International Journal of Engineering Sciences & Emerging Technologies, Dec. 2024
ISSN: 22316604 Volume 11, Issue 4, pp: 746-752 ©IJESET

WEAPON DETECTION SYSTEM USING DEEP LEARNING ALGORITHM

Neha Gupta, Shaurya Sarswat, Vanshita Chauhan, Riya Pandey

Department of Computer Science Engineering, Moradabad Institute Of Technology, Moradabad, India

discoverngp@gmail.com, shauryasarswat01@gmail.com, vanshitachauhan709@gmail.com, riyapandey1014@gmail.com

ABSTRACT

The escalating threats posed by firearms necessitate advanced weapon detection systems for enhanced security. Our research introduces a novel weapon detection system based on YOLOv8, capable of real-time threat identification. By leveraging deep learning, our system provides high accuracy with minimal false positives, augmenting surveillance measures in diverse environments. It can be deployed in airports, schools, stadiums, and urban centers, offering proactive security measures to protect lives and prevent potential security breaches. This research contributes to the development of intelligent security solutions, fostering safer communities and addressing the urgent need for robust threat detection systems.

KEYWORDS

FPN, CNN, YOLO, mAP

1. INTRODUCTION

In an era marked by heightened security concerns, the development of efficient and accurate weapon detection systems is imperative to safeguard public spaces. Traditional security measures often prove inadequate in addressing modern threats, underscoring the need for advanced technologies such as deep learning and computer vision[1]. This paper introduces a novel approach to weapon detection utilizing YOLOv8 (You Only Look Once version 8), a state-of-the-art object detection model renowned for its real-time performance and high accuracy. The primary objective of this research is to devise a robust weapon detection system capable of swiftly and accurately identifying various types of weapons including grenade, knife, pistol, rifle and missile in diverse environmental conditions. To achieve this, we have taken into consideration various object detection[2][6] based deep learning models like Region-Based Convolutional Neural Networks (R-CNN), Faster Region-Based Convolutional Neural Networks (Faster R-CNN) and Mask Region-Based Convolutional Neural Networks and You Only Look Once (YOLO) and depending upon the accuracy and prediction time requirements of the project we have finalized to use the YOLOv8. YOLOv8 is a faster approach with good detection accuracy supporting the problem statement of the quick and accurate detection of weapons in the crowded areas. You Only Look Once (YOLO) models directly predict bounding boxes and class probabilities from image grids, offering speed and simplicity. The weapon detection system under consideration employs YOLOv8, selected for its exceptional efficiency and accuracy in object detection. YOLOv8's capacity for real-time processing, coupled with its ability to uphold high precision, renders it an optimal choice for the task at hand. Its architectural design encompasses a backbone convolutional neural network (CNN) complemented by detection heads, enabling the precise prediction of bounding boxes and class probabilities. Through rigorous training on a comprehensive dataset comprising varied instances of weapons and non-weapon items, the system achieves a robust understanding of potential threats, enhancing its effectiveness in security applications. The architecture of YOLOv8 seamlessly integrates a deep CNN, ensuring efficient and accurate object detection. Distinguished by its streamlined design, YOLOv8 employs a singular neural network to directly infer bounding boxes and class probabilities from input images, facilitating real-time processing. Within this architecture, the backbone network undertakes feature extraction, while detection heads operating at multiple scales refine predictions. Leveraging advancements like feature pyramid networks (FPN) and optimization techniques, YOLOv8 strikes a harmonious balance between speed and precision, rendering it a preferred solution across diverse applications, notably including weapon detection. Its adaptable architecture enables effortless integration into surveillance systems and security frameworks, markedly enhancing threat detection capabilities. Comprehensive testing is undertaken to assess the efficacy of the proposed weapon detection system. Evaluation involves rigorous examination across diverse benchmark datasets and real-world surveillance footage sourced from varied environments. Precision, recall, and mean average precision (mAP) metrics are meticulously employed to gauge detection accuracy and system robustness. Additionally, qualitative analysis is conducted to scrutinize the system's proficiency in addressing complex scenarios, including occlusions, cluttered backgrounds, and varying viewing angles. Through systematic experimentation and thorough assessment, the system's performance under challenging conditions is thoroughly scrutinized, ensuring its suitability for real-world deployment with confidence.

2. SYSTEM OVERVIEW

The Real-time Weapon Detection System that we are introducing aims to improve security by detecting and recognizing various weapons such as grenades, missiles, pistols, rifles, and knives in a live video broadcast stream, with input in the form of an Internet Protocol (IP) address that hosts the video stream array. The system uses a You Only Look Once (YOLO) strategy for object detection, fine-tuning the YOLOv8 model with the weapons dataset. When a possible weapon with a frame of the object and a security alert is found, the technology immediately alerts the authorized personnel to the security dangers. The components of the system include :

2.1. User Interface Module

The Internet Protocol (IP) based camera detection interface, is a key component in our research. This interface blends in perfectly with our system, providing an efficient way to connect to the input camera. The Internet Protocol address of the camera is provided in the input field which on click of a stat button connects to the input camera and leads to the detection script. This method allows us to apply our concept to the camera apparatus that is currently on the risk-prone sites.

2.2. YOLOv8 Model

We are using the YOLOv8 model that is imported in the form of a python package that comes as a part of the Ultralytics python module. We have used the cli to train and fine-tune the yolov8 model by providing the weapons dataset and using the best weights file to finally perform the prediction on the input coming in the form of a video from a live feed Video source. The modes that we have explored with the model are "train" for fine-tuning the model by training on the weapons dataset and "predict" to identify the objects.

2.3. Detection Module

The real-time weapon detection module is meant to accept input in the form of a live video stream from a camera whose Internet Protocol address was provided by the user interface using the openCV script and Python's torch module. In order to start the current model state, it eventually uses the Ultralytics module to import the Yolov8 model and loads the last-best fine-tuned weights file, or "Best.pt." then uses yolo-based detection in the "predict" mode to ascertain in real time whether the supplied frame has a weapon or not.

2.4. Alerting Module

The alerting module uses the Python Simple Mail Transfer Protocol (SMTP) module for email delivery and authentication in order to interface with the implemented detecting modules. The module receives two addresses: a base address used for application authentication and a to_address used to receive alert-related mail. The alerting system quickly generates an email including the object and its accompanying bounding box, or Object Bounding Box (OBB) frame, upon detecting a true positive item with threshold confidence percentage, sends it to the recipient's email address.

3. METHODOLOGY USED

The following steps were followed as shown in fig.1 from starting till end to make this weapon detection system using the YOLOv8 model[5].

3.1. Data Retrieval and Dataset Preparation

We acquired the dataset with 10,000 photos associated with the problem statement. We fetched the dataset using the Roboflow dataset API. The dataset format that we selected for our training experience is Object Bounding Box notation (YOLO-OBB).

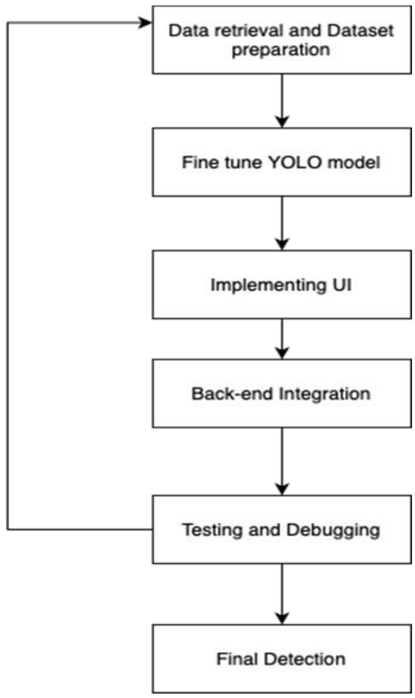

Figure 1. Workflow of Proposed Approach

There were five different classes ie. Pistol, Rifle, Missile, Grenade and Knife. The dataset exhibits a complete class balance but has a slight under-representation in the “Knife” and “Missile” categories. Because of the variety of classes available the model may be trained thoroughly and perform well on a large range of object detection.

3.2. Fine Tune YOLO Model

We have utilized the Ultralytics package[5] to import the YOLOv8 model, leveraging its command line training mode and fine-tuning beginning with the YOLOv8X.pt weights, we initiated the training with the mode set to “train” employing Google Collab, we ensured the computational power of CUDA-enabled devices. The fine-tuning process spanned 400 epochs in total divided into the intervals of 20,30 and 40 epochs. To facilitate continuity we iteratively used the “last.pt” weights file from the preceding session of training. This systematic approach allowed us to progressively refine the model’s performance and get us closer to better detection rates and reduced false detections. Fine-tuning improves the detection accuracy and improves confidence.

3.3. Implementing User Interface

The user-interface presents a front-end view where users can input the internet protocol (IP) address of a camera module, which streams live video broadcasts. This Internet Protocol (IP) address serves as the entry point for the data. Additionally, the interface includes a “Start” button that triggers the entire process of the object detection on the incoming video stream. Behind the scenes, the back-end module handles the processing of the video stream, executing object detection algorithms to analyze and identify objects within the live feeds.

3.4. Back-end Integration

The back-end integration to the front-end takes the input Internet Protocol address and begins to receive the continuous live feed using the open-cv script, followed by the object detection employing the loading of the YOLOv8 best set weights ie. Best.pt file, then detecting the objects per-video frame, and subsequently using the python's SMTP module to return the object detected frame along with an alert signal to the authorized email address.

3.5. Testing and Debugging

Testing and debugging are iterative development steps that we have implemented into our project, which was constructed using an agile development approach. The foundation of the initiative has been the ongoing development and application of the diverse datasets combined with tailored training opportunities. Three validation checks were conducted to determine the accuracy of the detection: image-based detection, video-based detection, and, in the last stage, live video stream-based detection.

3.6. Final Detection

The project deployment involves connecting the surveillance equipment and the video stream device over WiFi after the video stream device has been set up at the desired location. uploading the best training weights file ie. “Best.pt”, configuring the project's environment, giving the video stream device's IP address, modifying the authority emails to the desired email addresses, and then carrying out the detection.

4. RESULT ANALYSIS

The dataset[3] used is taken from roboflow.org[4] which is already annotated and is publicly available for access and is fetched by the means of the roboflow inference hosted API as shown in fig.2 is taken from roboflow[3].

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="MkkUtwIZWSewtgfN0YUh")
project = rf.workspace("test-7awfy").project("weapon-detection-f1lih")
dataset = project.version(1).download("yolov8-obb")
```

Figure 2. Roboflow Inference hosted API for Dataset

The dataset follows the You Only Look Once - Object Bounding Box (YOLO-OBB) format and is having the class management as described in the table.1

4.1. Data Retrieval and Dataset Preparation

We acquired the dataset with 10,000 photos associated with the problem statement. We fetched the dataset using the Roboflow dataset API. The dataset format that we selected for our training experience is Object Bounding Box notation (YOLO-OBB).

Table 1. Weapon Detection Dataset Class management.

S.No	Classes Categories	Count	Class Balance
1	Pistol	4,410	Balanced
2	Rifle	3,874	Balanced
3	Grenade	2,490	Balanced
4	Missile	2,223	Under Represented
5	Knife	1,984	Under Represented
Total		9,633	

The dataset has 0 missing annotations and 24 Null examples, there are total of 14,981 annotations with an average of 1.6 per image divided across 5 classes [0: Pistol, 1: Rifle, 2: Grenade, 3: Missile, 4: Knife]. The average image size is 0.41 Million Pixels (mp)[3]. The following dataset was implemented to fine tune the YOLOv8 model and we have got the final result as described using the fig.3 as confusion matrix at 400 epochs and fig.4 through graphs plotted against box loss (box_loss), class loss (cls_closs), and focal loss (dfl_loss) at training and validation. Also the performance metrics as precision, recall and mean average precision (mAP) at training and validation were shown.

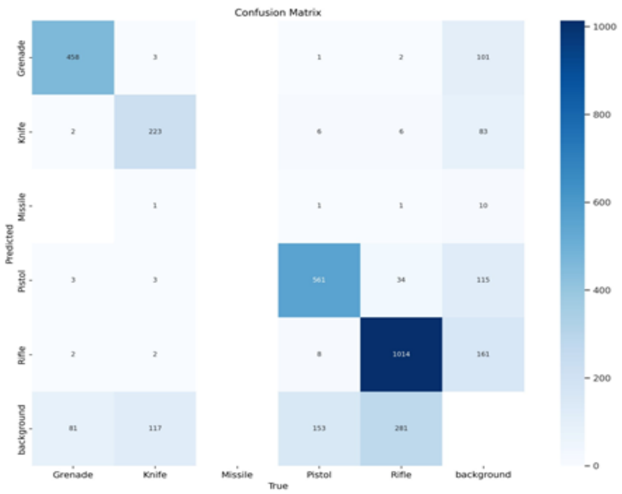

Figure 3. Confusion Matrix at 400 epochs

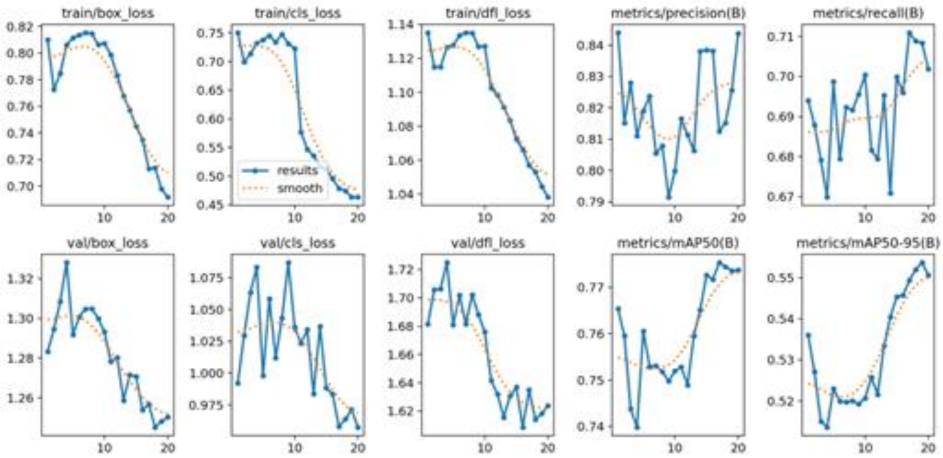

Figure 4. Performance metrics graphs at 400 epochs

On testing with the live video feed the detections at 75% confidence were shown in fig.5

Figure 5. Weapon detection on live video feed

5. CONCLUSION

Our weapon detection technology, which uses the YOLOv8 model for seamless real-time object recognition, has been proven effective through significant research and rigorous experimentation. This method offers robust potential threats quickly and precisely, which is a huge development in security technology.

With an eye toward the future, we remain unwavering in our resolve to promote the incorporation of such revolutionary technology into more comprehensive security infrastructures. Even if our results show promise, further development and ethical considerations need to be taken into account. With more study and cooperation, integrating such technology into public safety measures holds out a lot of promise.

6. FUTURE PROSPECTIVE

The application of weapon detection technology on IP-based detection CCTV cameras can be generalized in the future to work on input feeds from multiple video sources at the same time, and the approach can be expanded by implementing later versions of the YOLO models (e.g., YOLOv9)[7] after the models have proven their performance.

It is necessary to expand the support for commodity hardware by broadening the acceptance of input and developing a unique training environment with images that differ from the one shown in the respective video environment[8].

REFERENCES

[1] M. Monika, Udutha Rajender, A. Tamizhselvi, Aniruddha S Rumale, "Real-time object detection in videos using deep learning models", *ICTACT Journal on image and video processing, 2023, Volume. 14*

[2] Chang Ho Kang, Sun Young Kim, "Real-time object detection and segmentation technology: an analysis of the YOLO algorithm", *JMST Adv. 5,69-76, 2023*

[3] https://universe.roboflow.com/test-7awfy/weapon-detection-f1lih/

[4] https://universe.roboflow.com/

[5] https://github.com/ultralytics/ultralytics

[6] Delong Qu, Weijun Tan, Zhufu Liu, Q Yao, Jingfeng Liu, "A Dataset and system for real-time gun detection in surveillance video using deep learning" in *arXiv preprint arXiv: 2105.01058, 2021*

[7] Chien-Yao Wang, I-Hau Yeh, Hong-Yuan Mark Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information" in arXiv preprint arXiv: 2402.13616, 2024

[8] Yuan Zhu, Yanqiang Wang, Yadon An, Hong Yang, Yiming Pan, " Real-Time Vehicle Detection and Urban Traffic Behaviour Analysis Based on UAV Traffic Videos on Mobile Devices" in arXiv preprint arXiv: 2105.01058, 2024

Authors

Neha Gupta is an assistant professor in the Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. She had done B. Tech, M.Tech and now pursuing Ph. D. Her research area involves Machine learning, Deep Learning, Data Science and Data Security Measures.

Shaurya Sarswat is a B.Tech 4th Year Student in the Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. His research interests include Machine Learning, Deep learning and Video data processing.

Riya Pandey is a B.Tech 4th Year Student in the Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. Her research interests include Machine Learning, Deep learning and Video data processing.

Vanshita Chauhan is a B.Tech 4th Year Student in the Computer Science and Engineering Department of Moradabad Institute of Technology affiliated with Dr. A.P.J. Abdul Kalam Technical University. Her research interests include Machine Learning, Deep learning and Video data processing.