Visual Computing Magazine

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

1. Introduction

In the evolving landscape of education, the integration of technology presents both opportunities and challenges, with academic integrity facing new threats. As institutions embrace innovative assessment methods, the rise of academic dishonesty necessitates robust solutions. Our collaborative effort focuses on designing a Cheating Detection System (see figure 1), leveraging cameras and Convolutional Neural Networks (CNNs) to identify and deter cheating during examinations.

This paper explores the project's conception, design, and implementation stages. In the context of modern educational challenges, we highlight the significance of our Cheating Detection System. Combining NVIDIA Jetson Nano with deep learning algorithms, our approach provides a comprehensive solution to mitigate dishonest behavior during exams.

Figure 1. Cheating by passing paper, sheet on hand

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

This paper not only addresses immediate concerns of cheating detection but contributes to the broader discourse on maintaining academic integrity in the technological age. As technology continues to reshape education, our commitment to upholding assessment sanctity remains steadfast, exemplified by our Cheating Detection System—a stride towards fortifying academic integrity.

2. Method

2.1 Dataset preparation

In the realm of single-board computers crucial to the development of our project, both the Jetson Nano and Raspberry Pi Camera play integral roles. Developed by NVIDIA, the Jetson Nano is a high-performance single-board computer tailored for Artificial Intelligence (AI) and deep learning applications. Its robust computational power makes it well-suited for tasks demanding advanced processing capabilities.
The Raspberry Pi Camera, designed explicitly for use with Raspberry Pi single-board computers, boasts essential features for our project, including an 8-megapixel camera capable of capturing photographs at 3280 x 2464 pixels and video capture

To capture a diverse range of scenarios involving both cheating and non-cheating actions, a multitude of images were meticulously captured during various simulated scenarios.
The next phase involved processing these images using YOLO (You Only Look Once) (see figure 2). Each frame, depicting instances of cheating or non-cheating actions, underwent YOLO to identify and establish bounding boxes around the students. Subsequently, these bounding boxes were used to crop the images, and the cropped results were locally saved.

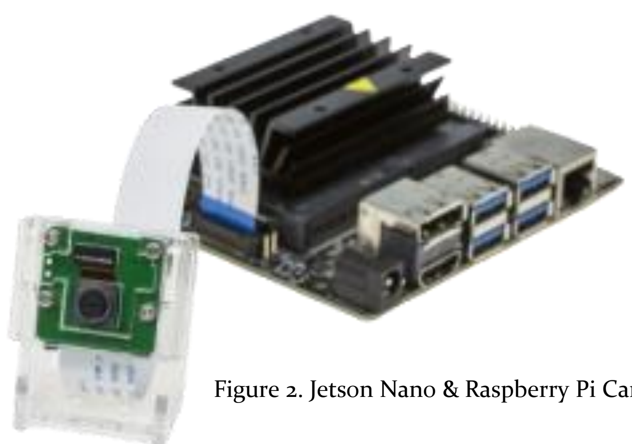

Figure 2. Jetson Nano & Raspberry Pi Camera

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

Following the image cropping process, the dataset underwent meticulous labeling. The cropped students' images, identified using YOLO, were manually separated into distinct folders dedicated to cheating and non-cheating instances (see figure 3).

Augmentation

While the initial dataset provided a solid foundation, we recognized the need to augment the data for increased variability. To achieve this, a data augmentation process was implemented to introduce another batch of variation into our dataset.
The data augmentation process was a strategic play on diverse transformations. Techniques such as applying a median filter, blurring, and rotating the images were systematically employed. These augmentation methods not only expanded the dataset but also introduced variations in lighting, orientation, and other factors, ensuring a more robust and comprehensive training set for our Cheating Detection System. This section explores the significance of data augmentation in fortifying the dataset and, consequently, the overall efficacy of our innovative cheating detection solution.

Figure 3. Cheating images - results of cropping

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

2.2 Model Architecture

To find an optimal model architecture we began with the utilization of the VGG Net architecture. However, confronted with the practical constraints of embedded systems, particularly a 1 GB model deemed impractical for our project, we embarked on a strategic shift. The chosen solution was a more lightweight approach—a Convolutional Neural Network (CNN) with two layers comprising 32 and 64 convolutions, aligning with the project's embedded system requirements.

In response to the limited dataset, we introduced a Max Pooling layer and a Dropout layer to enhance the model's generalization capabilities. The final refinement involved incorporating a dense layer with 256 neurons, culminating in a binary decision neuron. This adjustment in the model architecture represents a delicate balance between computational efficiency and performance robustness, ensuring optimal functionality within the constraints of our Cheating Detection System. This section delves into the rationale behind these adaptations, elucidating how the refined model strikes a harmonious balance to meet the specific demands of our innovative cheating detection solution.

Augmenting Data Resources and Streamlining Annotation

Additional videos and images were systematically captured to bolster our resources. This expansion aimed to ensure a diverse and comprehensive set of data for training and refining our Cheating Detection System.

To streamline the annotation process for object detection, we employed Roboflow—an efficient tool in our workflow. Focusing on two distinct classes, namely "cheating" and "not_cheating," we undertook the manual annotation task meticulously. This section provides insights into the rationale behind enriching our dataset and the strategic use of Roboflow for effective annotation, contributing to the robustness and accuracy of our innovative cheating detection solution.

Seamless Data Integration and Enhanced YOLO Architecture

To seamlessly integrate our annotated data into the Darknet framework, we leveraged Roboflow, a versatile tool that supports the download of data in various formats. Crucially, we adopted the YOLO Darknet format. With this approach, every image corresponds to a text file sharing the same name, housing crucial information such as class labels and the corresponding object box coordinates in the format: <class> <x> <y> <w> <h>. This labeling process sets the stage for optimal utilization of the Darknet framework in our Cheating Detection System.

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

Optimized YOLO Architecture

Our commitment to refining the YOLO architecture led to a strategic optimization with a tailored approach. The enhanced YOLO architecture comprises 38 layers, featuring two detection layers with distinct scales. A key innovation is the incorporation of a CSP connection, further enhancing the model's capacity for detecting and classifying instances. This illuminates the rationale behind these optimizations, shedding light on how the streamlined YOLO architecture enhances the precision and efficiency of our innovative cheating detection solution

Empowering YOLO Detection

Harnessing the power of our dataset and the Darknet framework, we embarked on the training phase for our Yolov4-tiny model (see figure 4). The model underwent 6000 iterations, aligning with the recommended minimum for effective training. A discerning analysis of the training process reveals a consistent decline in the average loss, a pivotal indicator of the model's adept learning. This phenomenon is vividly illustrated on the training chart, showcasing the model's progressive improvement.

Remarkably, the mean Average Precision (mAP) achieved an impressive 98% accuracy, further affirming the robustness and efficacy of our Yolov4-tiny model (see figure 5). This section unfolds the comprehensive journey of training and validation, emphasizing key performance metrics and the remarkable learning curve observed during the training process for our innovative cheating detection solution (see figure 6).

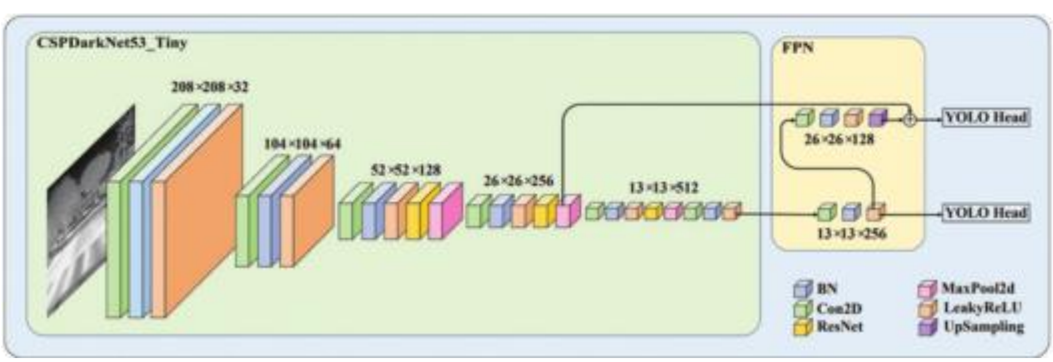

Figure 4. YOLO v4 tiny architecture

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

3. Results

We tested our model using full frames instead of cropped ones on our custom data. The results turned out pretty good, encouraging us to broaden our testing scope by including online images.

The model performed well in diverse testing conditions, showing adaptability and accuracy. Expanding our evaluation to include online images allowed us to further validate its effectiveness in real-world scenarios. This section provides insights into our testing process, highlighting the model's performance across different datasets and affirming its practical applicability in various situations.

Figure 5. Results - Few students not cheating

Conclusion

In conclusion, our cheating detection project employed two methods—CNN and YOLOv4 Tiny using Darknet—revealing that the latter surpassed the former in terms of effectiveness. The utilization of YOLOv4 Tiny showcased superior results, highlighting its efficacy as a robust approach for cheating detection compared to the CNN method. This outcome underscores the significance of leveraging advanced object detection techniques, particularly within the framework of YOLOv4 Tiny, to enhance the accuracy and reliability of cheating detection systems.

To enhance the camera's performance, we opted for C language over Python, prioritizing speed and smooth operation. The decision to use C language contributed to a faster and smoother camera operation, ensuring real-time efficiency in our cheating detection system.

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

It's worth noting that the first method, utilizing CNN, was primarily applied to images of one person, whereas the second method, employing YOLOv4 Tiny, delivered superior results when applied to full frames instantaneously. This distinction highlights the versatility and instantaneity offered by the YOLOv4 Tiny approach, making it a preferred choice for comprehensive cheating detection.

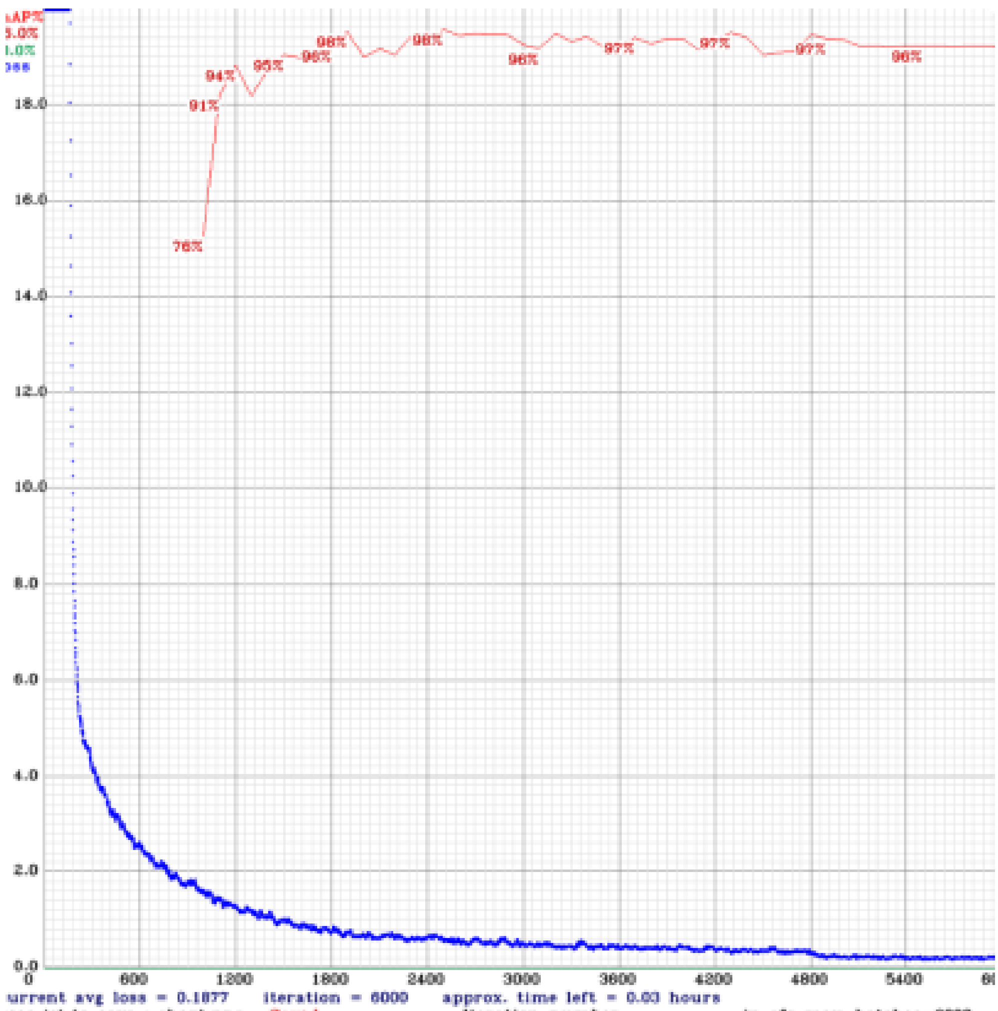

Figure 6. Loss and accuracy of the model

Embedded Cheating Detection with NVIDIA Jetson Nano

Mohammed Kadri, Zohdi Kilani, Nassim Kaddouri, Ilyes Djebara, Alaa Guermat Computer Science Faculty, USTHB University.

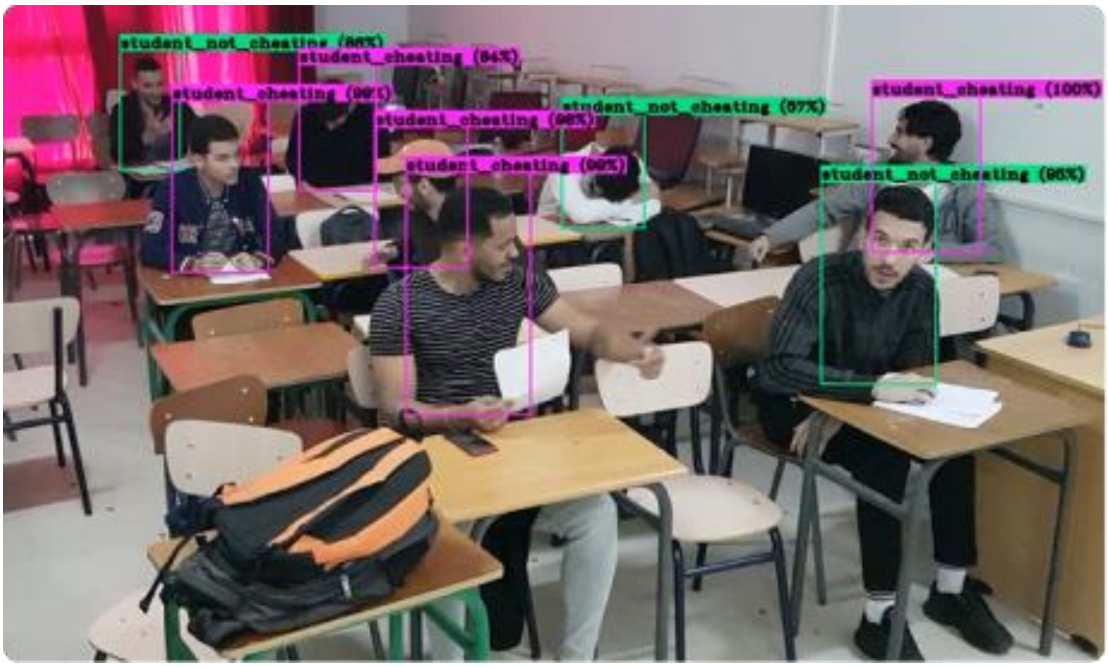

Figure 7. Results - Few students cheating

References

[1] Jacob Solawetz, Samrat Sahoo.
Train YOLOv4-tiny on Custom Data - Lightning Fast Object Detection.
https://blog.roboflow.com/train-yolov4-tiny-on-custom-data-lighting-fast-detection/
[2] NVIDIA Jetson Nano
https://www.nvidia.com/fr-fr/autonomous-machines/embedded-systems/jetson-nano/
[3] https://www.raspberrypi.com/documentation/accessories/camera.html