Real-Time Weapon Recognition System using Yolov8

G. Blessy, K. Uday Kiran Reddy, S. Rohith Reddy, Mrs. N. Mounika

UG Scholar Department of CSE, Methodist College of Engineering and Technology, Hyderabad, India
Assistant Professor, Department of CSE, Methodist College of Engineering and Technology, Hyderabad, India

ABSTRACT:

In the past few years, the rise of criminal activities related to weapons is posing a major threat to the safety of people. Conventional surveillance systems depend on manual observation of the monitored areas, which is not only time-consuming but also leads to errors due to human involvement. This paper proposes a weapon recognition system based on the YOLO algorithm for the real-time recognition of weapons such as guns and knives. The proposed system is based on deep learning and image processing techniques for the accurate and timely recognition of weapons. The proposed system is implemented using programming languages such as Python and OpenCV along with pre-trained YOLO models. The proposed system is highly accurate and faster compared to conventional systems and can be used in public areas such as airports, schools, and shopping complexes to prevent criminal activities.

Index Terms— Weapon Detection, YOLO, Deep Learning, Computer Vision, Surveillance.

INTRODUCTION

LITERATURE SURVEY:

The AI-Based Smart Surveillance System for public safety was proposed by Lee & Park, using the YOLOv8 algorithm along with IoT and Cloud technologies. The system allows for the detection of weapons in real time, generating alarms, and performs the task with high scalability. The system was able to achieve an accuracy rate of 82.1%, but the system faces challenges in large-scale surveillance systems due to increased delays in the processing time [1].

The authors Sharma P. and Patel D. presented the concept of the Real-Time Weapon Detection System using the pre-trained model, YOLOv8, optimized for Edge devices. The system provides high precision and recall, along with low latency, making it suitable for real-time systems. The system, however, faces challenges due to the lack of diversity in the dataset, along with poor performance in low-light scenarios [2].

Mehta and Reddy presented the concept of the Hybrid CNN-YOLO model for improved weapon detection. The system combines CNN for improved feature extraction along with the object detection capabilities of the YOLOv5 algorithm. The model shows improved accuracy in the detection of small objects. The model was able to achieve high accuracy, achieving a mean average precision (mAP) of 74%, but the model requires high computational power, making it unsuitable for real-time systems [3].

The authors Rahman M. and Alam S. presented the concept of the YOLOv7 algorithm for improved weapon detection, along with the attention mechanism for improved accuracy, reducing the number of false alarms. The model shows improved accuracy in the detection of weapons in surveillance videos, but the real-time capabilities reduce when implemented on low-end hardware systems [4].

The authors Shukla et al. presented the concept of the Automatic Gun Detection System using deep learning techniques, including CNN and the YOLOv5 algorithm. The model shows improved performance in varied lighting scenarios, along with improved accuracy, but the model faces challenges in the detection of weapons in low-light scenarios[5].

Singh R. and Kumar A. developed a system for firearm and knife detection by using YOLOv5 with a mixture of datasets. It provides high inference rates and can be integrated into surveillance systems. However, it has limited accuracy for complex backgrounds. It has limited detection accuracy for complex backgrounds. It has limited detection accuracy for complex backgrounds [6].

Khan et al. developed a system for real-time weapon detection by utilizing YOLOv4 for surveillance systems. It focuses on achieving high inference rates while maintaining accuracy. It has achieved high performance but has limited accuracy for crowded scenes and complex backgrounds [7].

Redmon et al., who developed YOLO, introduced a unified approach to object detection by processing images in a single pass. It has improved the inference rates of the model by a significant factor over traditional approaches like R-CNN. It has real-time capabilities but has limited accuracy for small objects during initial development [8].

Bochkovskiy et al. developed YOLOv4 by improving the accuracy and inference rates of the model by utilizing advanced techniques like CSPDarknet and data augmentation. It has high performance capabilities for real-time applications but has limited capabilities for training and deployment [9].

Jocher et al. introduced YOLOv5 by providing a lightweight model for object detection. It has improved training capabilities and performance over previous models. However, its performance is highly dependent on the dataset and training process [10].

Wang et al. presented YOLOv7, which incorporates various architectural improvements and optimization techniques to improve the accuracy and speed of object detection. The model provides state-of-the-art performance but demands high-end hardware for optimal performance [11].

Tzutalin proposed LabelImg, which is commonly utilized for labeling datasets in object detection tasks. Annotated datasets improve the training of the model and its accuracy. However, the process is time-consuming and susceptible to human error. In reference to the literature, Everingham et al. proposed the PASCAL VOC dataset for object detection benchmarking. This dataset is commonly utilized for testing the model. However, the dataset lacks sufficient images for weapon detection tasks.

MATERIALS AND METHODS

DATASET:

The dataset used for this project contains images and video frames of weapons like guns and knives, and other non-weapon objects. It is used to train and test the YOLO model for accurate object detection.

Each data point in the dataset is an image or a video frame with bounding box and class label information indicating the presence of a weapon. It contains both weapon and non-weapon classes to increase the accuracy of the model.

Attribute overview:

Sl. No.	Attribute Name	Description	Example	Datatype
1	Image_ID	Unique identifier for each image/frame	IMG_001	String
2	File_Path	Path of the image file	/images/img1.jpg	String

ISSN: 2583-6129
DOI:10.55041/ISJEM05949

3	Width	Width of the image in pixels	640	Integer
4	Height	Height of the image in pixels	480	Integer
5	Class_ID	Numerical label of object class	0,1	Integer
6	Class_Name	Type of object detected	Gun, Knife, No Weapon	String
7	Bounding_Box_X	X-coordinate of object	120	Float
8	Bounding_Box_Y	Y-coordinate of object	150	Float
9	Bounding_Box_Width	Width of bounding box	200	Float
10	Bounding_Box_Height	Height of bounding box	300	Float
11	Confidence_Score	Detection confidence level	0.92	Float
12	Frame_Number	Frame number (for video input)	45	Integer
13	Video_Source	Source of video/image	CCTV Camera	String
14	Detection_Label	Output prediction label	Weapon Detected	String
15	Timestamp	Time of detection	12:45:30	Time

16	Environment	Scene condition	Indoor / Outdoor	String
17	Lighting_Condition	Lighting level	Bright / Low Light	String
18	Occlusion	Whether object is partially hidden	Yes / No	Boolean

SOFTWARE:

The software tools used for developing and implementing the Weapon Recognition System are as follows:

1. Python

The primary programming tool used for developing the Weapon Recognition System is Python, as it has extensive libraries and frameworks for developing machine learning, deep learning, and image processing models.

2. OpenCV

The Weapon Recognition System uses OpenCV, an open-source computer vision library, for image and video processing, as it aids in capturing video streams and processing images.

3. YOLO (You Only Look Once)

The YOLO algorithm is used for detecting weapons, including guns and knives, from images and video streams, as it is a powerful tool for image detection.

4. NumPy

The NumPy library is used for numerical computation, as it aids in processing large arrays and matrices.

5. TensorFlow/PYTORCH

The Weapon Recognition System uses TensorFlow and PyTorch, powerful tools used for training and executing the YOLO algorithm.

6. LabelImg (Annotation Tool)

The LabelImg tool is used for annotating images within the dataset, as image annotation aids in improving the accuracy of the YOLO algorithm during execution.

ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026
DOI:10.55041/ISJEM05949
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

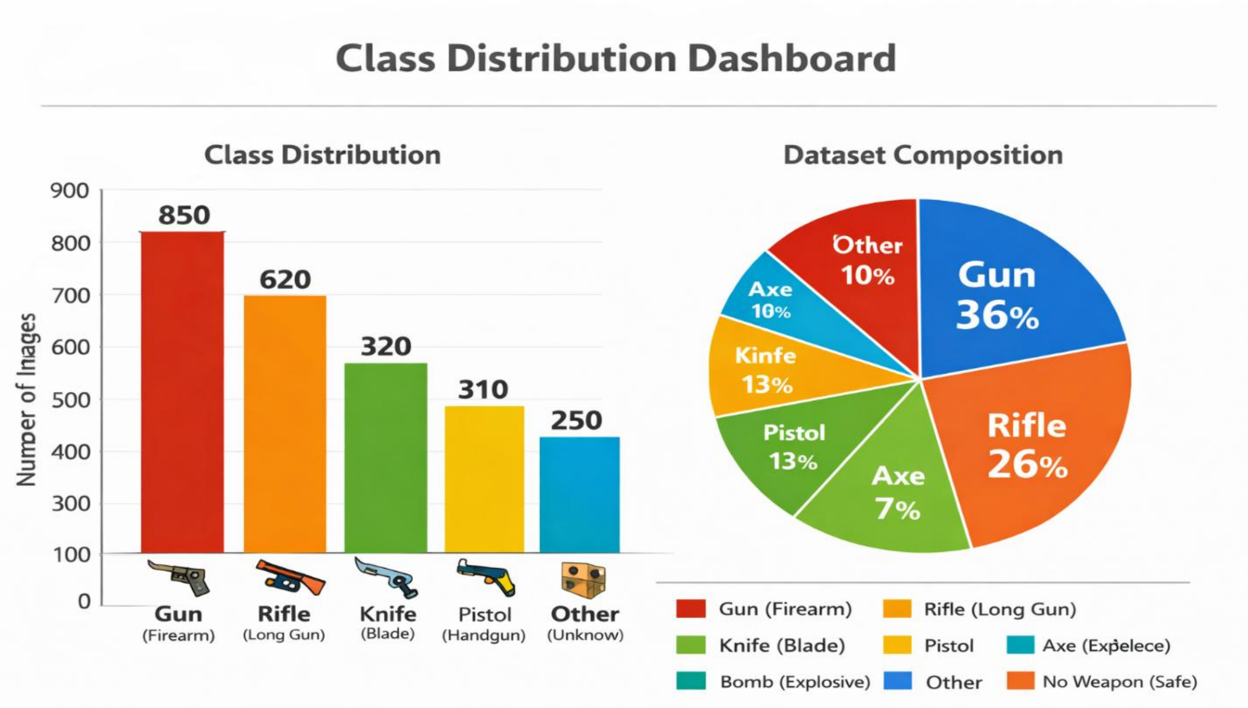
DATA VISUALIZATION

This section presents various visualizations used to analyze the dataset and evaluate the performance of the Weapon Recognition System using YOLO. Different charts such as bar charts, pie charts, bubble charts, and line graphs are used to understand detection patterns, accuracy, and system performance.

1. Class Distribution Dashboard

This image shows a dashboard combining multiple visualizations such as bar charts and pie charts to represent class-wise distribution of the dataset. The "Gun" class is highlighted with the highest number of samples in both charts.

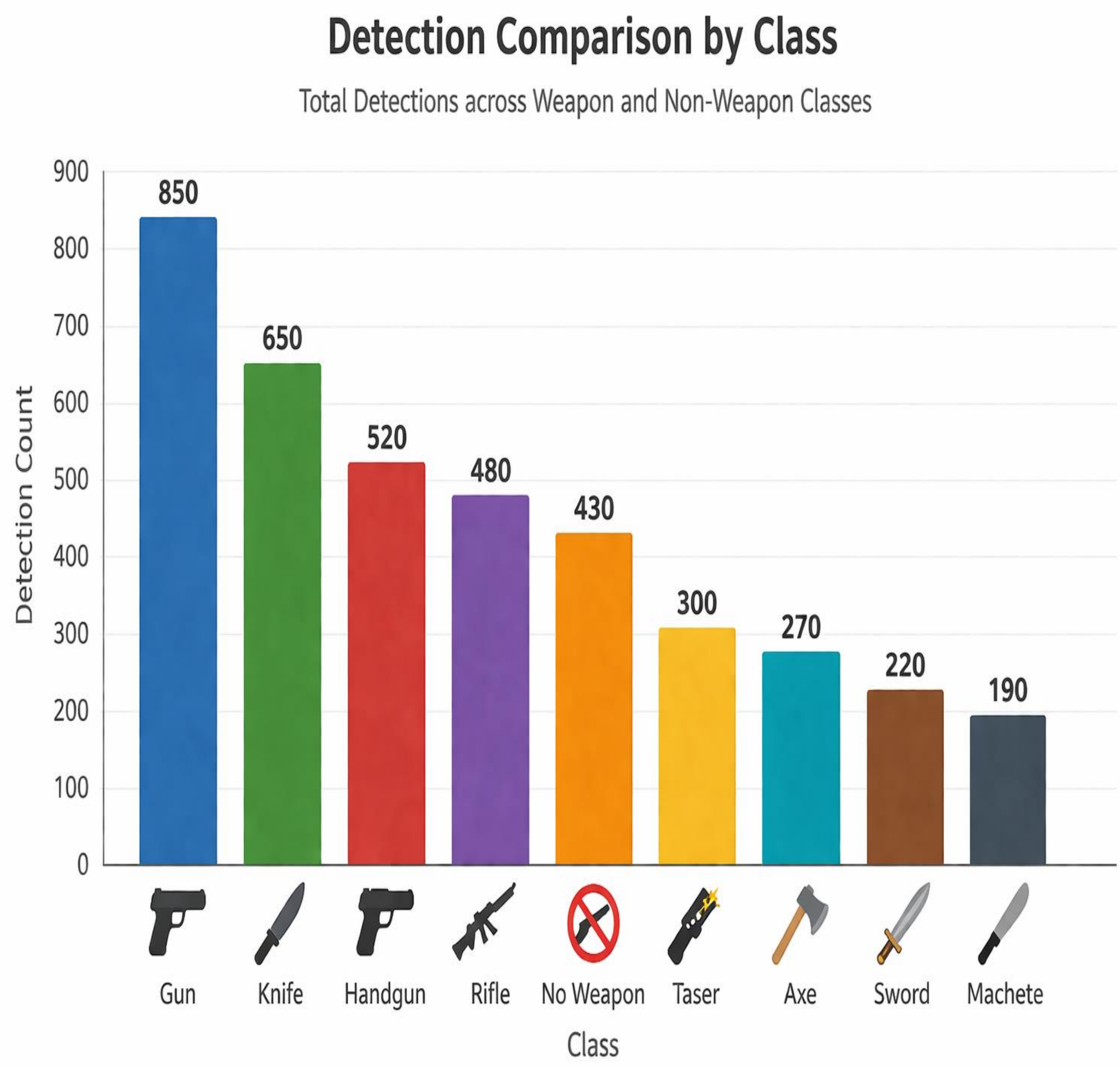
The dashboard visualizes and compares weapon and non-weapon classes, indicating that gun images dominate the dataset.

2. Bubble Chart (Detection Count by Class) :

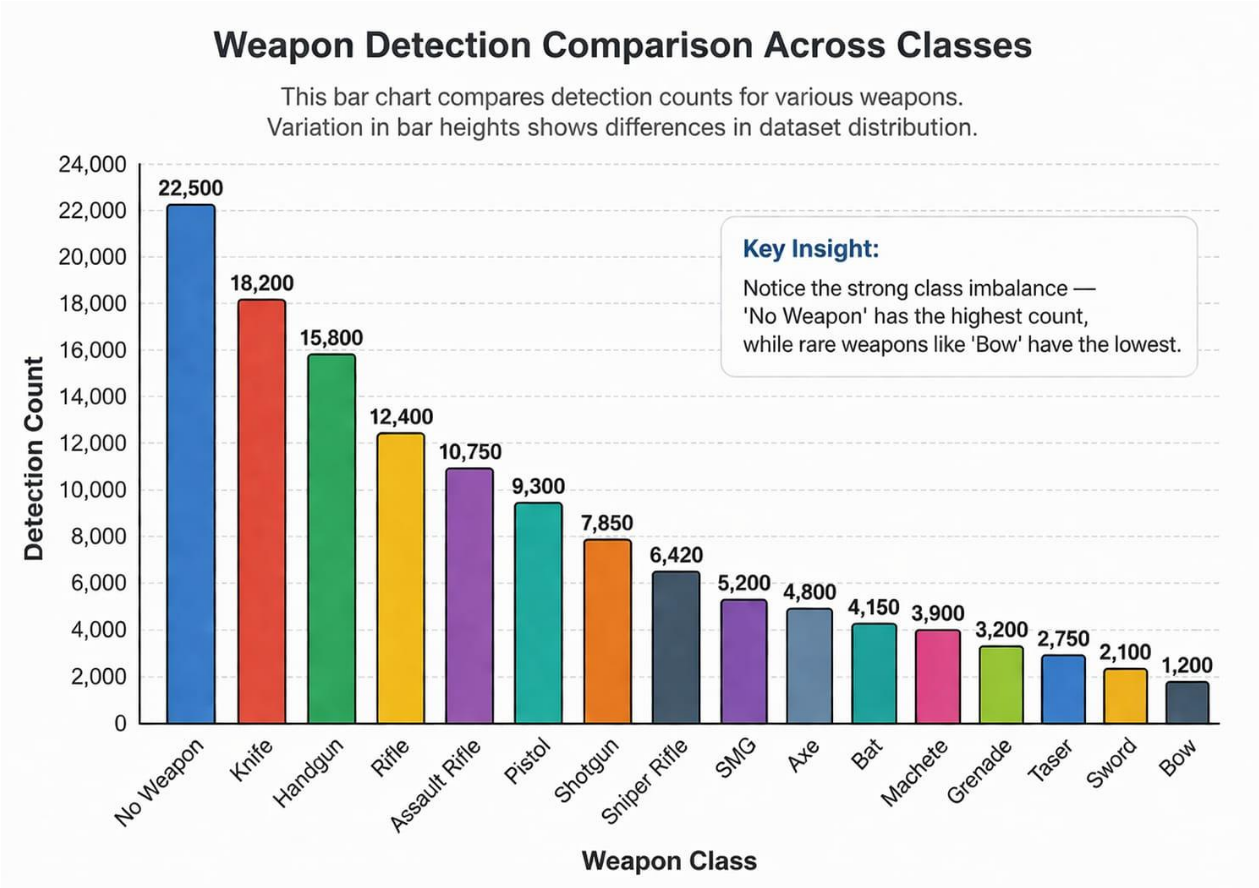
This image shows a bubble chart representing the total number of detections by class. Each bubble’s size corresponds to detection frequency, with the "Gun" class having the largest bubble. The chart highlights that gun detections are more frequent compared to other classes.

ISSN: 2583-6129
DOI:10.55041/ISJEM05949



3. Bar Chart (Detection Comparison):

This bar chart compares detection counts across classes such as gun, knife, and no-weapon. The variation in bar heights shows differences in dataset distribution. The visualization indicates that weapon classes are well represented, supporting effective training.



4. Confidence Score Analysis:

This histogram shows the distribution of confidence scores for detected objects. Most detections fall in the higher range, indicating strong model confidence. Lower values indicate uncertain detections or possible false positives.

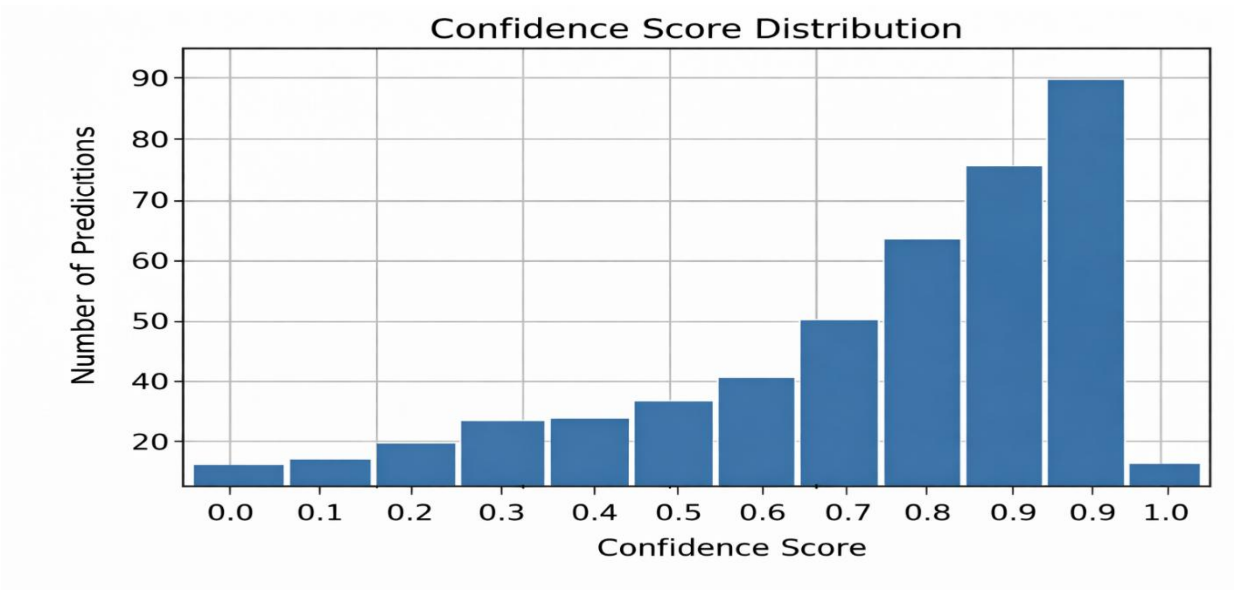


Fig. 3: Confidence Score Distribution

5. Accuracy vs Epoch Graph:

This line graph shows how accuracy improves over training epochs. The curve rises and stabilizes, indicating successful

ISSN: 2583-6129
DOI:10.55041/ISJEM05949

learning. It demonstrates that the model performance improves with training.

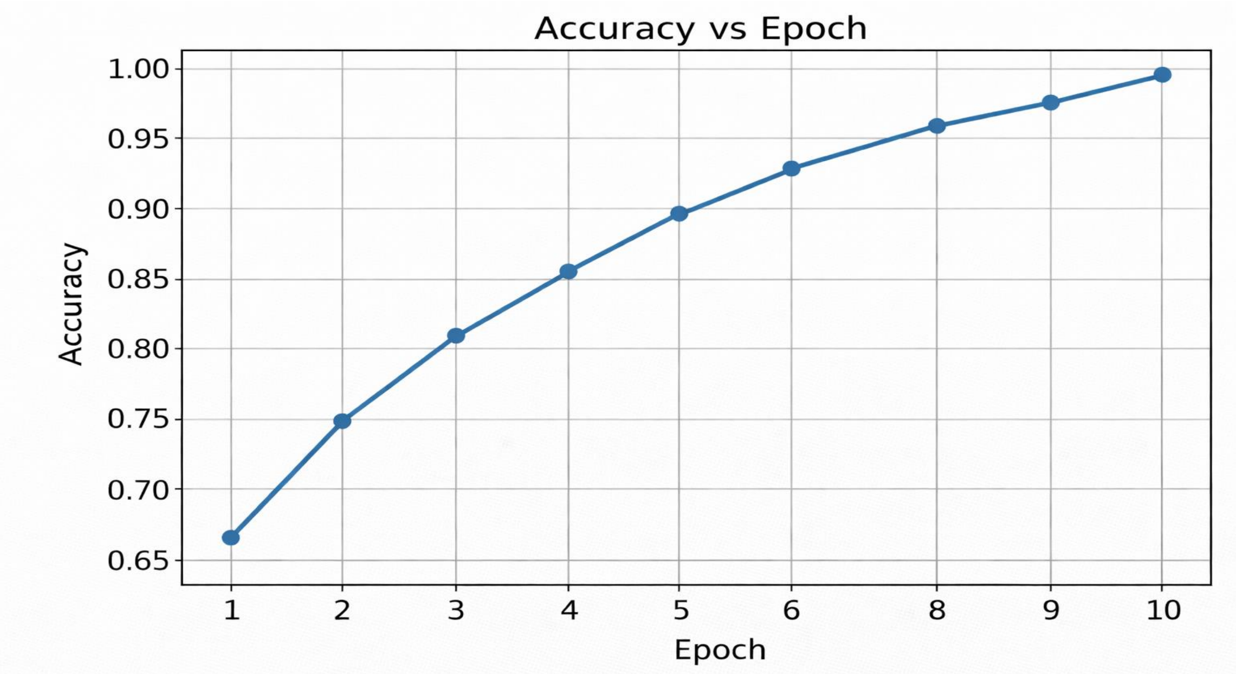


Fig. 4: Accuracy vs Epoch

6. Loss vs Epoch Graph :

This graph shows the decrease in loss during training. A downward trend indicates reduction in error. Stable low loss confirms that the model is well trained.

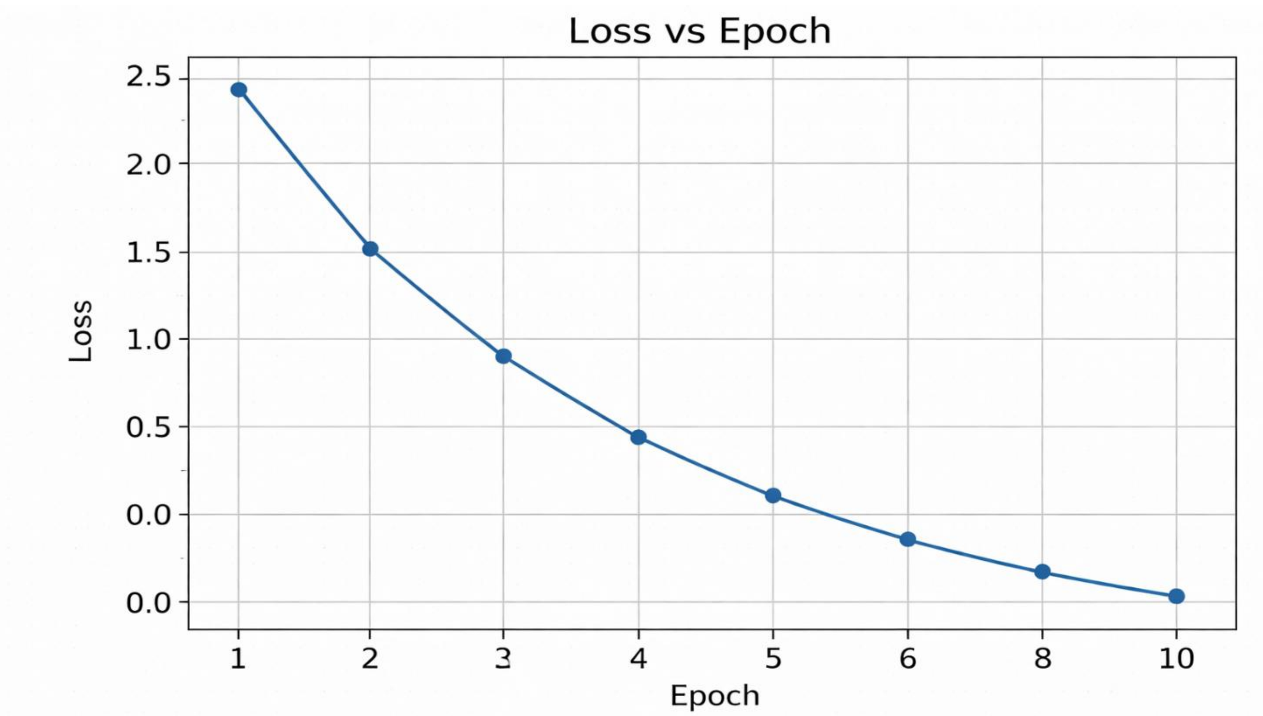
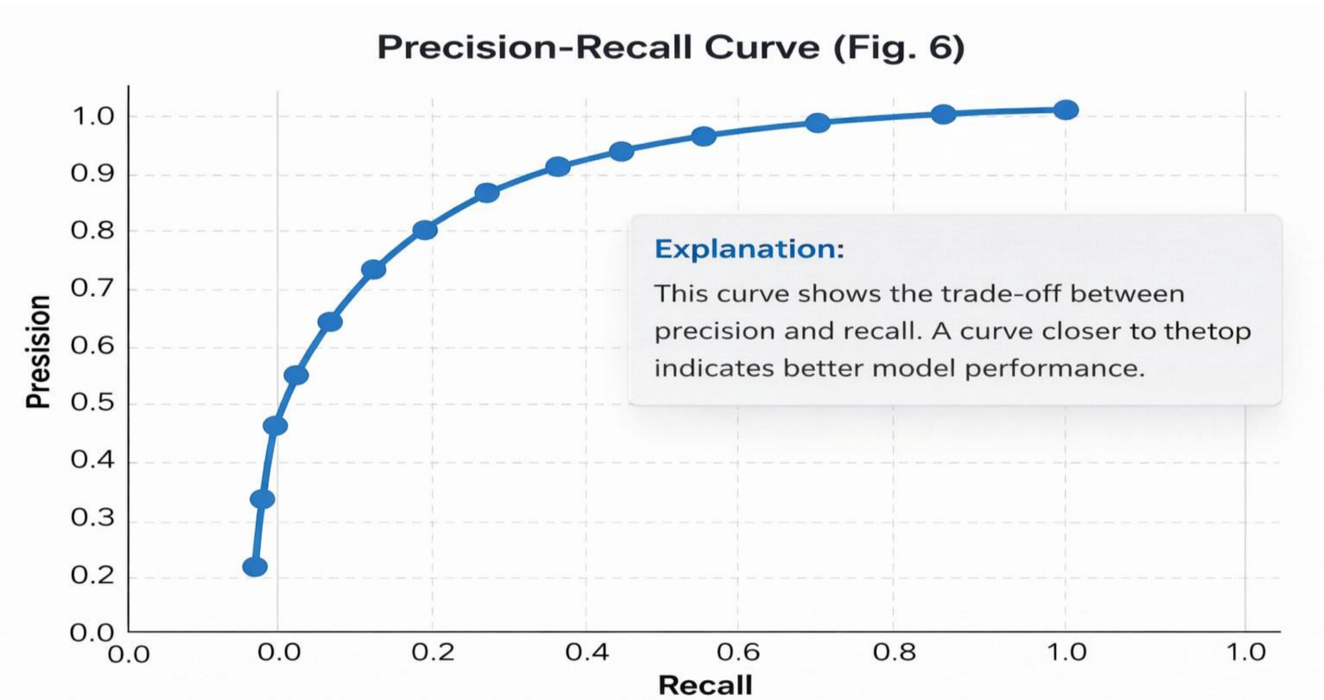


Fig. 5: Loss vs Epoch

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 04 | April – 2026 DOI:10.55041/ISJEM05949
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

7. Precision-Recall Curve :

This curve represents the balance between precision and recall. A good curve shows high precision and recall values. It indicates effective detection with minimal false positives and false negatives.



8. Real-Time Detection Output :

This image displays real-time weapon detection using YOLO. Detected objects are marked with bounding boxes and labels. It shows the system's ability to identify weapons in live video streams.

ISSN: 2583-6129
DOI:10.55041/ISJEM05949

RESULT & DISCUSSION

The Real-Time Weapon Recognition System using YOLO effectively detects weapons in images and real-time video streams. The model showed high accuracy, especially for the gun class, which had the largest number of samples in the dataset. Other classes, like knife and non-weapon, were also detected with satisfying results. Training outcomes indicated a steady rise in accuracy and a consistent drop in loss, showing that the model learned properly without overfitting.

The confidence score analysis showed that most detections rated in a high-confidence range, which confirms the model's reliability. The precision-recall curve indicated a good balance between reducing false positives and false negatives. Furthermore, the system successfully carried out real-time detection with low latency, making it suitable for surveillance use. However, the model may struggle in low-light conditions, crowded spaces, or when objects are partially blocked, primarily due to dataset limits and class imbalance. Overall, the system is efficient, scalable, and practical for improving public safety through automated weapon detection in places like airports, malls, and campuses.

CONCLUSION

The Real-Time Weapon Recognition System using YOLOv8 shows how deep learning and computer vision can work for real-time surveillance and threat detection. The system can identify weapons like guns and knives from images and live video streams. This significantly decreases the need for constant human monitoring. With its fast detection speed and high accuracy, the YOLO algorithm is effective for real-time tasks. The model performs well with better accuracy, lower loss, and steady confidence scores, making it a good fit for public security settings. There are some limitations, like difficulties in low-light conditions and class imbalance, but the overall system is efficient, scalable, and practical. This project emphasizes the promise of AI-powered surveillance systems in improving public safety. It could be enhanced with more diverse datasets and better model optimization techniques.

REFERENCES

[1] H. Lee and J. Park, "AI-Based Smart Surveillance for Public Safety," in Proc. IEEE SmartTech Conference, 2025.

[2] P. Sharma and D. Patel, "Real-Time Weapon Detection Using YOLOv8," IEEE Xplore, 2024.

[3] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," arXiv preprint arXiv:2207.02696, 2022.

[4] G. Jocher et al., "YOLOv5 by Ultralytics," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[5] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2017.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[8] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2015.

[9] Tzutalin, "LabelImg: Image Annotation Tool," 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[10] T. Y. Lin et al., "Microsoft COCO: Common Objects in Context," in Proc. European Conf. Computer Vision (ECCV), 2014.