



## Weapon Detection for Smart Surveillance System using YOLOV6

\*Jamilu Sulaiman<sup>1</sup>, Majeed Hussain Mohammed Abdul<sup>2</sup>

<sup>1</sup>Robotics & AI Research Lab, Department of Electrical/Electronic Engineering Technology, Kano State Polytechnic, Nigeria

<sup>2</sup>University of Grenoble Alpes, France

DOI: 10.5281/zenodo.7272124

Submission Date: 26<sup>th</sup> Oct. 2022 | Published Date: 31<sup>st</sup> Oct. 2022

\*Corresponding author: Jamilu Sulaiman

Robotics & AI Research Lab, Department of Electrical/Electronic Engineering Technology, Kano State Polytechnic, Nigeria

### Abstract

In recent years, many parts of the world recorded an increase in weapon violence, which led to a lot of innocent killings. The purpose of this work is to develop a computer vision-based automated smart surveillance system that detects visible weapons (a variety of knives, guns, and rifles) and sends alerts to the appropriate authorities for necessary action. Deep learning has provided a lot of solutions across the various sectors of human endeavor. In our proposed system, we used a single-stage target deep learning algorithm called YOLOv6 (You Only Look Once version 6) to detect visible weapons in real-time. Early detection of visible weapons increased response time by providing authorities and law enforcement with precious time to address the situation.

**Keywords:** Deep learning, Object detection, Machine learning, Artificial Intelligence, Neural Networks.

## INTRODUCTION

Weapon violence is one of the contemporary human rights issues across the globe; it threatens some of our fundamental human rights, such as the right to life, health care, education, e.t.c <sup>[15]</sup>.

According to Amnesty International, more than 500 people die every day from gun violence, 44% of all homicides globally involve gun violence, there were 1.4 million firearm-related deaths recorded globally between 2012 and 2016, an estimated 2000 people are injured by gunshots daily and at least 2 million people are living with firearm injuries around the globe <sup>[15]</sup>. According to 2018 crime statistics released by the FBI, knives were more than five times to be used for murder than rifles in the U.S.A. <sup>[17]</sup>. Also, studies show that most of the crimes such as robbery, kidnapping e.t.c. were carried out using weapons like knives, guns, and rifles. Our research was further motivated by many people. While designing the algorithms, we received interest and positive remarks from many national and local authorities, and a lot of companies that install CCTV monitoring systems.

Nowadays, a lot of security cameras are installed in various streets for surveillance purposes. Although human beings are good at recognizing the events happening in the street, a global view of multiple people and determining their situation can be done using smart systems. Deep learning algorithms are now doing an excellent job of recognizing, detecting, understanding, and performing a wide range of tasks in various industries. In the last decade, there has been an exponential rise in data on the internet and huge improvements in GPU hardware. To perform detection and recognition tasks better, this algorithm was trained on a big GPU.

In recent years, Artificial Intelligence algorithms <sup>[1, 2]</sup> in the domain of computer vision have provided a lot of algorithms in security systems <sup>[3, 4]</sup>. Narejo et al. <sup>[5]</sup> used the yolov3 model in their paper, which has lower accuracy than ours. We have used YOLO V6 <sup>[6]</sup>, which is more accurate, and we have annotated our own data with more labels of weapons. We also have an auto-triggering alerting system without human intervention. In this research work, we developed a smart surveillance system to detect visible weapons (a variety of knives, guns, and rifles). To accomplish this task, we have used a few computer vision techniques and algorithms. Recent work in the field of deep learning, especially with convolutional neural networks, has shown significant results in object detection and recognition. In recent years, there have been a lot of updates to YOLO versions. In this task to perform object detection and recognition, we

trained the classifier model of YOLO V6 (You Only Look Once Version 6) <sup>[6]</sup>. After the object detection and recognition, we perform post processing, where we collect the location GPS data for further use in alert systems. This program was developed to be installed directly on every security camera so that the camera not only just records but also sends an alert to appropriate authority whenever a visible weapon is detected.

Various challenges are present in object detection and recognition. A few of the major issues are: partial occlusion, lower resolution cameras, inaccurate detections, and less annotated training data. Keeping all of the challenges in this task in mind, we proposed a method to address the problem and perform the detection task as efficiently as possible.

## LITERATURE REVIEW

In deep learning, convolutional neural networks are widely known for object detection and recognition tasks. Convolutional neural networks <sup>[7]</sup> are one of the types of neural networks that have been proven to have great performance in image recognition and classification. ConvNets have been useful in the detection and recognition of objects. To solve computer vision tasks, various convolutional networks (ConvNets) architectures are available, including LeNet, VGG16, ResNets, and AlexNet. All the architectures consist of four main operations: convolution, non-linearity activation functions, pooling, and classification (fully connected layer). ConvNets derive their name from the convolution operator. The main purpose of convolution in the case of a ConvNet is to extract feature maps from the given input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. Every image is forwarded multiple layers by performing convolutions to learn the feature maps. The size of the feature map is controlled by the number of filters, padding, and strides. After every convolutional layer, there is an activation function, which is used to introduce non-linearity in ConvNet since most of the real-world data on which ConvNet learns is non-linear. The most widely used activation function is ReLU. It stands for Rectified Linear Unit. ReLU is an element-wise operation applied per pixel and replaces all negative pixel values in the feature map by zero. Pooling, also called subsampling or down-sampling, is used to reduce the dimensionality of each feature map but it retains the most important information. Pooling is of various types, such as maximum, average, and global average pooling. In the case of Max Pooling, we define a spatial neighborhood, for example, a  $2 \times 2$  window, and take the largest element from the rectified feature map within that window and discard all the other minimum values. In average pooling, the average of the elements in the window would be taken. In practice, Max Pooling has been shown to work better for learning the discriminative features for image recognition. Finally, the FC layer, as the name implies, connects every neuron in the previous layer to every neuron in the next layer. It uses a softmax function in the output layer. The outputs from all the previous convolutional layers and pooling layers (learnt deep features) are forwarded to the fully connected layer to classify the given input image. Here, a soft-max layer at the end of the fully connected layer gives the probability scores of the input image identity.

## Object Detection algorithms

In this section, we will review various object detection algorithms and reasons for choosing a specific algorithm. For object detection, various methods are available before Deep Learning. Object detection involves several step processes, starting from using low-level features like edges, scale invariant features, some of the techniques like SIFT, HOG, DPM, and so on. These output images were then compared with existing object templates, usually at multi scale levels, to detect and localize objects present in the image. Presently, due to the huge advancement in deep learning, various methods are available for object detection tasks. Some of the popular algorithms are YOLO (You only look at once algorithm) <sup>[6]</sup>, SSD (Single Shot Detector) <sup>[8]</sup>, Retina Net <sup>[9]</sup>, Faster R-CNN (Faster Region-based Convolutional Neural Networks) <sup>[10]</sup>, R-FCN (Region-based Fully Convolutional Networks) <sup>[11]</sup>. Choosing the right object detector is an essential step for this research since inaccurate object detection in the images may lead to failure in that object re-identification in further frame. The above-mentioned methods could be categorized into two variants, namely: the one-step object detection model and the two-step object detection model. Models in the R-CNN family are all region-based. The detection happens in two stages. (1) First, the two-stage model proposes a set of regions of interest by select search or regional proposal network. The proposed regions are sparse as the potential bounding box candidates can be infinite. (2) Then a classifier only processes the region's candidates. On the other hand, the one-stage model approach skips the region proposal stage and runs detection directly over a dense sampling of possible locations. This is how a one-staged object detection algorithm works, which is faster and simpler, but might potentially drag down the performance a bit. YOLO, SSD, and Retina Net are good examples of one-stage detectors. R-CNN families such as R-CNN, Fast R- CNN, Faster R-CNN, and R-FCN are also good examples of two-staged detectors.

YOLO (You Only Look Once), Introduced first in 2016 by Joseph Redmon et al. via a paper titled, "You Only Look Once: Unified, Real-Time Object Detection,"<sup>[18]</sup> it is considered a breakthrough in this field. Over the years, this model has undergone several iterations and advancements. Version 2 was released in 2017 (YOLO9000: Better, Faster, Stronger) <sup>[19]</sup>, followed by YOLOv3 (YOLOv3: An Incremental Improvement) in 2018<sup>[20]</sup>, YOLOv4 (YOLOv4: Optimal Speed and Accuracy of Object Detection) in April 2020<sup>[21]</sup>, and YOLOv5 in May 2020. YOLOv6 was recently introduced by Chinese company Meituan. It is not part of the official YOLO series but was named so since the authors of

this architecture were heavily inspired by the original one-stage YOLO [16]. The YOLO algorithm [6] is a convolutional neural network for object detection in real-time scenarios. The algorithm applies a single neural network to the full image and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. The YOLO algorithm is popular because it achieves nearly state-of-the-art accuracy in real-time. This algorithm only looks once at the image in the sense that it requires only one forward propagation pass through the neural net to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes.

Compared to all the models, we chose the YOLOv6 model since we opted for speed and accuracy. In surveillance, most of the videos are continuous day and night. In order to process that huge amount of data, we need a model that is fast in processing and also gives the best results. Therefore, we chose the YOLOv6 model, which outperforms all models in terms of speed and accuracy.

## MATERIALS AND METHOD

In this section, we will discuss the materials and methodology. For this work, we need a simple RGB IP camera. We have used machine learning libraries such as tensor flow for training the model and annotation tool namely label studio for data annotation and finally we have used amazon cloud services for alerting systems and database.

### YOLOv6 Architecture

YOLOv6 was inspired by the original YOLO architecture. Though it provides outstanding results, it's important to note that MT-YOLOv6 is not part of the official YOLO series. YOLOv6 is a single-stage object detection framework dedicated to industrial applications, with a hardware-friendly, efficient design and high performance. It outperforms YOLOv5 in detection accuracy and inference speed (Fig. 1), making it the best OS version of the YOLO architecture for production applications.

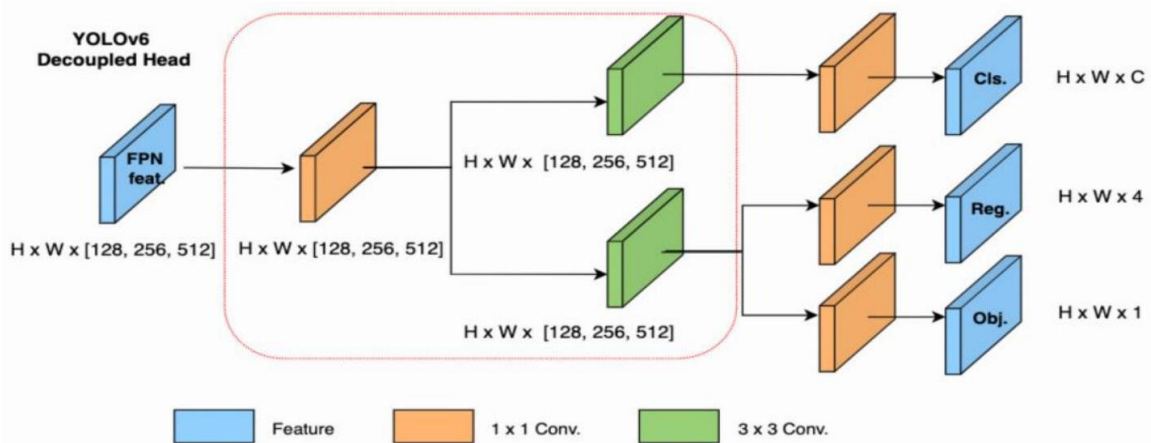


Fig. 1: Yolov6 Architecture [6]



Fig. 2: Comparison table [16]

YOLOv6-s achieves 43.1mAP on the COCO val2017 dataset with 520 FPS on T4 using TensorRT FP16 for bs32 inference (Fig. 2). As shown in Fig. 2, YOLOv6-s (red) outperforms all previous versions of YOLOv5 in terms of mean average precision (mAP), with approximately 2x faster inference time. We can also see a huge performance gap between the YOLO-based

Model	Size (pixels)	mAP <sup>val</sup> <sub>0.5:0.95</sub>	Speed <sup>V100</sup> <sub>fp16 bs32</sub> (ms)	Speed <sup>V100</sup> <sub>fp32 bs32</sub> (ms)	Speed <sup>T4</sup> <sub>trt fp16 b1</sub> (fps)	Speed <sup>T4</sup> <sub>trt fp16 bs32</sub> (fps)	Params (M)	Flops (G)
YOLOv6-n	416 640	30.8 35.0	0.3 0.5	0.4 0.7	1100 788	2716 1242	4.3 4.3	4.7 11.1
YOLOv6-tiny	640	41.3	0.9	1.5	425	602	15.0	36.7
YOLOv6-s	640	43.1	1.0	1.7	373	520	17.2	44.2




Fig. 3: MT-YOLOv6-Benchmark<sup>[16]</sup>

As we see from the benchmarks (Fig. 3), YOLOv6 in terms of speed and accuracy is higher than the rest of the models. Given all of the benefits of Yolov6, we have decided to use it for our proposed method

**Proposed Pipeline**

In order to perform this task, we have proposed a pipeline (Fig. 4).

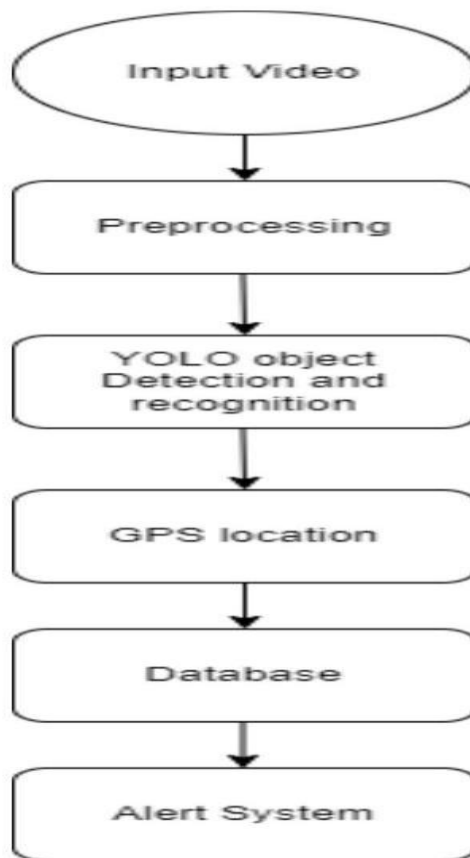


Fig. 4: Proposed Pipeline

Before we get into methodology, we will discuss a few building blocks concepts regarding deep learning.

**Preprocessing** - Before sending the video frames to the YOLOv6 object detection model, we perform a few preprocessing steps for better results. In our task, we have smoothed the frames using a Gaussian filter and further resized the images according to the input shape of the YOLOv6 object detection model.

**YOLOv6 Object detection** - In this step, each image is forwarded to the YOLO detector. The output of the model is a bounding box with labels. These outputs are forwarded to database along with GPS data provided by CCTV camera.

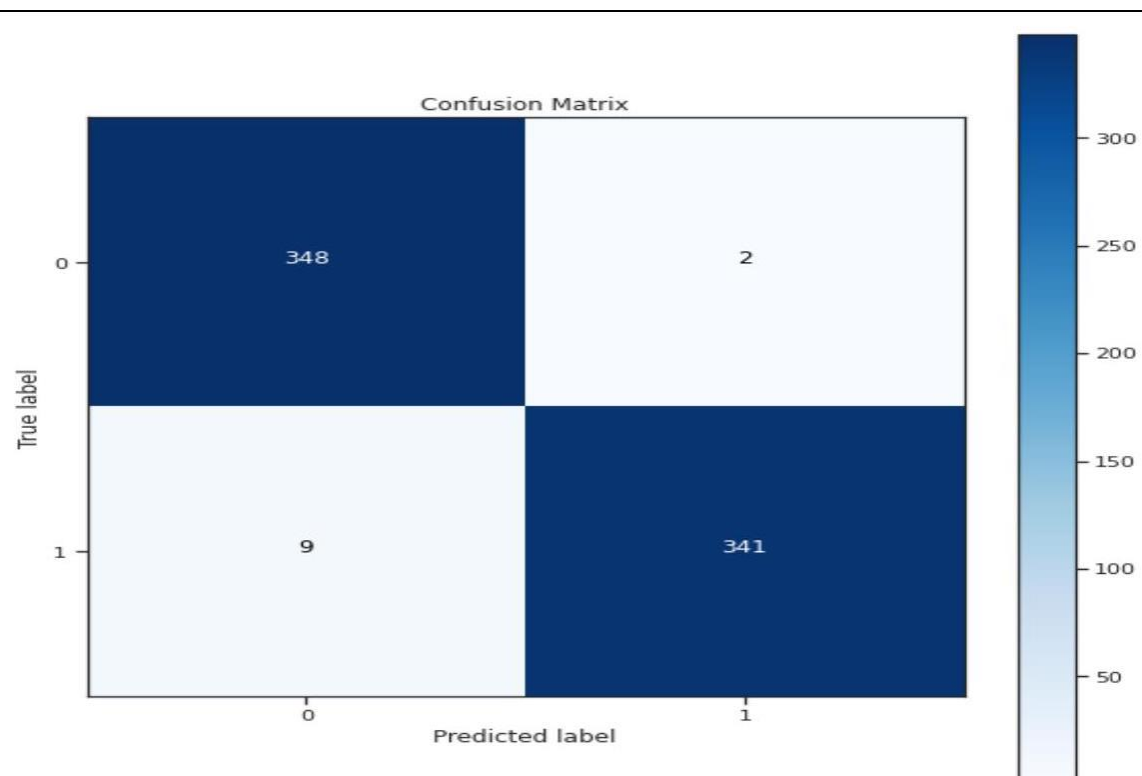
**GPS Location** - In this work, the GPS location of the place where the camera mounted is part of the alert content sent to the appropriate nearby authorities.

**Database** - All the activities of day-to-day life and processing of object detection and recognition are stored in a database. The database also has the GPS data for all the cameras. Whenever any kind of weapon is detected from the processed frames, the system will trigger a notification to the appropriate authorities with information of weapon detected image frame along with GPS location.

**Alert System** - For the alert system, we used email notification to notify the appropriate authorities whenever a weapon is detected. The contents of the email include the GPS location of the camera and the frame of where the weapon was detected.

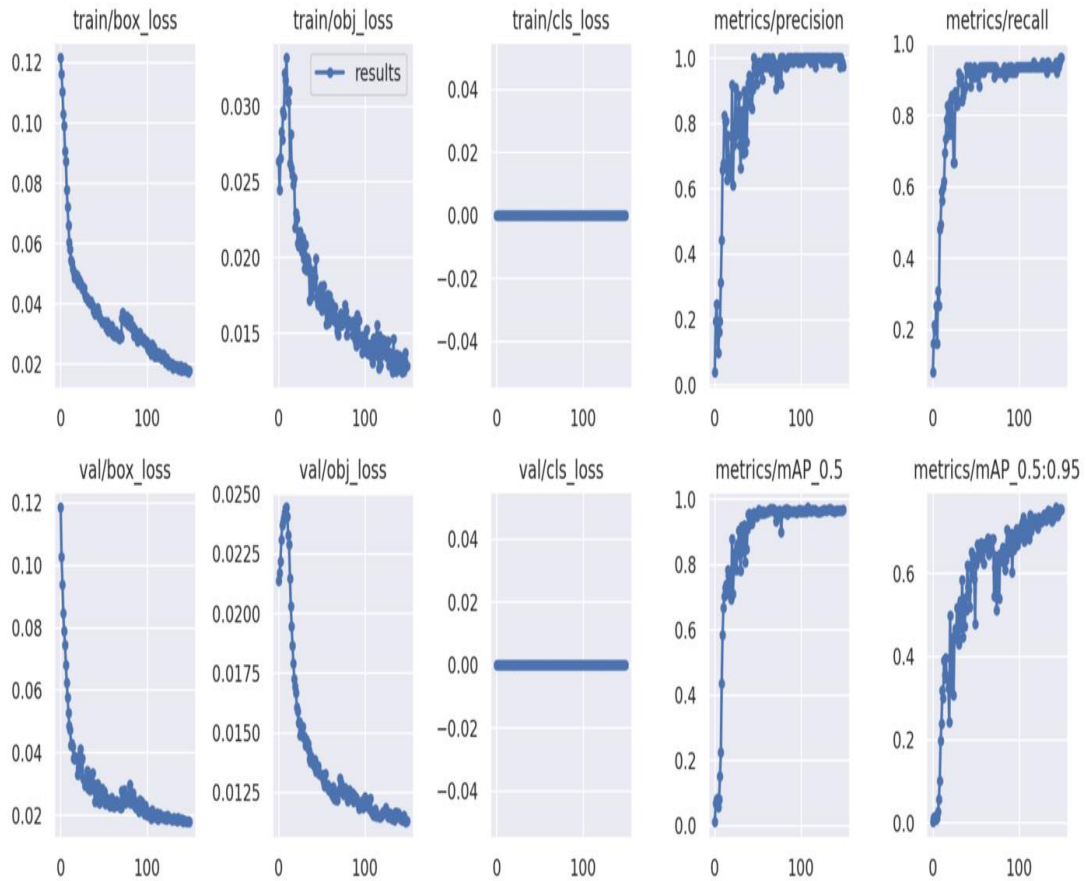
## RESULTS

In this section, we will discuss the results. The model is trained with two labels: "gun" and "knife". Fig. 5 shows the confusion matrix of our trained model. A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically supervised learning. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class or vice versa.



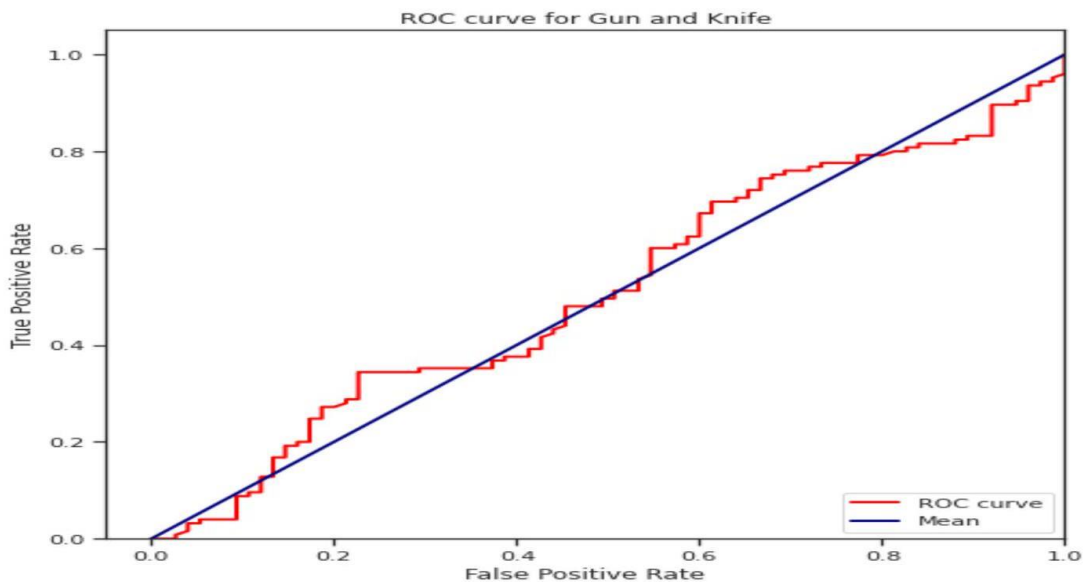
**Fig. 5:** Confusion matrix

Fig. 2 is the confusion matrix of our trained model. A confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically supervised learning. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class or vice versa.



**Fig. 6:** Training curves

**Fig. 6:** shows the training curves for training and validation loss along with metrics, precision, and mean Average Precision (mAP). ROC curve for Gun and Knife



**Fig. 7:** Receiver Operation Characteristic (ROC) Curve

Fig. 7 shows the receiver operating characteristic (ROC) curve, which is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

1. The percentage of true positives
2. The rate of false positives

The True Positive Rate (TPR) is a synonym for recall and is therefore defined as follows:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Where, TP stands for True Positive and FN stands for False Negatives

The False Positive Rate (FPR) is defined as follows:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

Fig. 8 and Fig. 9 are the outputs images of the detection algorithm.



**Fig. 8:** Knife detection results



**Fig. 9:** Gun detection results

## DISCUSSION

In this work although the results were satisfactory yet the following updates could be done in the future work:

- Currently model is trained with two labels of dataset namely Guns and Knife. In future we could add many labels with more datasets to address various kinds of weapons. - Training accuracy could be increased with adding more data, since data is scarce; we could generate a synthetic data using other machine learning algorithms like Generative adversarial networks<sup>[14]</sup> and train model on those generated data.
- Speedup the training time and inference time using parallel processing with multi-gpu system.

## CONCLUSION

Overall, this work can significantly help in addressing the security challenges by detecting the visible weapons and sending alerts to the appropriate authorities in real time.

## ACKNOWLEDGEMENT

In this paper, we acknowledge the various datasets that were used and manually labelled. Some of the datasets we have used are from Kaggle, such as the Real-Life Violence Situations Dataset <sup>[12]</sup>. It has 1000 videos, and the Smart-City CCTV Violence Detection Dataset <sup>[13]</sup>. All of these dataset's videos are extracted into frames and annotated manually to train our model.

## REFERENCE

1. Fujita, Hamido. (2022). Advances and Trends in Artificial Intelligence. Theory and Practices in Artificial Intelligence. 10.1007/978-3-031-08530-7.
2. Fujita, Hamido. (2022). Intelligence of Artificial Intelligence: Philosophy. 10.31219/osf.io/d9fe3.
3. Kodali, Ravi & Jain, Vishal & Bose, Suvadeep & Boppana, Lakshmi. (2016). IoT based smart security and home automation system. 1286-1289. 10.1109/CCAA.2016.7813916.
4. Saifuzzaman, Mohd & Hossain, Ashraf & Nessa, Nazmun & Nur, Fernaz. (2017). Smart Security for an Organization based on IoT. International Journal of Computer Applications. 165. 33-38. 10.5120/ijca2017913982.
5. Narejo, M. (2021). Weapon Detection Using YOLO V3 for Smart Surveillance System. Mathematical Problems in Engineering, 2021, 9975700.
6. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., Li, Y., Zhang, B., Liang, Y., Zhou, L., Xu, X., Chu, X., Wei, X., & Wei, X.. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications.
7. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097– 1105. Curran Associates, Inc., 2012.
8. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng- Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector, 2015. cite arxiv:1512.02325Comment: ECCV 2016
9. T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(2):318–327, 2020.
10. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6):1137–1149, 2017.
11. Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, NIPS, pages 379–387, 2016.
12. <https://www.kaggle.com/datasets/mohamedmustafa/real-life-violence-situations-dataset>
13. <https://www.kaggle.com/datasets/toluwaniamemu/smartcity-cctv-violence-detection-dataset-scvd>
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672–2680).
15. <https://www.amnesty.org/en/what-we-do/arms-control/gun-violence/>
16. <https://dagshub.com/blog/yolov6/>
17. <https://americanmilitarynews.com/2019/10/fbi-stats-show-5-times-more-murders-by-knives-than-rifles-in-2018/#:~:text=The%20Federal%20Bureau%20of%20Investigation%20released%20crime%20statistics,studying%20the%20types%20of%20weapons%20used%20in%20murders.>
18. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
19. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
20. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
21. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

### CITE AS

Jamilu Sulaiman, & Majeed H. M. Abdul. (2022). Weapon Detection for Smart Surveillance System using YOLOV6. Global Journal of Research in Humanities & Cultural Studies, 2(5), 199–206.  
<https://doi.org/10.5281/zenodo.7272124>