Table of Contents

Abstract

This project focuses on the crucial task of firearm detection using the YOLOv7 architecture and a synthetic dataset. In recent years, the alarming rise in gun-related incidents and fatalities has prompted an urgent need for innovative solutions. Our primary objective is to develop a robust system capable of accurately identifying firearms in images. By doing so, we aspire to significantly enhance security, support law enforcement, and reduce firearm-related incidents.

We leverage the potent capabilities of the YOLOv7 architecture to train our detection model. The cornerstone of our approach is the creation of a meticulously curated synthetic dataset. This dataset encompasses a diverse array of firearm types, positions, and environmental conditions. Essentially, it offers our model a versatile training regimen, allowing it to excel in a wide range of real-world scenarios. This adaptability is pivotal in ensuring the model's effectiveness and reliability when deployed in various security and law enforcement contexts.

The significance of this project is deeply rooted in its potential to make a tangible difference in society. By providing a dependable tool for early and precise weapon detection, we aim to make significant contributions to public safety and security. Our work aligns with the broader mission of preventing firearm-related incidents, mitigating potential harm, and ultimately saving lives in an increasingly complex and challenging world.

Through innovation and rigorous training using synthetic data, this project represents a progressive step in harnessing cutting-edge technology to address a critical societal concern. Our ultimate goal is to create a safer and more secure future for all, where the benefits of advanced technology are harnessed to protect and preserve human lives.

Introduction

In our increasingly uncertain world, public safety has become a major concern. Tragic incidents involving weapons in public places have witnessed a concerning rise. To tackle this issue, we're focused on developing a system that can keep an ever-watchful eye on public areas.

Our aim is pretty straightforward, monitor public spaces more effectively. Our project uses the latest tools for object detection to identify weapons in real-time. This means it can quickly spot concealed or potentially dangerous items, offering a crucial layer of security. We have trained our model in very challenging situations with complex images where the weapon is barely visible along with a synthetic dataset which makes the model to be able to detect even in complex environments.

Our system doesn't replace human vigilance; instead, it enhances it. It acts as an extra set of eyes, working tirelessly alongside security personnel and law enforcement. The ultimate goal is to save lives by detecting threats early, reducing response times, and minimizing the chances of harm in public places.

Our focus is not just on technology but on people's safety. We want to ensure that public spaces are safer for everyone and that security personnel have the support they need to protect our communities. By combining the simplicity of our system with the sophistication of YOLOv7, powered by neural networks and brought to life through Python programming and neural networks, we're striving to make public spaces a little safer for us all.

Literature Survey:

1) Robert-Manuel Bota-Ioana, Anca Gavrilas (Military Management, Romania)

Title: Application for threat surveillance using Python and Yolo-V7

In that study, the critical need for public safety surveillance is highlighted in terms of AI's crucial role in mimicking humanlike intelligence, as there is a growing availability of military-grade equipment and possible threats to densely populated areas and commercial space. The research concerns the use of YOLOv7, an efficient Open Source Object Detection System and a Machine Learning tool to assist in detecting objects, most notably firearms, as quickly as possible.
A simple downloader of images to be used for data acquisition and preprocessing, as well as user-friendly Graphical User Interfaces that allow easy interaction are employed in the study. The results show that YOLOV7 is capable of detecting objects under a variety of challenging conditions, e.g. blurted images, showing its potential for security applications. This study concludes that YOLOV7 is proving to be effective and promising in the rapid detection of objects, thereby demonstrating its significant capacity for enhancing the safety of citizens, particularly in security systems.

2) R J Anandhi (New Horizon College of Engineering Bengaluru, India) **Title:** Edge Computing-Based Crime Scene Object Detection from Surveillance Video Using Deep Learning Algorithms

The study aims at increasing the surveillance system for crime scenes by using automatic identification of real-time objects. Firearms are a common feature of crime and require prompt identification to be noticed by security personnel. A fully synchronised NeuralCNNs network has been integrated into YOLO.V7. and Edge Computing in the proposed approach, creating a powerful hybrid system. Automated monitoring is needed due to shortcomings and limitations in the traditional supervision of humans. The system can achieve 92% accuracy, 95% precision, 88% recall and an F1 rating of 88.5% with the use of Deep Learning algorithms like YOLOV7. The harmonised system shows efficiency as regards the rapid identification of potential threats, which in turn has a significant impact on public safety.
Regression linear-based object detection with YOLO-V7 and classification using CNNs are the two primary steps of the method. Different YOLO iterations (V2, V3, V4) and classification models (RCNN, Fast-RCNN, Faster-RCNN, CNN) are trained using a dataset of crime item photos. YOLO-V7 is the best option for real-time detection because comparisons show that it has greater precision and accuracy. The system's dependability is increased when YOLO-V7 and CNN-based verification are used together, reducing false positives. The study confirmed the effectiveness of a combination approach as proactive surveillance and demonstrated promising results in increasing the safety and detection of suspects carrying weapons.

3) Moahaimen Talib Abdullah, Jamila Harbi Alameri (Mustansiriyah University, Baghdad, Iraq)
<u>Title</u>: A Multi-Weapon Detection Using Synthetic Dataset and Yolov5

Weapon detection must be accurate and quick because of the rising threat of international assaults, both terrorist and criminal-related. The idea of a system utilizing deep learning and Convolutional Neural Networks (CNNs) is motivated by the difficulties currently encountered in classifying various weapon kinds. Weapon detection has been studied in the past, with a focus on knives and guns, employing techniques like Faster R-CNN and CNN-based recognition. This study sets itself apart by using YOLOv5 to identify various weapon types, including the AKM, M4, and RPG. Due to its high accuracy, quick processing, and lightweight characteristics, YOLOv5, known for real-time object identification, is helpful. The YOLOv5x6 model is trained using image preprocessing and augmentation together with a well-curated dataset. The model's performance in the experimental findings is exceptional, with 99% precision, 93% recall, and a 95% mean average precision (mAP). Notably, the model is very promising for real-world security applications because of its diversity in recognizing different weapon kinds and its ability for quick and precise identification. Future work could increase the sorts of weapons that the model can detect and broaden its usefulness to drone-based monitoring, improving security operations even more. In the end, the proposed approach has the potential to considerably advance autonomous security systems and provide a safer environment.

4) Anjali Goenka, K. Sitara (National Institute of Technology, Tiruchirappalli) **<u>Title</u>**: Weapon Detection from Surveillance Images Using Deep Learning

The paper begins by discussing the frightening statistics of gun violence around the world, highlighting the urgent need for efficient technologies to combat this problem. It emphasizes how deep learning has the potential to improve surveillance systems, particularly in the areas of object recognition and image segmentation. The motivation and issue statement highlight the drawbacks of manual gun detection and outline the difficulties that make automatic firearm identification through intelligent surveillance necessary. The fundamental component of the suggested remedy is Mask RCNN, a pre-trained object instance segmentation model that focuses on handgun identification using CNN-based learning. The approach details dataset features, experimental findings, evaluation measures, and preprocessing implications. It also describes key preprocessing stages and the architecture of Mask RCNN. The presentation ends with a summary of accomplishments, emphasizing improved real-time processing and potential future developments like the YOLOv4 application and dataset.

5) Naresh Yeddula, B. Eswara Reddy(JNTUA College of Engineering, Ananthapuramu) **<u>Title</u>**: Effective Deep Learning Technique for Weapon Detection in CCTV Footage

The report that is being presented emphasizes the critical need for cutting-edge technology to address the urgent worldwide problem of gun violence. Despite the prevalence of surveillance cameras, important data cannot always be captured via conventional means. A potential answer is deep learning, particularly in object detection and image segmentation. The necessity for quick, effective, and automated handgun detection is driven by the drawbacks of manual gun detection. The suggested method uses the Mask Region-based Convolutional Neural Network (Mask RCNN) to focus on automatic pistol identification utilizing security cameras. The methodology includes important preprocessing steps and the Mask RCNN architecture. The proposed system successfully achieved automatic pistol recognition in the study's experiments, laying the path for further developments such as YOLOv4 integration and dataset expansion to diverse gun categories.

6) Deepak R. Hawale, Pavin S. Game (Pune Institute of Computer Technology (PICT), Pune)
Title: Real-time Weapon Detection Using Drone

The paper focuses on integrating object detection technology, in particular those that aim at the identification of weapons, into Unmanned Aircraft Systems, which are one of the most common Internet of Things devices deployed for different applications. The system uses OpenCV and YOLOv3 to scan real-time objects, using the Raspberry Pi 4's camera module that is mounted on a DIY drone. Live video feed is transmitted to a computer over the internet, where object detection occurs, and alerts are generated upon weapon identification. The system has shown that it can detect a wide range of weapons, promising to be used for security and law enforcement purposes. In the coming years, further efforts will aim at enhancing its connectivity and coverage of more extensive security applications.

System Methodology

The system is already been trained with a model using Pytorch, Yolo v7, git, and CUDA and the system is ready to detect the weapon.
As the system starts, it will ask you to upload the image. The image needs to be uploaded, then the system analyzes the image using the technologies it has been trained with and then it will check the image and see if it contains a weapon.
If the system detects a weapon the system will highlight the weapon.
If there is no weapon in the image then it just returns the original photo.

The activity diagram for the system

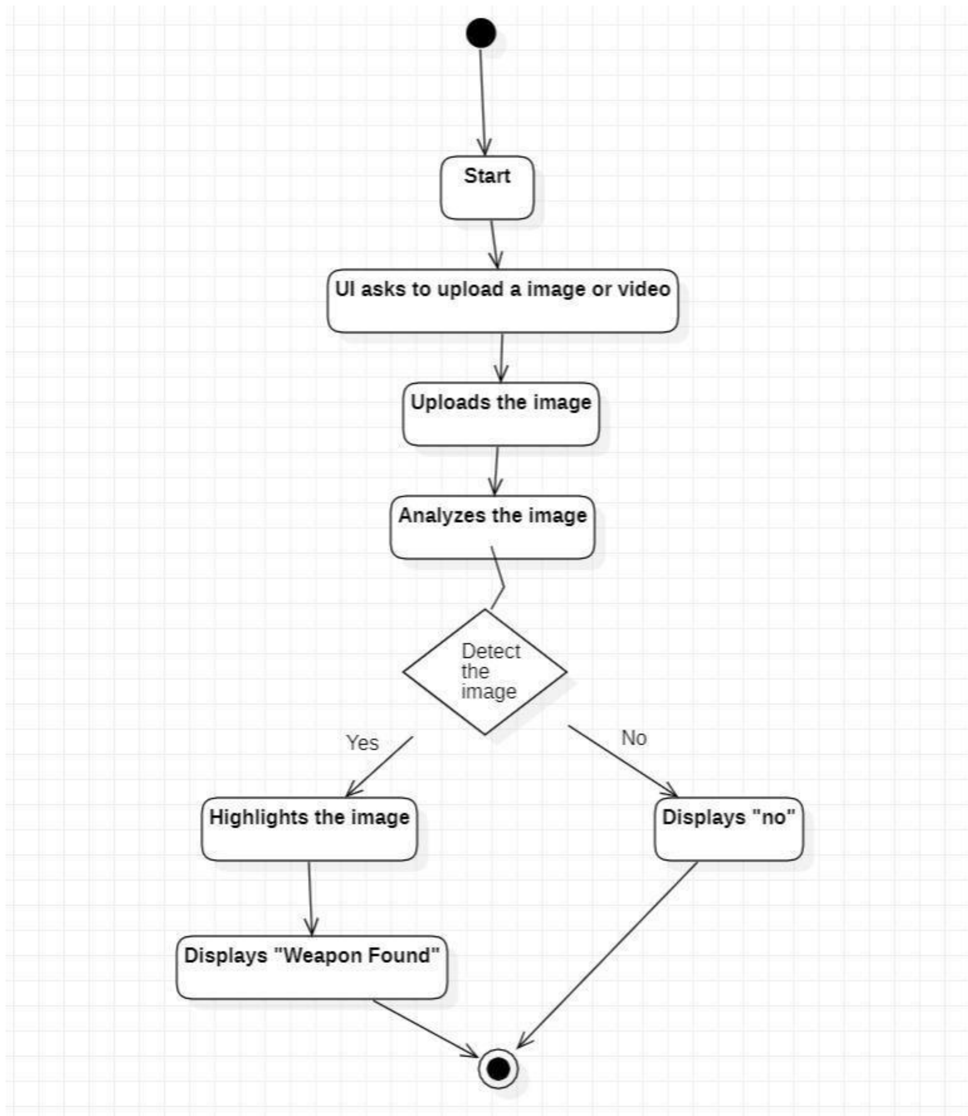

Figure 1

The design for the system

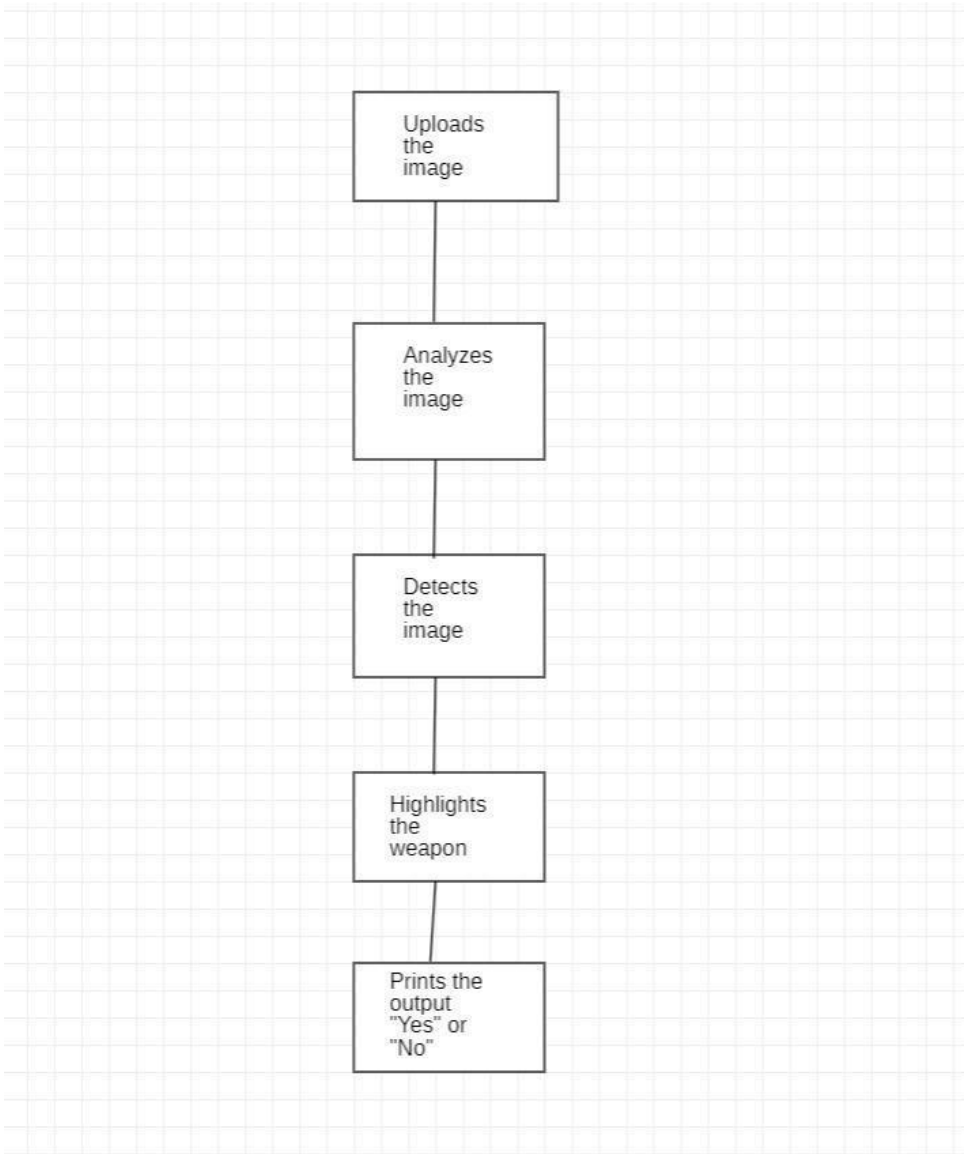

Figure 1

Overview of Technologies

YOLO v7:

YOLO architecture is FCNN(Fully Connected Neural Network) based. However, Transformerbased versions have also recently been added to the YOLO family.

The YOLO framework has three main components:

- Backbone
- Head
- Neck

The Backbone mainly extracts essential features of an image and feeds them to the Head through Neck. The Neck collects feature maps extracted by the Backbone and creates feature pyramids. Finally, the head consists of output layers that have final detections.

There are many versions in the YOLO family but the YOLO v7 had improved speed and accuracy by introducing several architectural reforms. Similar to Scaled YOLOv4, YOLOv7 backbones do not use ImageNet pre-trained backbones. Rather, the models are trained using the COCO dataset entirely. The similarity can be expected because YOLOv7 is written by the same authors as Scaled YOLOv4, which is an extension of YOLOv4.

Architecture:

The architecture is derived from YOLOv4, Scaled YOLOv4, and YOLO-R. Using these models as a base, further experiments were carried out to develop new and improved YOLOv7.

E-ELAN (Extended Efficient Layer Aggregation Network)
The E-ELAN is the computational block in the YOLOv7 backbone. It takes inspiration from previous research on network efficiency. It has been designed by analyzing the following factors that impact speed and accuracy.

- Memory access cost
- I/O channel ratio
- Element wise operation
- Activations
- Gradient path

In simple terms, E-ELAN architecture enables the framework to learn better. It is based on the ELAN computational block. The ELAN paper has not been published yet when writing this post. We will update the post by adding ELAN details when available.

PyTorch:

PyTorch is an open-source machine learning library primarily used for deep learning tasks. It provides a flexible framework for developing and training neural networks. PyTorch is known for its dynamic computational graph, which allows for more intuitive and imperatively defined models.

Key features of PyTorch include:

<u>Dynamic Computational Graph</u>: PyTorch utilizes a dynamic computation graph, allowing for changes and modifications to the network architecture during runtime. This is in contrast to static computation graphs used by some other frameworks.

<u>Tensor Computation</u>: At its core, PyTorch provides multi-dimensional arrays known as tensors. These tensors, similar to NumPy arrays, facilitate efficient numerical computations and form the foundation of building blocks for neural networks.

<u>Automatic Differentiation</u>: PyTorch supports automatic differentiation, a crucial feature for training neural networks through techniques like backpropagation. This simplifies the calculation of gradients and enables efficient optimization algorithms.

<u>Module System</u>: PyTorch offers a modular approach to building models using pre-built layers, activation functions, loss functions, and optimization algorithms. This modularity promotes code reusability and simplifies model construction.

<u>Eager Execution</u>: PyTorch follows an eager execution paradigm, enabling developers to evaluate operations immediately as they are called, aiding in debugging and experimentation.

<u>Libraries and Utilities</u>: PyTorch provides several utility functions and libraries for tasks such as data processing, image and video processing, natural language processing, and reinforcement learning, making it a comprehensive tool for various machine learning tasks.

<u>Scalability and GPU Support</u>: PyTorch supports GPU acceleration, allowing for faster training of deep learning models, making it suitable for handling large datasets and complex networks.

<u>Community and Support</u>: PyTorch has a strong and active community, contributing to its growth and continuous improvement. The community provides extensive documentation, tutorials, and examples, making it accessible for both beginners and experienced practitioners.

<u>Git</u>:

Git is a distributed version control system designed to handle projects of all sizes with speed and efficiency. It allows multiple users to collaborate on a project, tracking changes made to the source code and providing mechanisms for merging and managing these changes seamlessly.

<u>Key Features and Concepts</u>:

Repository (Repo):
A repository is the core element in Git, representing the project and containing all files, history, and configurations.

Commits:
Commits are snapshots of the repository at a specific point in time. They record changes made to the codebase, providing a detailed history.
Branches:
Branches are divergent timelines of the repository, allowing parallel development. Developers can create, switch, merge, and delete branches.

Merging:
Merging combines changes from one branch into another, typically used to integrate completed features back into the main project.

Pull Requests:
Pull Requests are proposals to merge changes from one branch into another. They facilitate code review and discussion before merging.

Remote Repositories:
Remote repositories are copies of the project hosted on servers, allowing collaboration among multiple contributors.

Clone:
Cloning creates a local copy of a repository, enabling developers to work on the project without affecting the original.

Push and Pull:
Push uploads local changes to a remote repository, while pull fetches and merges changes from a remote repository to the local copy.

CUDA:

CUDA is a parallel computing platform developed by NVIDIA, primarily for accelerating computations using NVIDIA GPUs. It provides an interface for developers to harness the immense parallel processing power of modern GPUs for a wide range of applications beyond traditional graphics rendering.

Key Features and Concepts:

Parallel Computing with GPUs:
CUDA enables parallel processing by utilizing the thousands of cores available in modern GPUs, significantly accelerating computations compared to traditional CPU-based processing.

CUDA Cores:
CUDA cores are individual processing units within a GPU. They handle tasks in parallel, making it possible to perform a multitude of computations simultaneously.

Kernel Functions:
In CUDA, developers write kernel functions that run in parallel on the GPU. These functions are specifically designed to exploit the GPU's parallel processing capabilities.

Grids, Blocks, and Threads:
CUDA organizes parallel execution into a hierarchy of grids, blocks, and threads. A grid contains blocks, and each block contains threads. This structure allows for efficient management of parallel tasks.

Memory Management:
CUDA provides mechanisms to efficiently manage memory, including global memory, shared memory, constant memory, and registers, optimizing data movement between CPU and GPU.

CUDA Toolkit and SDK:
The CUDA Toolkit includes the necessary compilers, libraries, and tools to develop CUDA applications. The CUDA Software Development Kit (SDK) offers examples and resources to help developers get started.

Implementation

We have imported the official YOLO v7 from the GitHub repository.
We have unzipped the dataset. Then we trained the model and tested it

Code:

```
!git clone https://github.com/WongKinYiu/yolov7
%cd yolov7

!unzip -q ../dataset.zip -d ../dataset

#training

!python '/content/yolov7/train.py' --workers 8 --device 0 --batch-size 8 --data '/content/dataset/coco.yaml' --img 416 416 --cfg '/content/yolov7/cfg/training/yolov7.yaml' -weights yolov7.pt --name yolov7 --hyp '/content/yolov7/data/hyp.scratch.p5.yaml'

#testing

!python '/content/yolov7/test.py' --weights '/content/yolov7/runs/train/yolov72/weights/best.pt' -data '/content/dataset/coco.yaml' --img-size 416 --conf-thres 0.001 --iou-thres 0.65 --task val -name yolov7
```

When training the data set this is the analysis that is generated

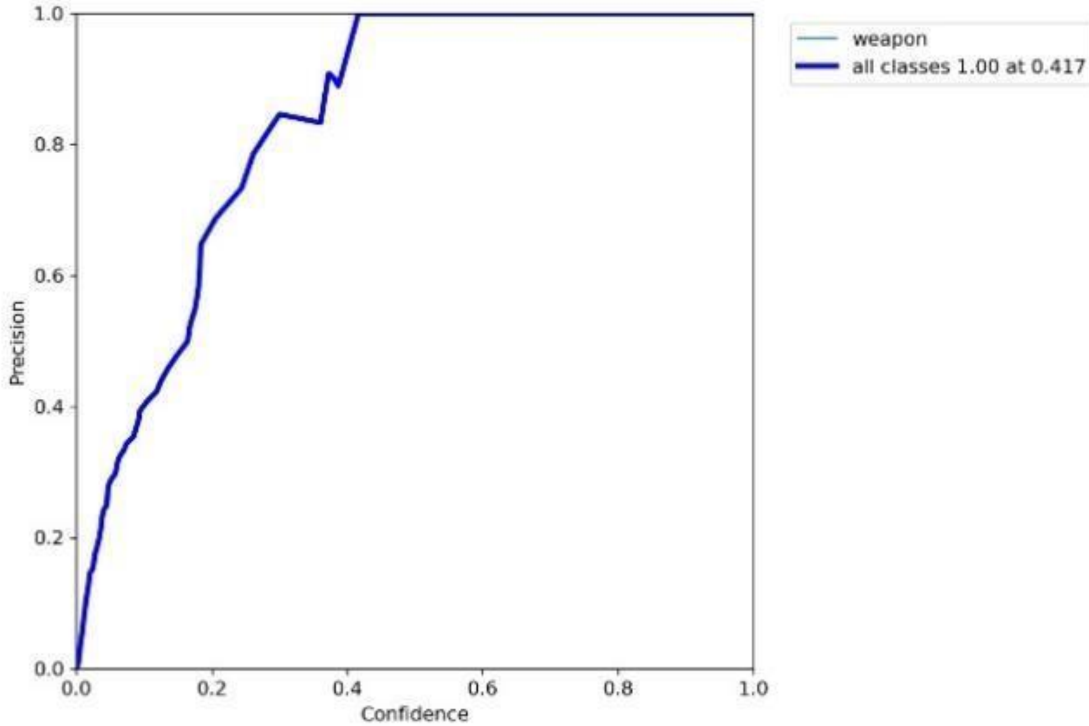

Figure 3

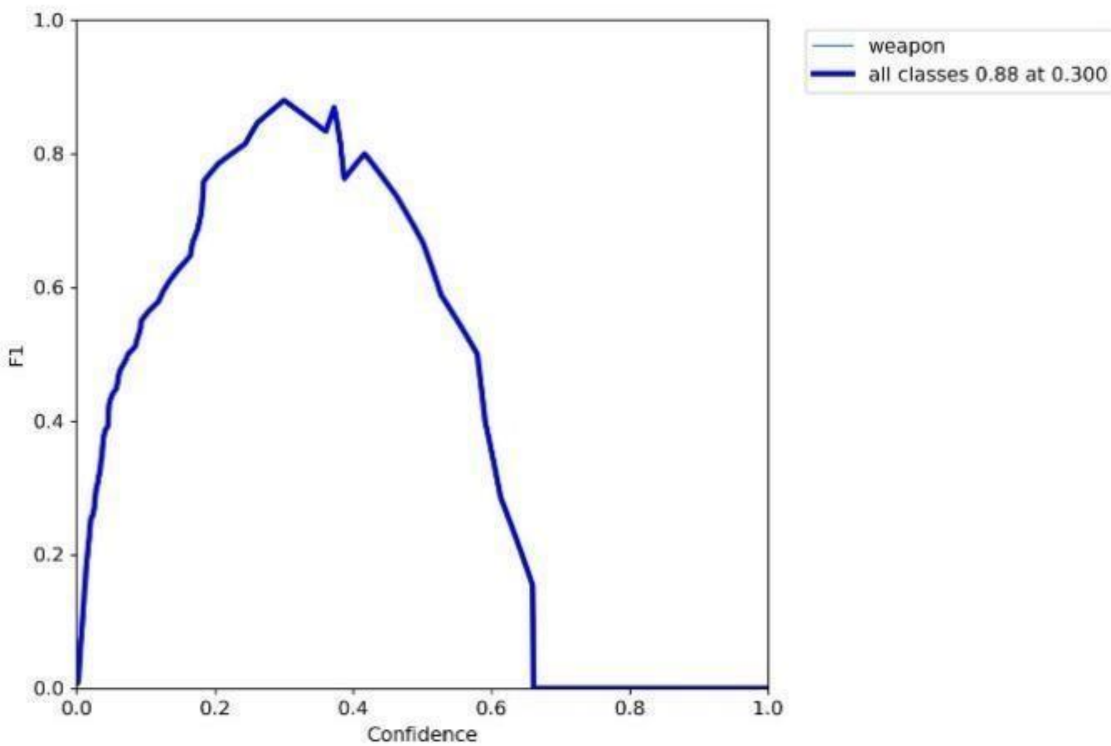

Figure 4

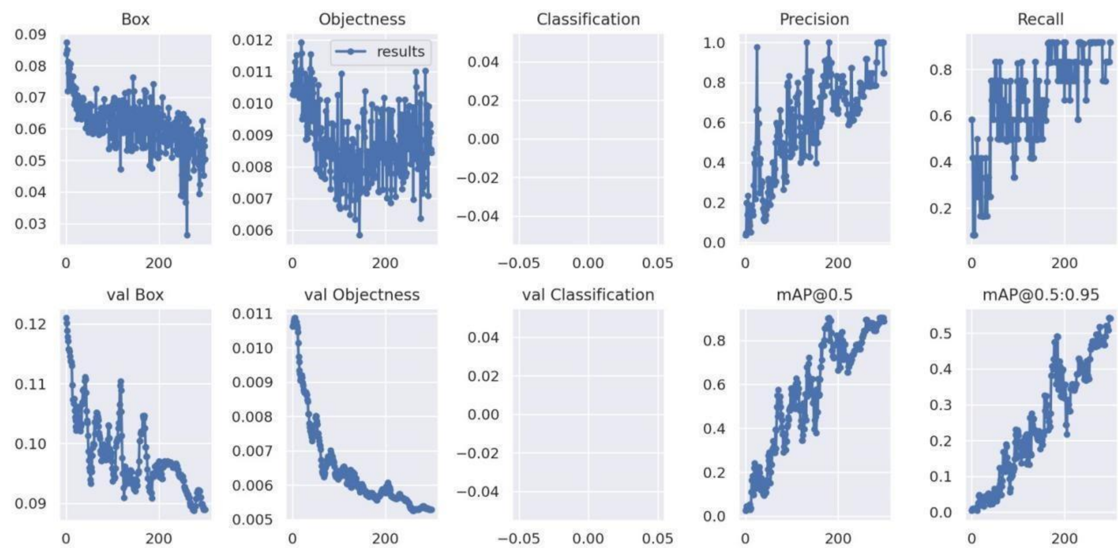

Figure 5

Testing:

For testing the model we have used 10% of our dataset, and the accuracy of the model is more than 90%, which is a satisfactory result.
And this is how we trained the dataset, here are some sample images

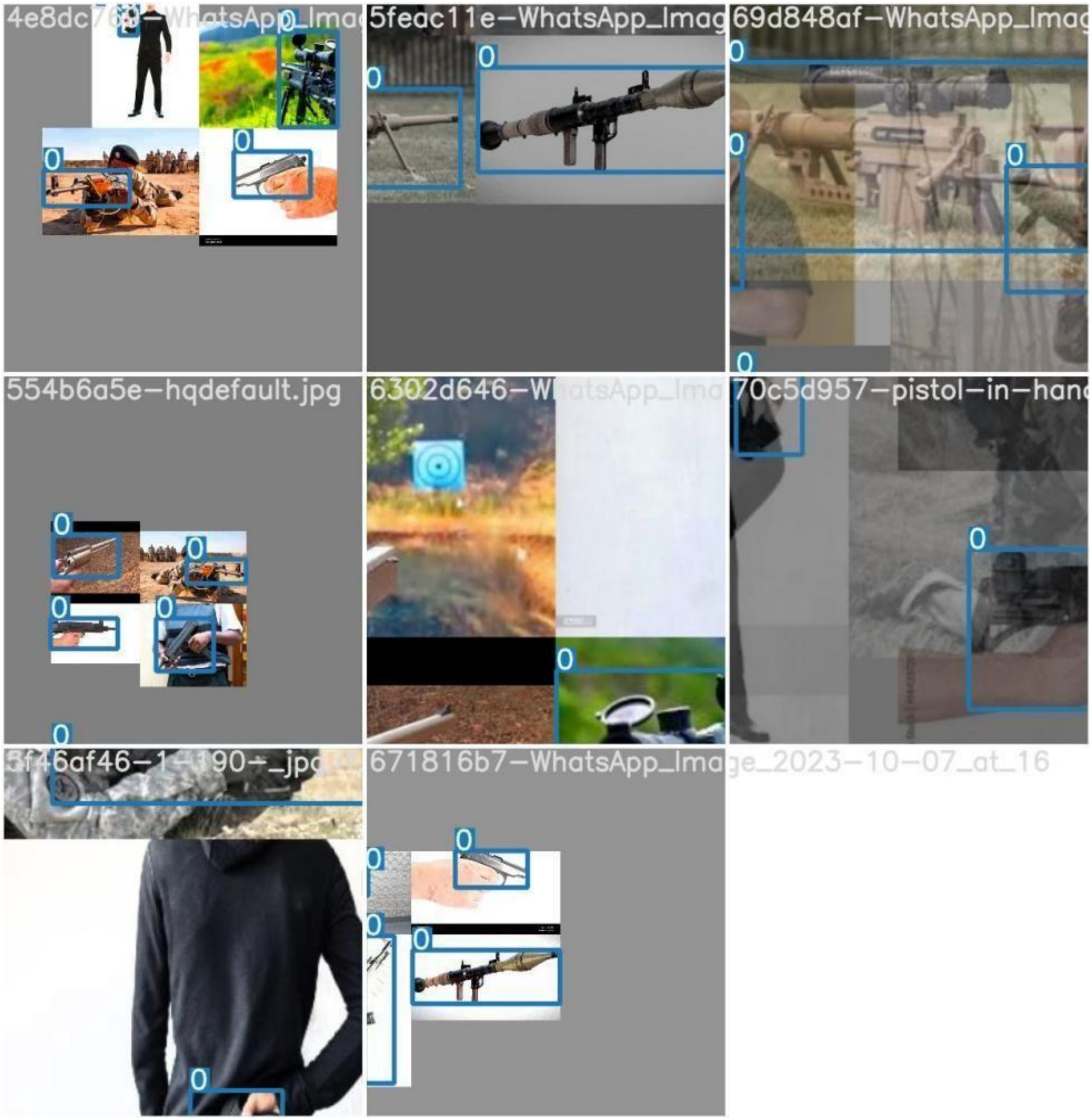
5feac11e-WhatsApp_Imag
69d848af-WhatsApp_Imag
554b6a5e-hqdefault.jpg
70c5d957-pistol-in-han
671816b7-WhatsApp_Image_2023-10-07_at_16

Figure 6

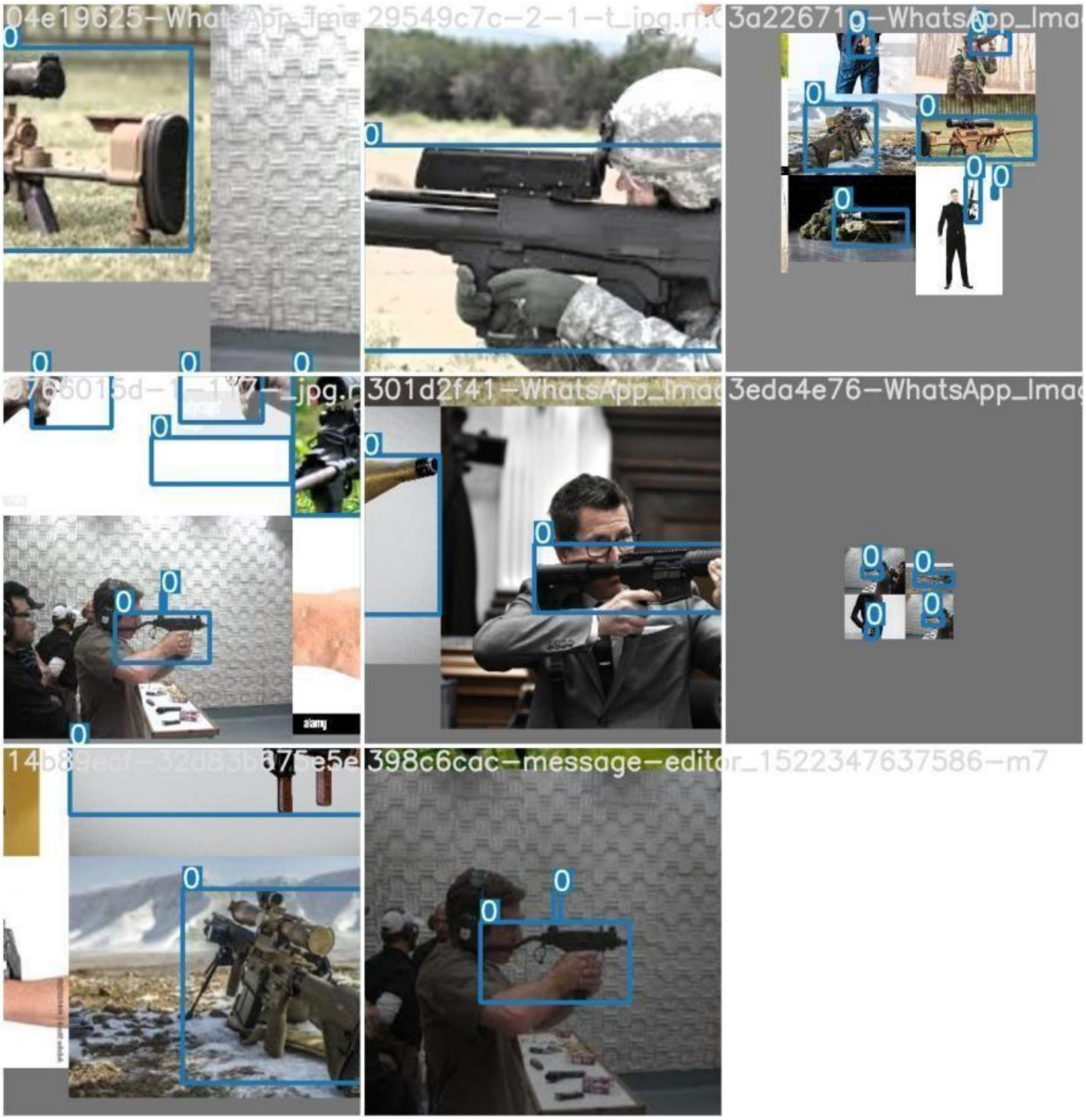
04e19625-WhatsApp_Ima
29549c7c-2-1-t_jpg.rf
301d2f41-WhatsApp_Imag
3eda4e76-WhatsApp_Imag
398c6cac-message-editor_1522347637586-m7
alamy

Figure 7


4e8dc769–WhatsApp_Imag
5feac11e–WhatsApp_Imag
69d848af–WhatsApp_Imag
554b6a5e–hqdefault.jpg
6302d646–WhatsApp_Ima
671816b7–WhatsApp_Image_2023–10–07_at_16

Figure 8

Figure 9

<u>Results</u>

When testing the model we have gained a successful outcome, and the weapon is getting highlighted. This is a sample outcome of the image that we uploaded

Figure 10

Figure 11

Conclusion

A weapon detection system using YOLO v7 (You Only Look Once version 7) has been implemented and evaluated, and the results have shown promise and the ability to improve security measures. Weapon detection is a crucial problem that needs to be addressed, and the YOLO v7 algorithm, noted for its effectiveness and accuracy in real-time item detection, has shown to be a useful tool in doing so.

The effective incorporation of YOLO v7 into this project, which was centered on weapon detection, demonstrated its capacity to correctly identify weapons in a variety of complex situations. The model demonstrated an excellent level of precision, recall, and mean average precision (mAP), demonstrating its effectiveness in identifying weaponry in a variety of scenarios.

The YOLO v7 model has a number of noteworthy advantages, including real-time processing capabilities that allow quick reactions to possible threats. This function is essential in security applications since prompt detection and action can mean the difference between life and death. Additionally, the model's high accuracy ensures consistent detection while minimizing false negatives and false positives.

The system's ability to adapt and scale is demonstrated by the YOLO v7 model's effective integration with a variety of datasets, including various scenarios and weapon kinds. The model can be improved for use in broader security applications, and its adaptability paves the path for future developments and extensions, such as the addition of new weapon types.
In summary, the incorporation of YOLO v7 into a weapon detection system has the potential to improve security protocols and increase public safety by enabling quick and precise identification of weapons. The likelihood of subsequent improvements and extensions emphasizes the significance of this project and demonstrates its value in developing security technology.

The accomplishment of this project serves as a springboard for further study and development, with the ultimate aim of developing stronger and more effective weapon detection systems, ultimately resulting in a safer and secure society.

References

- Application for threat surveillance using Python and Yolo-V7
- Edge Computing-Based Crime Scene Object Detection from Surveillance Video Using Deep Learning Algorithms
- A Multi-Weapon Detection Using Synthetic Dataset and Yolov5
- Weapon Detection from Surveillance Images using Deep Learning
- Effective Deep Learning Technique for Weapon Detection in CCTV Footage
- Real-time weapon detection using Drone
- Dataset

ORIGINALITY REPORT

14 %	12%	8%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	learnopencv.com Internet Source	5 %
2	github.com Internet Source	4 %
3	ieeexplore.ieee.org Internet Source	1 %
4	Deepak R. Hawale, Pavin S. Game. "Real-time weapon detection using Drone", 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, 2022 Publication	1 %
5	Submitted to Shrewsbury Sixth Form College Student Paper	1 %

6	R J Anandhi. "Edge Computing-Based Crime Scene Object Detection from Surveillance Video Using Deep Learning Algorithms", 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), 2023	Publication	1%
7	medium.com	Internet Source	<1%
8	Submitted to Sreenidhi International School	Student Paper	<1%
9	hackmd.io	Internet Source	<1%
10	worldwidescience.org	Internet Source	<1%
11	Robert Manuel, Bota Ioana, Anca Gavrilas. "Application for threat surveillance using Python and Yolo-V7", 2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH), 2023	Publication	<1%

12	blog.codinginvaders.com	Internet Source	<1%
13	doaj.org	Internet Source	<1%
14	www.mdpi.com	Internet Source	<1%
15	dev.to	Internet Source	<1%