2023 5th International Conference on Energy, Power and Environment: Towards Flexible Green Energy Technologies (ICEPE) | 979-8-3503-1312-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEPE57949.2023.10201506

Weapon Object Detection Using Quantized YOLOv8

1st Muralidhar Pullakandam
Department of Electronics and Communication NIT Warangal
National institute of technology Warangal
Warangal,India
pmurali@nitw.ac.in

2nd Keshav Loya
Department of Electronics and Communication NIT Warangal
National institute of technology Warangal
Warangal,India
loya_911916@student.nitw.ac.in

3rd Pranav Salota
Department of Electronics and Communication NIT Warangal
National institute of technology Warangal
Warangal,India
salota_811946@student.nitw.ac.in

4rd Rama Muni Reddy Yanamala
Deparment of Electronics and Communication NIT Warangal
National institute of technology Wranagal
Warangal,India
yanamalamunireddy@gmail.com

5th Pavan Kumar Javvaji
Department of Electronics and Communication NIT Warangal
National institute of technology
Warangal,India
javvaj_921915@student.nitw.ac.in

Abstract— **Video surveillance is essential for creating a secure and hassle-free environment in all areas of life. It helps identify theft, detect unusual events in crowded locations, and monitor the suspicious behavior of individuals. However, monitoring surveillance cameras manually is quite challenging, and thus, fully automated surveillance with smart video-capturing capabilities is gaining popularity. This approach uses deep learning methodology to remotely monitor unusual actions with accurate information about the location, time of occurrence, and identification of criminals. Detecting criminal conduct in public settings is difficult due to the complexity of real-world scenarios. CCTV cameras can record suspicious incidents in public areas, such as carrying weapons, which helps authorities to take preventive measures to protect citizens. The proposed system employs the state-of-the-art YOLOv8 model for real-time weapon detection, which is faster, more accurate, and better than YOLOv5. To ensure fast performance, the weights of YOLOv8 were quantized. In our experiments, we evaluated the performance of the YOLOv8 and YOLOv5 models for weapon detection. The mean Average Precision (mAP) value achieved using YOLOv8 was 90.1%, which outperformed the mAP value of 89.1% obtained with YOLOv5. Furthermore, by applying weight quantization to the YOLOv8 model, we reduced the inference time by 15% compared to the original YOLOv8 configuration.**

Keywords— YOLO, Roboflow, mAP, Weight Quantization, Inference time.

I. Introduction

In recent years, the use of weapons in public places such as schools, malls, and airports has resulted in numerous tragedies. Aiming to reduce such kinds of incidents, automatic weapon detection systems have been developed. The use of deep learning techniques, particularly object detection algorithms, has successfully detected weapons in real-time videos and images. Currently, the detection of weapons in public areas involves the use of sensors that are designed to identify potentially suspicious items. However, these sensors are costly, inefficient, and have limited coverage. To address these shortcomings of the traditional approach, we are turning our attention to object detection using deep learning algorithms, which are more effective than relying solely on sensors.

In this research paper, we will focus on one such algorithm, the YOLOv8 (You Only Look Once version 8) [1], which is an improvement over the popular YOLOv5 [2]. We will examine the effectiveness of YOLOv8 in detecting weapons in different situations, such as in crowded places, low light conditions, and with different types of weapons.

The YOLOv5 model utilizes the detection architecture of YOLO and leverages multiple algorithmic optimization techniques from the convolutional neural network domain. Comprised of three primary components, including the backbone, neck, and output, the YOLOv5 network begins by executing data pre-processing tasks such as mosaic data augmentation and adaptive image filling at the input terminal. YOLOv5 also integrates adaptive anchor frame calculation on the input to enable adaptation to various datasets, thus automatically establishing the initial anchor frame size when the dataset changes. Now Coming to YOLOv8, it has better accuracy than previous YOLO models. The latest YOLOv8 implementation comes with a lot of new features, especially the user-friendly CLI and GitHub repo. It also supports object detection, instance segmentation, and image classification. One of the major changes that came from YOLOv5 to YOLOv8 was the change in the backbone of the system, which changed with the introduction of the C2f block, replacing the C3 block. Fig.1 shows the architecture of C2f and C3 blocks. Another major change was the introduction of an anchor-free box. Anchor-free detection is when an object detection model directly predicts the center of an object instead of the offset from a known anchor

979-8-3503-1312-3/23/$31.00 ©2023 IEEE

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on October 26,2024 at 07:16:45 UTC from IEEE Xplore. Restrictions apply.

box. It has the added advantage that it is more flexible and efficient, as it does not require the manual specification of anchor boxes, which can be difficult to choose and can lead to suboptimal results in previous YOLO models [3].

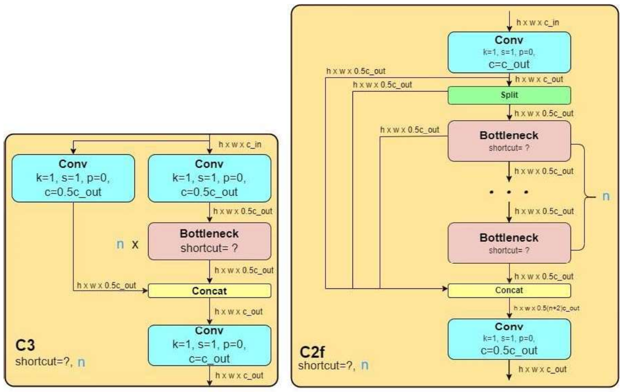

Fig.1. C2f and C3 blocks in YOLOv8 [4]

The main focus of this research paper is gun and pistol detection in public places. When employing video surveillance that encompasses multiple real-time objects, it can be exceedingly challenging to identify a particular object of interest. However, the YOLOv8 algorithm has demonstrated the ability to accomplish this task with high efficiency and accuracy, particularly in the realm of crowd analysis. The research findings will provide valuable insights into the effectiveness of using YOLOv8 and other deep learning techniques for automatic weapon detection systems. This research paper will be of interest to researchers, practitioners, and law enforcement agencies seeking to enhance the security and safety of public spaces by implementing automatic weapon detection systems.

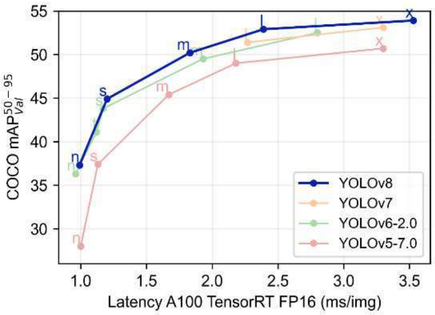

Fig.2. shows the performance of YOLOv8 [5]

Fig.2 shows the improvement in the performance of the YOLOv8 when compared to all the previous versions on the COCO dataset, and this tells us about the greater significance of the highest version while applying the YOLO architecture to a particular dataset.Further, using a hardware accelerator for CNN like [6,7] in YOLO models will further speed up the detection speed.

II. Literature Survey

Computer vision is used extensively in the fields of public safety and security, including weapon detection. Modern object identification algorithm YOLOv5 has demonstrated remarkable accuracy and real-time performance. A modified version of the YOLOv5 algorithm designed to operate on devices that possess restricted resources is referred to as the YOLOv5small model. YOLOv5small has been applied for weapon identification in several sectors in recent years. The architecture of YOLOv5 is given below in Fig.3.

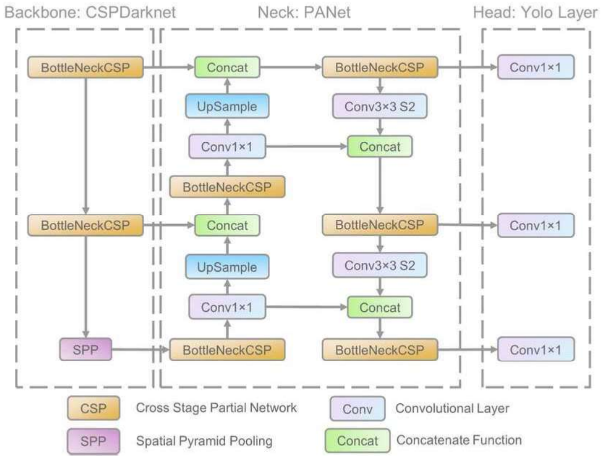

Fig.3. Architecture of YOLOv5 [8]

Tahir Mahmood et al. proposed a YOLO-based deep learning model to detect weapons in real-time CCTV Videos in their paper " Weapon Detection in Real-Time CCTV Videos using Deep Learning." They used various algorithms like VGG16, Inception-V3, Inception-ResnetV2, SSDMobileNetV1, Faster-RCNN Inception-ResnetV2 (FRIRv2), YOLOv3, and YOLOv4 and found that YOLOv4 gave the best accuracy with F1-score of 91% and mAP value of 91.73%.[9]
JunYi Lim et al. utilized the M2Det model to analyze data obtained from various sources, including the Granada dataset comprising 3,000 images of different guns, the UCF crime dataset consisting of 7,247 images, and their own dataset consisting of 5,500 images. The researchers also examined different environmental conditions during their analysis. Two datasets were used in their study, with Model 1 relying solely on the Granada dataset and Model 2 combining both the Granada and their own dataset. The results demonstrated that Model 2 had superior accuracy due to its consideration of both low and high-resolution images. [10]

Ultralytics created the real-time object identification model known as YOLOv8. It is the eighth edition of YOLO, and compared to earlier versions, it is faster, more accurate, and more effective. The highest Mean Average Precision (mAP) in

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on October 26,2024 at 07:16:45 UTC from IEEE Xplore. Restrictions apply.

YOLO history of 53.9 has been marked with YOLOv8. The model was made in PyTorch, and it can be executed on both a CPU and a GPU. Many formats are supported by and quite effective with YOLOv8. While YOLOv5 is simpler to use, YOLOv8 is quicker and more accurate. Both YOLOv8 and YOLOv5 have advantages and disadvantages. YOLOv8 is preferable for applications that require real-time object detection, nevertheless, we have also looked at how YOLOv8 compares to other deep learning networks for real-time applications in this literature review. The YOLO algorithm has been upgraded with YOLOv8, which is quicker and more precise than previous iterations. It is a single-stage detector that executes real-time object detection using a deep neural network. Fig.4 shows the detailed architecture of YOLOv8.

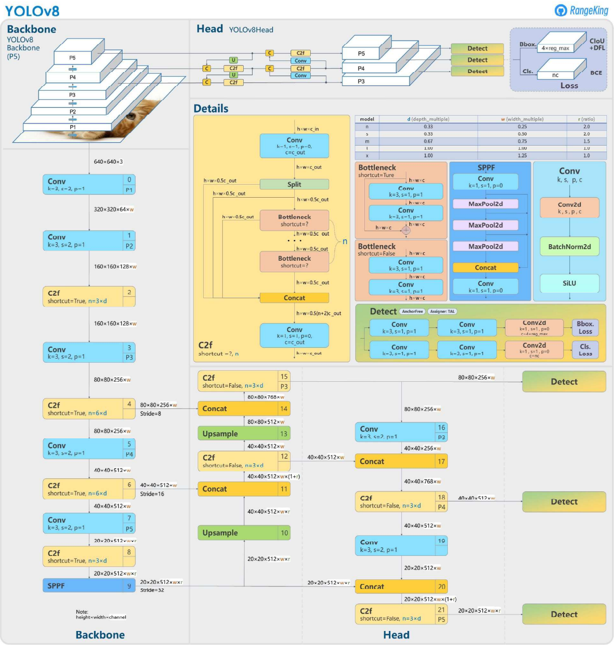

Fig.4. Architecture of YOLOv8 [4]

The effect of quantization on deep learning models has also been studied in the context of gun detection. Quantization is a technique that reduces the precision of weights and activations in deep learning models, resulting in smaller model sizes and faster inference times. D. Berardini et al. discovered that by employing Quantization Aware Training techniques, it is possible to mitigate the accuracy reduction that can occur during the optimization and quantization of models. [11]

III. Experimental Setup

A. Dataset used

The Roboflow computer vision development platform makes it simple to deploy custom datasets, use model training techniques, and gather data[12]. Users can access numerous publicly available datasets on it, and they can also contribute their own datasets. In this study, 2986 photos of weapons are collected from various sources like Google, YouTube, Instagram, etc., and the annotations are done using the Roboflow website. Roboflow provides easy integration of datasets into YOLO models. There are labels and photos in the dataset. A training set and a validation set are created from the dataset. 80% of the data is used for training purposes, and the remaining 20% is used for validation.

B. Working principle of YOLOv5

The YOLO (You Only Look Once) algorithm is widely used for object detection and operates by partitioning an input image into a grid of cells and forecasting bounding boxes and class probabilities for every cell. The general working principle of YOLO can be described in the following steps:

1. Input Image: The algorithm takes an input image and resizes it to a fixed size.
2. Grid Division: This Involves dividing the input image into a grid of cells, with each cell being assigned the task of detecting objects located in its corresponding area.
3. Anchor Boxes: YOLO uses anchor boxes, which are pre-defined shapes that are used to detect objects of different sizes and aspect ratios. Each cell predicts a fixed number of anchor boxes that are defined by the user.
4. Predictions: For each anchor box, YOLO predicts the class probability and the coordinates of the bounding box that encloses the object. The class probability represents the likelihood that the object belongs to a particular class, while the coordinates of the bounding box are represented as offsets from the top-left corner of the cell.
5. Non-Maximum Suppression: YOLO applies non-maximum suppression (NMS) to remove duplicate detections and select the most accurate bounding boxes. NMS works by comparing the overlap between different bounding boxes and discarding those with a low confidence score.
6. Output: The outcome is a roster of class probabilities and bounding boxes that correspond to all the identified objects in the input image.

C. Working Principle of YOLOv8

Although the goal of this study is to compare the performance of the two models, the working concept of YOLOv8 and YOLOv5 is relatively comparable. The usage of an anchor-free detection, a new and improved loss function, and improvements to internal convolutional blocks in version 8 is the major changes that have enabled this version to achieve higher mAP values. All the remaining steps are similar to that of the previous versions of YOLO.

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on October 26,2024 at 07:16:45 UTC from IEEE Xplore. Restrictions apply.

D. *Importance of quantization and its use in the project*

The process of quantization involves lowering the precision of the weights, biases, and activations to minimize their memory requirements. In essence, it entails converting a neural network, which usually represents parameters with 32-bit floats, to a smaller representation like 8-bit integers. A significant advantage of quantization is that it drastically reduces memory consumption. For example, transitioning from 32-bit to 8-bit would shrink the model size by a factor of 4. Fig.5 shows the working principle of Quantization in a simpler way.

Quantization
Floating point
Integer
27.1265
27
32 bit
8 bit

Fig.5. Quantization working Principle

Another advantage of quantization is the possibility of lowering network latency and enhancing power efficiency. By utilizing integer data types instead of floating-point, operations can be carried out more quickly within the network. These integer operations require fewer computations, which most CPU cores, including those found in microcontrollers, are capable of performing. Hence, power efficiency is increased overall as a result of reduced processing and memory access.[13]

Although quantization has its benefits, one drawback is that it may affect the accuracy of neural networks. This is because quantization alters how information is represented in the network. However, research has shown that the loss of accuracy due to quantization is often negligible compared to the gains in latency reduction, memory savings, and energy efficiency. The extent to which accuracy is affected depends on the level of precision loss, the architecture of the network, and the specific training and quantization techniques used.

E. *Metrics*

To calculate the accuracy, the metric used in the project is mAP. The value of mAP over the set of all objects (O) in a dataset can be expressed as shown in Equation (1)

$$mAP = \frac{1}{|O|}\sum_{c \in O} AP(c) \qquad (1)$$

Here AP is Average Precision, and for each object class c, one can calculate the average precision (AP) as shown in Equation (2)

$$AP(c) = \frac{TP(c)}{TP(c)+FP(c)} \qquad (2)$$

where TP(c) represents the number of true positive instances and FP(c) is the number of false positive instances for class c. For an arbitrary class c, value AP(c) = 1 would represent a perfect detection and AP(c) =0 the worst

IV. Results

Table 1 shows that there is an improvement in accuracy when you move from YOLOv5 to YOLOv8 .The first table figure show, YOLOv5 has obtained percentage accuracy around 89.1% whereas the percentage accuracy increases to 90.1 when moved to YOLOv8.

Table 1. Comparison of YOLOv8 with YOLOv5

Version	Precision	Recall	mAP50	mAP50-95	Inference Time
YOLOv5	92.4%	84.2%	89.1%	67.4%	8.7ms per image
YOLOv8	92.6%	81.7%	90.1%	72.7%	9.0ms per image

Weight quantization changes the 32-bit float value to an 8-bit integer value which reduces the complexity of the architecture in YOLOv8. As observed from Table 2, due to quantization, there is a slight difference in the accuracy percentage (slightly lower), but the inference time has gone down by approximately 15%, i.e., from 9ms to 7.6ms hence proving the point that quantization gives the results faster with acceptable accuracy. Also, when comparing the quantized YOLOv8 model to the unquantized YOLOv5 model, we observe that the inference time has gone down by approximately 12.5%, i.e., from 8.7ms to 7.6ms with just a slight decrease in accuracy.

Version	Precision	Recall	mAP50	mAP50-95	Inference Time
YOLOv8 (without quantization)	92.6%	81.7%	90.1%	72.7%	9.0ms per image
YOLOv8 (with quantization)	91.6%	81.4%	88.1%	70.2%	7.6ms per image

Table II . Comparison of inference time of YOLOv8 with and without quantization

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on October 26,2024 at 07:16:45 UTC from IEEE Xplore. Restrictions apply.

V. Conclusion

For the protection and security of the general populace, finding weapons is a crucial duty, and YOLO has demonstrated significant promise in taking on this task. The research discussed in this article shows that YOLO works in a variety of contexts, including photographs from social media and surveillance videos. YOLOv5 has some features that give great accuracy with some good results, but this was not sufficient. Later, YOLOv8 was considered as an alternative for greater accuracy, but this, too, had time inefficiencies. To simplify the architecture and complete the prediction of guns with high accuracy in the shortest amount of time possible, weight quantization was done. This helped to obtain an inference time of just 7.6ms with a mean Average Precision(mAP) value of 88.1%. Public safety is a major concern, so the algorithm must be well-designed and have highly accurate real-time predictions of the weapons.

References

[1] Jocher, G., Chaurasia, A., and Qiu, J. (2023). YOLO by ultralytics

[2] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, Yonghye Kwon ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (v7.0). Zenodo. https://doi.org/10.5281/zenodo.7347926

[3] “YOLOv8 docs,” https://docs.ultralytics.com, accessed March 31

[4] “Dive into YOLOv8: How does this state-of-the-art model work?,” https://openmmlab.medium.com/dive-into-yolov8-how-does-this-state-of-the-art-model-work-10f18f74bab1, accessed March 31, 2023

[5] YOLOv8 GitHub repository, https://github.com/ultralytics/ultralytics, accessed March 31, 2023

[6] R. M. Reddy Yanamala and M. Pullakandam, "An Efficient Configurable Hardware Accelerator Design for CNN on Low Memory 32-Bit Edge Device," in 2022 IEEE International Symposium on Smart Electronic Systems (iSES), pp. 112-117, 2022.

[7] R. M. Reddy Yanamala and M. Pullakandam, "A high-speed reusable quantized hardware accelerator design for CNN on constrained edge device," Design Automation for Embedded Systems, vol. 1, no. 25, 2023.

[8] Xu, Renjie, et al. "A forest fire detection system based on ensemble learning." *Forests* 12.2 (2021): 217.

[9] Bhatti, Muhammad & Khan, Muhammad Gufran & Aslam, Masood & Fiaz, Muhammad. (2021). Weapon Detection in Real-Time CCTV Videos using Deep Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3059170.

[10] J. Lim, M. I. Al Jobayer, V. M. Baskaran, J. M. Lim, K. Wong and J. See, "Gun Detection in Surveillance Videos using Deep Neural Networks," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Lanzhou, China, 2019, pp. 1998-2002, doi: 10.1109/APSIPAASC47483.2019.9023182.

[11] D. Berardini, A. Galdelli, A. Mancini and P. Zingaretti, "Benchmarking of Dual-Step Neural Networks for Detection of Dangerous Weapons on Edge Devices," *2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Taipei, Taiwan, 2022

[12] D. Berardini, A. Galdelli, A. Mancini and P. Zingaretti, "Benchmarking of Dual-Step Neural Networks for Detection of Dangerous Weapons on Edge Devices," *2022 18th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, Taipei, Taiwan, 2022

[13] Faraone, Julian, et al. "Syq: Learning symmetric quantization for efficient deep neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018.

Authorized licensed use limited to: VIT University- Chennai Campus. Downloaded on October 26,2024 at 07:16:45 UTC from IEEE Xplore. Restrictions apply.