges with generic MMDetection models, we fine-tuned a custom CNN network to achieve our objective. Our model was trained based on a custom dataset, and we obtained excellent results when running it on pre-recorded videos, as shown in Figures 2 (a) and 2 (b). The fine-tuned network performed exceptionally well, detecting numerous occurrences of cell phones even based on ground truth observation. The model classified smaller phone objects with higher accuracy and a better classification loss. Additionally, the bounding box loss was minimal, as shown in the graph in Figure 3 (b).

Despite issues with the created custom model, it effectively detected cell phones and scientific calculators. However, there were instances where the model struggled to differentiate between objects that looked like cell phones and actual cell phones, while scientific calculators were sometimes identified as cell phones. Additionally, the model wasn't extensively tested on various datasets and videos due to time constraints, and it was only trained using a small sample size of 10,000 images due to the meticulous and time-consuming task of annotating frames.

(a) Front view (b) Rear view

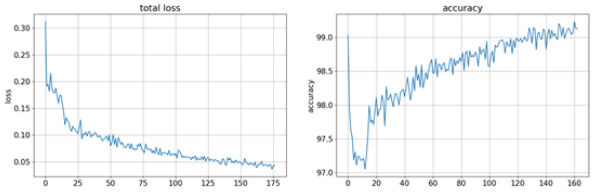
Figure 2: A front (a) and top-rear (b) view of a classified phone by the fine-tuned model

| Class | Ground Truths | Detections | Recall | Avg Precision |
|---|---|---|---|---|
| Cellphone | 19 | 18 | 0.795 | 0.783 |
| Mean Avg Precision | | | | 0.783 |

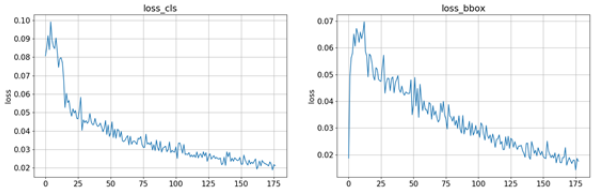
Table 3: Results of the mAP of the fine-tuned model when run on the custom data set and videos

To evaluate the model's performance, a random video was selected and tested, and it performed better than the generic model, achieving a mean average precision (mAP) of 0.783 and a recall of 0.795. The model detected 18 out of 19 ground truths, as shown in Table 3, indicating an improvement in object classification. The fine-tuned model also achieved an F-measure of 0.697, which indicates better performance. Although the precision rate wasn't as high and the difference between precision and recall wasn't significant, the F-score suggests that the model's classifier was precise and robust enough to be effective.

In Figure 2 (a), the fine-tuned model shows an accuracy of 98.9%. We also rely on the F-score to determine the detector's precision and robustness. The total loss decay indicates that the fine-tuned model had better training performance and behaved normally. The model successfully identified the region proposals, classified the phone object from the background, and accurately classified the bounding box RPN of the phone object.




(a) Training results for accuracy and total loss




(b) Training results for classification loss, bounding box loss

Figure 3: Training results for classification loss, bounding box loss, accuracy and total loss for the fine-tuned model

### 3.1 Ethical Consideration

In this study, ethical issues cannot be disregarded. One of the issues is privacy. The privacy of the participants in the experiment was significantly considered by restricting the scope for more domain-specific actions. All experiment subjects were provided with a consent form as stipulated by [11],[18],[26]. Furthermore, data was securely stored and only used for the intended purpose. Throughout the research process, transparency in the models and algorithms was critical to make the decision-making processes clear and understandable.

## 4 Conclusion

We fine-tuned an existing CNN network to detect cell phones in the exam room. In an event where the camera captures a cell phone, the model detects it as a prohibited material. Based on the custom data set created, the model was trained to detect small-sized cell phones from different sides and angles of the cell phone to maximize the chances for detection. Eventually, our model performed well in our data on study and achieved an accuracy of 98.9%. Much as not all occurrences of phones in a video could be detected; the model achieves its objective. The model achieved a 98.9% accuracy, a recall of 0.795, an F-measure of 0.697, and an average precision of 0.783. This detection system is essential in preventing cheating and promoting academic integrity, fairness, and quality education for institutions. A significant obstacle in identifying cheating during exams is students' resourcefulness in concealing their phones as they are adaptive adversarial agents. They are resourceful and adept at disguising or stashing phones under their desks, and they become aware of camera placements. To overcome this issue, we propose utilizing adversarial training models, which incentivise students with rewards to produce training data sets that assist the model in detecting cheating, even as they attempt to circumvent detection. Additionally, combining a self-supervised model with an adversarial-trained model would enhance the feasibility and efficacy of detecting cheating during exams.

## References


[1] J. Sheard, Simon, M. Butler, K. Falkner, M. Morgan, and A. Weerasinghe, *"Strategies for Maintaining Academic Integrity in First-Year Computing Courses,"* in Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education, 2017, pp. 244–249, doi: 10.1145/3059009.3059064.

[2] K. Curran, G. Middleton, and C. Doherty, *"Cheating in Exams with Technology,"* Int. J. Cyber Ethics Educ., vol. 1, no. 2, pp. 54–62, 2011, doi: 10.4018/ijcee.2011040105.

[3] S. Davis, Patrick, and Drinan, *Cheating In School: What We Know & What We Can Do.* Wiley-Blackwell, 2009.

[4] E. M. Anderman and T. B. Murdock, *"The Psychology of Academic Cheating,"* Psychol. Acad. Cheating, pp. 1–5, 2007, doi: 10.1016/B978-012372541-7/50002-4.

[5] E. R. Cavalcanti, C. E. Pires, E. P. Cavalcanti, and V. F. Pires, *"Detection and evaluation of cheating on college exams using supervised classification,"* Informatics Educ., vol. 11, no. 2, pp. 169–190, 2012.

[6] M. Abdaoui, *"Strategies for Avoiding Cheating and Preserving Academic Integrity in Tests,"* Alkhitab w el-Tawassol J., vol. 4, no. July 2018.

[7] K. A. D'Souza and D. V Siegfeldt, *"A Conceptual Framework for Detecting Cheating in Online and Take-Home Exams,"* Decis. Sci. J. Innov. Educ., vol. 15, no. 4, pp. 370–391, 2017, doi: 10.1111/dsji.12140.

[8] T. Liu, S. Fang, Y. Zhao, P. Wang, and J. Zhang, *"Implementation of Training Convolutional Neural Networks,"* arXiv pre-prints, 2015, arXiv:1506.01195.

[9] L. Jiao et al., *"A survey of deep learning-based object detection,"* IEEE Access, 2019, 7(3): 128837–128868.

[10] L. Liu et al., *"Deep Learning for Generic Object Detection: A Survey,"* International Journal of Computer Vision, 2020, 128(2): 261–318.

[11] S. L. Colyer et al., *"Legal Implications of Using AI as an Exam Invigilator,"* SSRN Electron. J., vol. 75, no. 1, pp. 333–347, 2019, doi: 10.1111/hequ.12275.

[12] S. Coghlan, T. Miller, and J. Paterson, *"Good proctor or 'Big Brother'? AI Ethics and Online Exam Supervision Technologies,"* no. Ml, pp. 1–14, 2020, [Online]. Available: http://arxiv.org/abs/2011.07647.

[13] C. Chen, M. Y. Liu, O. Tuzel, and J. Xiao, *"R-CNN for small object detection,"* in Asian Conference on Computer Vision, Springer, Cham, 2016: 214–230.

[14] N. Jmour, S. Zayen, and A. Abdelkrim, *"Convolutional neural networks for image classification,"* in Proceedings of 2018 International Conference on Advanced Systems and Electric Technologies (ICASET), 2018: 397–402.

[15] E. R. Cavalcanti, C. E. Pires, E. P. Cavalcanti, and V. F. Pires, *"Detection and evaluation of cheating on college exams using supervised classification,"* Informatics in Education, 2012, 11(2):169–190.

[16] K. Hylton, Y. Levy, and L. P. Dringus, *"Utilizing webcam-based proctoring to deter misconduct in online exams,"* Computers Education, 2016, 92(93): 53–63.

[18] K. A. D'Souza and D. V. Siegfeldt, *"A Conceptual Framework for Detecting Cheating in Online and Take-Home Exams,"* Decision Sciences Journal of Innovative Education, 15(4): 370–391.

[19] H. C. Shin et al., *"Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning,"* IEEE Transactions Medical Imaging, 2016, 35(5): 1285–1298.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, *"Mask R-CNN,"* in 2017 IEEE International Conference on Computer Vision (ICCV), Open Access, 2017: 2961–2969.

[21] K. Chen et al., *"MMDetection: Open MMLab Detection Toolbox and Benchmark,"* arXiv preprint, 2019, arXiv:1906.07155. [21] R. Girshick, "Fast R-CNN [J]," arXiv preprint, 2015, arXiv: 1504.08083v2-1.

[22] L. Jiao et al., *"A survey of deep learning-based object detection,"* IEEE Access, 2019, 7(3): 128837–128868. [23] L. Liu et al., *"Deep Learning for Generic Object Detection: A Survey,"* International Journal of Computer Vision, 2020, 128(2): 261–318.

[24] R. Girshick, *"Fast R-CNN,"* in 2015 IEEE International Conference on Computer Vision (ICCV), 2015:1440–1448. [25] S. Ren, K. He, R. Girshick, and J. Sun, *"Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,"* IEEE Trans. Pattern Anal. Mach. Intell., 2017, 39(6): 1137–1149.

[26] Y. Lin and J. Zhou, *"Detection method for cheating behavior in examination room based on artificial bee colony algorithm,"* in Proceedings of 2015 International Conference on Informative and Cybernetics for Computational Social Systems (ICCSS), Chengdu, 2015: 122–125.

[27] N. Sharma, V. Jain, and A. Mishra, *"An Analysis of Convolutional Neural Networks for Image Classification,"* Procedia Computer Science, 2018, 132(Iccids): 377–384.