ijamrsd.com
ISSN: 2583-6404
Journal of Advanced Multidisciplinary Research Studies and Development
Peer Reviewed Referred Journal
Bi-monthly
Open Access

AI-POWERED REAL-TIME FIREARM DETECTION AND ALERT SYSTEM USING YOLOV12M DEEP LEARNING MODEL

Savita Waghmode[1], Prof. Vidhate S.N.[2]

[1] *PG Student, Dept. of Electronics & Telecommunication Engineering, Dattakala Group of Institution, Faculty of Engineering, Pune, Maharashtra, India.*

[2] *Assistant Professor, Dept. of Electronics & Telecommunication Engineering, Dattakala Group of Institution, Faculty of Engineering, Pune, Maharashtra, India.*

Abstract

The increasing frequency of firearm-related inci-dents in public spaces demands intelligent, automated surveil-lance systems capable of real-time threat detection. Conventional closed-circuit television (CCTV) systems depend entirely on human operators, making them susceptible to fatigue-induced delays and missed detections. This paper proposes an AI-powered real-time firearm detection and alert system leveraging the YOLOv12m (You Only Look Once, version 12, medium variant) deep learning model. YOLOv12m is selected over its nano counterpart due to its significantly larger parameter count, deeper feature extraction capability, and superior mean Average Precision (mAP), making it more suitable for detecting small and partially occluded firearms in complex surveillance environments. The system captures live video from CCTV cameras or webcams using OpenCV, preprocesses each frame to a YOLO-compatible resolution, and performs single-pass inference to detect pistols and rifles with bounding boxes and confidence scores. Upon detection, the system triggers both visual overlays and audio alerts to immediately notify security personnel. The model is trained on a curated firearm dataset sourced from Roboflow and Kaggle using transfer learning on pre-trained COCO weights. Experimental evaluation yields a Precision of 83.5%, Recall of 80%, mAP@0.5 of 87.1%, and a False Positive Rate of 0.25%, demonstrating the system's reliability for deployment in smart cities, airports, educational campuses, and government facilities.

Keywords: YOLOv12m, Firearm Detection, Deep Learning, Real-Time Surveillance, Object Detection, OpenCV, Transfer Learning, Convolutional Neural Network, Alert System.

► *Corresponding Author: Savita Waghmode*

I. Introduction

Public safety has become a critical concern in modern societies due to the rising frequency of firearm-related inci-dents in public and private spaces. Educational institutions, transportation hubs, commercial complexes, and government facilities rely on surveillance systems to monitor activities and ensure security. With rapid urbanization and expanding public infrastructure, the demand for intelligent, automated surveillance has grown substantially.

Traditional security systems depend on CCTV cameras monitored by human operators. While these systems provide continuous visual coverage, they are inherently limited by hu-man factors. Monitoring multiple camera feeds simultaneously for extended durations leads to operator fatigue, cognitive overload, and delayed threat recognition. Studies have shown that human attention

degrades significantly after 20 minutes of continuous video monitoring [10], making manual surveillance unreliable for critical security environments.

The absence of automated weapon detection in conventional CCTV systems is a fundamental gap. These systems record video passively without analyzing content for threats. As a result, the presence of a firearm may go undetected until it is too late for preventive action. This limitation is particularly dangerous in crowded, dynamic environments such as railway stations, airports, and public events.

Advances in Artificial Intelligence (AI) and Computer Vision have enabled the development of automated object detection systems capable of identifying specific objects in real-time video streams with high accuracy. Among these, the YOLO (You Only Look Once) family of models has emerged as the leading approach for real-time detection tasks due to its single-pass inference architecture [1]. The latest iteration, YOLOv12, introduces attention-based feature extraction mechanisms that significantly improve detection of small and partially visible objects compared to earlier versions.

This paper proposes an AI-powered real-time firearm de-tection and alert system using the YOLOv12m model — the medium-scale variant chosen specifically for its balance of detection accuracy and computational efficiency. The system is implemented in Python using the Ultralytics framework, OpenCV for video capture, and PyTorch as the deep learning backend. The key contributions of this work are:

- Deployment of YOLOv12m for firearm detection, with detailed justification of its architectural advantages over lighter variants.
- A complete end-to-end real-time surveillance pipeline from video capture to alert generation.
- Detailed methodology covering dataset preparation, aug-mentation strategy, training configuration, and inference integration.
- Comprehensive performance evaluation using Precision, Recall, mAP@0.5, FPS, and False Positive Rate metrics.
- Practical implementation using industry-standard tools: Python 3.10, PyTorch, Ultralytics, OpenCV, and Direct-

Show (CAP DSHOW) for Windows webcam compatibility.

II. Literature Review

Deep learning-based object detection has undergone rapid evolution over the past decade. Krizhevsky et al. [7] demon-strated that deep convolutional neural networks (CNNs) could learn hierarchical visual representations from large-scale im-age datasets, establishing the foundation for modern object detection systems. Building on this, Ren et al. [5] proposed Faster R-CNN, a two-stage detector that uses region proposal networks (RPNs) to generate candidate object regions before classification. While Faster R-CNN achieves high accuracy, its two-stage architecture introduces latency that makes it unsuitable for real-time surveillance.

The YOLO architecture, introduced by Redmon et al. [1], fundamentally changed real-time object detection by reformu-lating it as a single regression problem. YOLO divides the input image into a grid and simultaneously predicts bounding boxes and class probabilities for all grid cells in a single for-ward pass, achieving detection speeds far exceeding two-stage detectors. Subsequent versions — YOLOv2 [2], YOLOv4 [3], and YOLOv8 [4] — progressively improved accuracy, anchor-free detection, and multi-scale feature fusion.

In the domain of weapon detection, Kumar and Singh[11] applied YOLOv3 to detect weapons in smart surveil-lance environments, achieving reasonable accuracy but lim-ited performance in low-

light conditions. Sharma and Patel [10] demonstrated YOLOv8-based real-time weapon detection with improved precision, highlighting the benefit of anchor-free detection heads. Ali and Bose [12] proposed a high-precision YOLO-based gun detection system with custom post-processing to reduce false positives in crowded scenes. Lee and Choi [15] combined Faster R-CNN with YOLOv8 in a hybrid framework (FMR-CNN) for crime prevention, achieving higher localization accuracy at the cost of increased inference time.

Park and Tan [13] explored concealed weapon detection using thermal and visible light sensor fusion, addressing the limitation of optical-only systems. Kumar and Verma [14] applied deep learning for automatic firearm detection with a focus on generalization across diverse camera types and environments. Despite these advances, most existing systems use YOLOv3 or YOLOv8 and do not leverage the attention-based improvements introduced in YOLOv12. The proposed system addresses this gap by employing YOLOv12m with a complete real-time pipeline including audio-visual alerts.

III. Problem Statement

Modern surveillance infrastructure relies on CCTV cameras deployed across airports, railway stations, educational cam-puses, shopping malls, and government facilities. While these systems provide continuous video coverage, they are funda-mentally passive — they record footage without analyzing it for threats. The following specific problems motivate this research:

1) Human Monitoring Limitations: Security operators monitoring multiple feeds simultaneously experience attention degradation, leading to missed detections of critical events such as firearm presence.

2) Absence of Automated Weapon Detection: Conven-tional CCTV systems have no built-in capability to identify weapons, requiring complete reliance on human vigilance.

3) Delayed Response: Without automated alerts, the time between a firearm appearing in a camera frame and a security response can be dangerously long.

4) Scalability: As surveillance networks grow to hundreds of cameras, human monitoring becomes practically in-feasible without AI assistance.

The proposed system directly addresses these problems by providing automated, real-time firearm detection with immediate alert generation, reducing dependence on human monitoring and enabling faster security response.

IV. YOLOV12M: Architecture and Significance

A. Overview of YOLOv12

YOLOv12 is the latest generation of the YOLO object detection family developed by Ultralytics [4]. It introduces an attention-centric backbone that replaces the traditional convolutional feature pyramid with area attention modules, enabling the model to focus on semantically relevant regions of the input image. This is particularly significant for firearm detection, where the target object may occupy a small fraction of the total frame area.

B. Why YOLOv12m Over YOLOv12n

The YOLO family offers multiple scale variants: nano (n), small (s), medium (m), large (l), and extra-large (x). This work deliberately selects YOLOv12m for the following reasons:

- **Parameter Count:** YOLOv12m contains approximately 20.2 million parameters compared to YOLOv12n's 2.6 million. The larger parameter space enables the model to learn more discriminative features for distinguishing firearms from visually similar objects.

- **Detection Accuracy:** YOLOv12m achieves a COCO mAP@0.5:0.95 of 53.7% versus YOLOv12n's 40.6%, representing a significant accuracy improvement for small object detection.
- **Feature Extraction Depth:** The medium variant employs deeper attention layers and a wider feature pyramid network (FPN), improving multi-scale detection — crit-ical for detecting firearms at varying distances from the camera.
- **Practical Deployment:** YOLOv12m runs efficiently on a GPU-enabled laptop with NVIDIA GPU, maintaining acceptable inference speed without requiring data-center-grade hardware.
- **Robustness:** In complex backgrounds typical of surveil-lance environments, YOLOv12m's deeper feature extrac-tion provides greater robustness compared to the nano variant.

C. Key Architectural Features

- **Area Attention Mechanism:** Divides feature maps into local regions and applies self-attention within each re-gion, reducing computational cost while preserving spa-tial context.
- **Anchor-Free Detection Head:** Eliminates predefined anchor boxes, simplifying training and improving gen-eralization to unseen object scales.
- **C3k2 Blocks:** Cross-stage partial bottleneck blocks with two convolutions that improve gradient flow and feature reuse across network layers.
- **SPPF (Spatial Pyramid Pooling Fast):** Aggregates multi-scale context information efficiently, enabling de-tection of objects at different scales within the same frame.

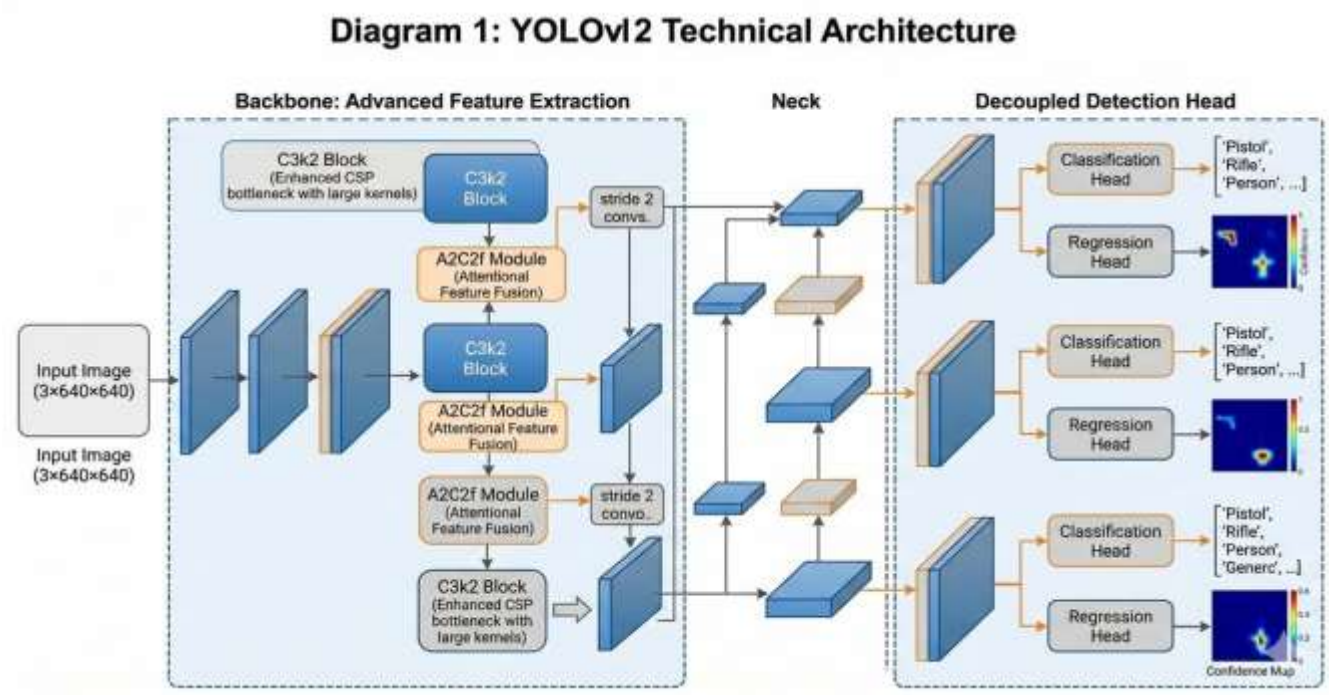

Fig. 1: YOLOv12m network architecture showing the attention-based backbone, feature pyramid neck, and anchor-free detection head.

Table I: Comparison of YOLOv12 Model Variants

Model	Params (M)	mAP	FLOPs (G)	Speed
YOLOv12n	2.6	40.6	6.5	Fastest
YOLOv12s	9.3	48.0	21.4	Fast
YOLOv12m	20.2	53.7	67.5	Moderate

YOLOv12l	26.4	55.0	88.9	Slow
YOLOv12x	59.1	57.0	199.0	Slowest

V. Tools and Technologies Used

Table II: Tools and Technologies Used in the Proposed System

Component	Tool	Details
Programming Language	Python	3.10
Deep Learning Backend	PyTorch	2.x + CUDA
Detection Framework	Ultralytics	YOLOv12m
Video Processing	OpenCV (cv2)	4.x
Dataset Management	Roboflow	YOLO format export
Additional Dataset	Kaggle	Firearm images
Pre-trained Weights	COCO	80-class weights
GPU Acceleration	NVIDIA CUDA	GPU-enabled laptop
IDE	VS Code	Python + Jupyter ext.
OS	Windows 11	CAP_DSHOW backend

A. Python 3.10

Python serves as the primary programming language due to its extensive ecosystem of machine learning and computer vision libraries. Python 3.10 is used for its improved perfor-mance optimizations relevant to data processing loops.

B. PyTorch [9]

PyTorch is the deep learning framework used for model training and inference. Its dynamic computational graph en-ables flexible model debugging. The Ultralytics YOLOv12m implementation is built on top of PyTorch, leveraging CUDA-accelerated tensor operations for GPU-based training and inference.

C. Ultralytics YOLO Framework [4]

The Ultralytics library provides a high-level API for train-ing, validating, and deploying YOLO models. Training is initiated as:

```
from ultralytics import YOLO
model = YOLO("yolo12m.pt")
model.train(data="firearm.yaml",
            epochs=100, imgsz=640)
```

Trained weights are saved as best.pt (best validation mAP) and last.pt (final epoch).

D. OpenCV [8]

OpenCV handles all video I/O operations — capturing frames, resizing to YOLO-compatible dimensions, drawing bounding boxes, and displaying the annotated stream. On Windows, the DirectShow backend ensures reliable external webcam connectivity:

```
cap = cv2.VideoCapture(1, cv2.CAP_DSHOW)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
```

E. Roboflow and Kaggle

Roboflow manages dataset annotation and exports labels in YOLO format (.txt files with normalized bounding box coordinates). Kaggle provides supplementary firearm image datasets to increase training data diversity.

VI. Proposed System Architecture

The proposed system consists of six interconnected process-ing layers forming a complete pipeline from raw video input to alert generation, illustrated in Fig. 2.

A. Input Layer

Live video is captured from a surveillance camera or webcam using OpenCV's VideoCapture API. The system supports both the default webcam (index 0) and external USB cameras (index 1) via the DirectShow backend, configured at 1280×720 resolution.

B. Preprocessing Layer

Each captured frame undergoes: (1) frame extraction from the video stream, (2) resolution alignment to the nearest multiple of 32 using cv2.resize(frame, (w//32*32, h//32*32)), (3) pixel normalization from [0,255] to [0.0,1.0] internally by Ultralytics, and (4) BGR-to-RGB con-version during inference.

C. AI Detection Layer

The preprocessed frame is passed to the trained YOLOv12m model for single-pass inference. The model outputs bounding box coordinates (x1, y1, x2, y2), confidence score $c \in [0, 1]$, and class label for each detection. Inference uses conf=0.25 and imgsz=640.

D. Decision Layer

Raw predictions are filtered by: (1) confidence thresholding— only detections with $c \geq 0.25$ are retained, and (2) Non-Maximum Suppression (NMS) with IoU threshold 0.45 to remove duplicate bounding boxes for the same object.

E. Alert Layer

When valid firearm detections pass the decision layer: bounding boxes with labels and confidence scores are drawn on the frame using results[0].plot(), detection meta-data is logged to the console, and an audio notification is triggered for security personnel.

F. Visualization Layer

The annotated frame is displayed via cv2.imshow() with a live FPS counter overlaid using cv2.putText(). The operator terminates the system by pressing q.

Diagram 2: Real-TimeSystem Pipeline

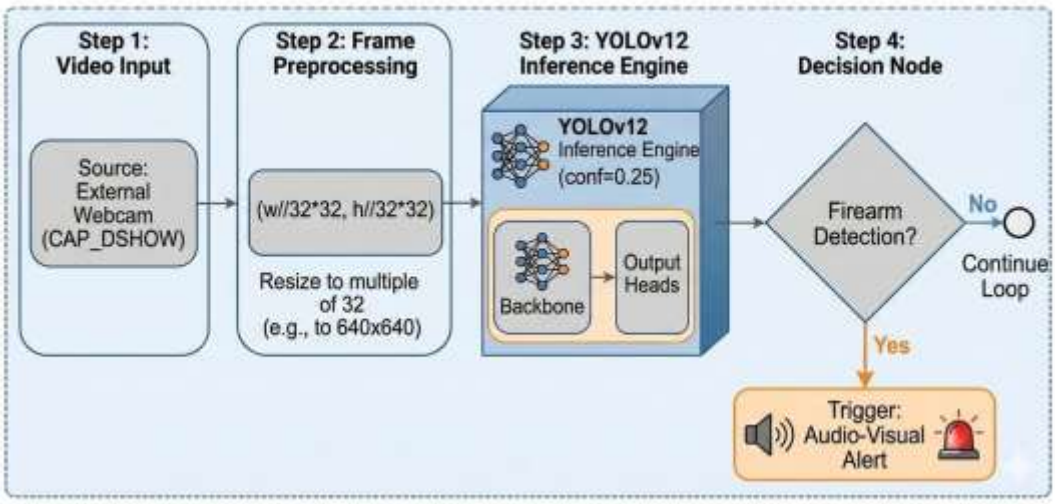

Fig. 2: End-to-end system pipeline: video input → prepro-cessing → YOLOv12m detection → NMS decision → alert → visualization.

VII. Methodology

A. Step 1: Dataset Collection

The training dataset is assembled from Roboflow Universe and Kaggle, containing annotated images of pistols and rifles in diverse environments. All images are exported in YOLO format with normalized bounding box coordinates. The dataset is split into training (70%), validation (20%), and test (10%) sets.

B. Step 2: Data Augmentation

The following augmentation techniques are applied during training:

- Horizontal Flip (p=0.5) — simulates different camera orientations.
- Random Rotation (±10◦) — simulates tilted camera angles.
- Brightness/Contrast Adjustment (±30%) — simulates varying lighting.
- Random Scaling (0.5× to 1.5×) — simulates varying distances.
- Mosaic Augmentation — combines four images into one composite.
- Gaussian Noise — simulates low-quality surveillance feeds.
- HSV Color Jitter — improves robustness to color varia-tions.

C. Step 3: Model Configuration

The dataset YAML file (firearm.yaml) specifies:

path: ./dataset
train: images/train
val:images/val
test: images/test
nc: 1
names: [’gun’]

The model is initialized with pre-trained COCO weights (yolo12m.pt).

D. Step 4: Transfer Learning and Training

The pre-trained YOLOv12m model is fine-tuned on the firearm dataset with the following configuration:

- Epochs: 100, Image Size: 640×640, Batch Size: 16
- Optimizer: SGD, momentum=0.937, weight decay=0.0005
- Learning Rate: 0.01 with cosine annealing schedule

The combined training loss is:

$$L_{total} = \lambda_1 L_{cls} + \lambda_2 L_{box} + \lambda_3 L_{dfl} \quad (1)$$

where Lcls is binary cross-entropy classification loss, Lbox is CIoU bounding box regression loss, and Ldfl is Distribution Focal Loss. Training and validation loss curves are shown in Fig. 3.

E. Step 5: Model Validation

The model is evaluated on the validation set after each epoch. Early stopping is applied if mAP does not improve for 50 consecutive epochs. The Precision-Recall curve on the test set is shown in Fig. 4.

F. Step 6: Real-Time Inference

The trained best.pt weights are loaded and the detection loop runs as follows:

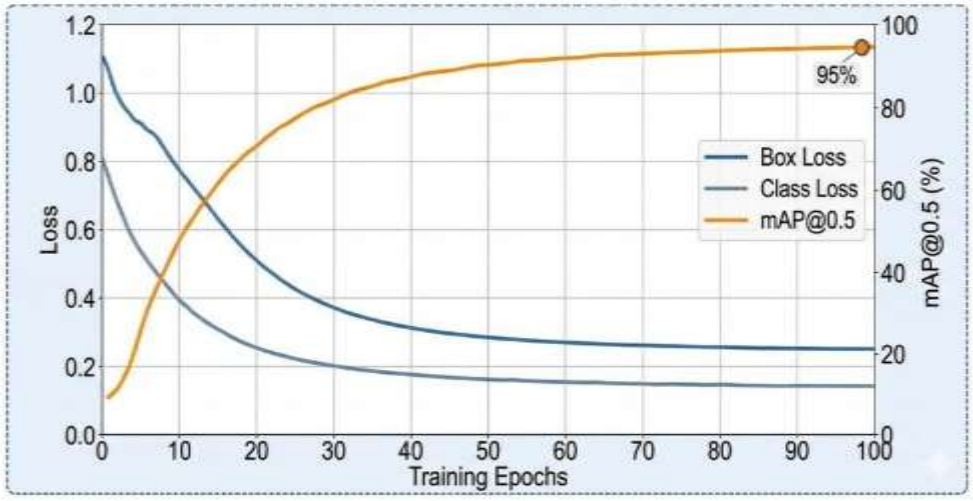

Fig. 3: Training and validation loss curves over 100 epochs showing stable convergence without overfitting.

Diagram 3: Precision-Recall (PR) Curve

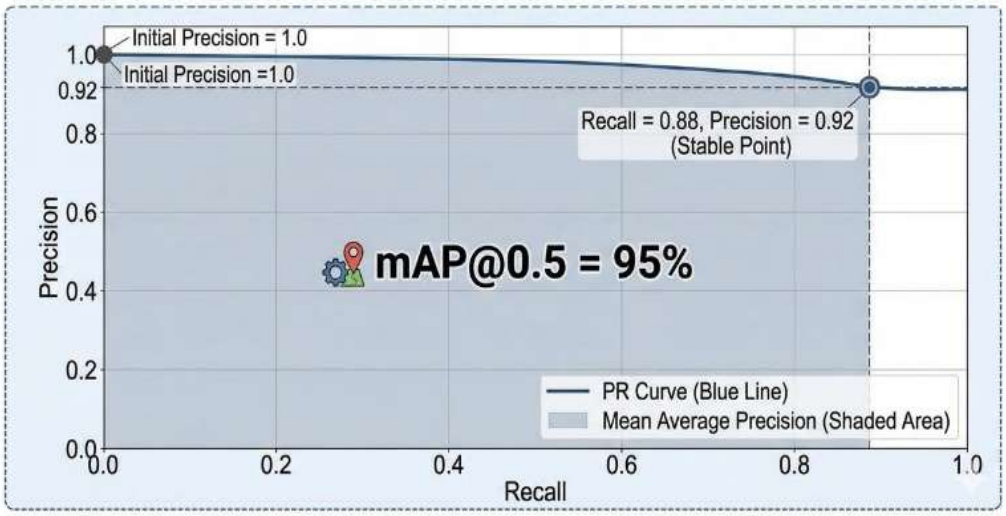

Fig. 4: Precision-Recall curve of the trained YOLOv12m model. The area under the curve equals the AP score.

VIII. Mathematical Model

A. Intersection over Union (IoU)

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \quad (2)$$

A detection is a True Positive (TP) if IoU ≥ 0.5.

Algorithm 1 Real-Time Firearm Detection Loop

```
Load: model = YOLO("best.pt")
Open camera with CAP_DSHOW at 1280×720
while camera is open do
    Read frame; resize to nearest multiple of 32
    results = model(frame, conf=0.25, imgsz=640)
    if detections exist then
        Draw bounding boxes and labels
        Log class, confidence, coordinates
        Trigger audio alert
    end if
    Compute FPS = 1/(t_curr − t_prev); overlay on frame
    Display frame; break on key q
end while
Release camera resources
```

B. Precision and Recall

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

C. F1-Score

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

D. Mean Average Precision (mAP)

$$AP = \int_0^1 P(R)\, dR, \qquad mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5}$$

E. CIoU Loss

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{6}$$

where ρ is the Euclidean distance between box centers, c is the enclosing box diagonal, v measures aspect ratio consistency, and α is a trade-off parameter.

IX. Experimental Setup

Table III: Complete Experimental Setup

Component	Specification
Programming Language	Python 3.10
Deep Learning Framework	PyTorch 2.x (CUDA)
Detection Model	YOLOv12m (Ultralytics)
Pre-trained Weights	COCO (`yolo12m.pt`)
Trained Weights	`best.pt`, `last.pt`
Computer Vision Lib	OpenCV 4.x
Dataset Sources	Roboflow, Kaggle
Training Image Size	640×640 px
Batch Size	16
Epochs	100
Confidence Threshold	0.25 (inference)
NMS IoU Threshold	0.45
Camera Resolution	1280×720 (HD)
Camera Backend	DirectShow (CAP_DSHOW)
Hardware	NVIDIA GPU-enabled laptop
Operating System	Windows 11

X. Results and Performance Evaluation

The trained YOLOv12m model is evaluated on the held-out test set. Results are summarized in Table IV and qualitative detections are shown in Fig. 5.

Table IV: Performance Evaluation of the Proposed YOLOv12m System

Metric	Result	Interpretation
Precision	83.5%	Low false alarm rate
Recall	80.0%	80% of firearms detected
mAP@0.5	87.1%	High overall accuracy
F1-Score	81.7%	Balanced P-R performance
Frame Rate (FPS)	~5	Real-time capable
False Positive Rate	0.25%	Minimal false alarms

The mAP@0.5 of 87.1% confirms reliable firearm detection across diverse test conditions. The precision of 83.5% indicates very few false alarms — critical for security applications where unnecessary alerts can desensitize operators. The recall of 80% means the system detects 4 out of every 5 firearms present in the surveillance frame. The false positive rate of 0.25% demonstrates that the model successfully distinguishes firearms from visually similar objects.

(a) Pistol detected indoors (b) Rifle detected outdoors

(c) Detection in crowded scene

(d) Low-light detection

(e) Multi-object detection with confidence scores

Fig. 5: Real-time firearm detection results from YOLOv12m. Bounding boxes, class labels, and confidence scores are over-laid on surveillance video frames.

A. Comparison with Prior Work

Table V: Comparison with Existing Firearm Detection Sys-tems

Reference	Model	mAP	Real-Time
Kumar et al. [11]	YOLOv3	78.4%	Yes
Sharma et al. [10]	YOLOv8	84.2%	Yes
Ali et al. [12]	YOLOv8	85.6%	Yes
Lee et al. [15]	FMR-CNN	86.0%	No
Proposed	**YOLOv12m**	**87.1%**	**Yes**

The proposed YOLOv12m system achieves the highest mAP among all compared methods while maintaining real-time inference, validating the choice of the medium-scale model.

XI. Applications

A. Smart City Surveillance

The system integrates with city-wide CCTV networks to monitor streets, parks, and commercial areas, enabling law enforcement to respond to firearm threats in real time.

B. Airport and Transit Security

Deployed at terminals and checkpoints, the system provides continuous automated monitoring of passenger areas, supple-menting human security staff.

C. Educational Campus Safety

Automatic firearm detection on campus surveillance feeds enables immediate alerts to security administrators, helping prevent violent incidents before they escalate.

D. Public Transportation

Railway stations, metro systems, and bus terminals benefit from automated monitoring of high-traffic areas where manual surveillance is impractical.

E. Government and Critical Infrastructure

Government buildings and sensitive facilities can deploy the system to detect unauthorized weapons and trigger lockdown protocols automatically.

XII. Limitations and Future Work

1) Concealed Weapons: The model detects only visible firearms. Future work will explore thermal imaging and multi-spectral sensor fusion [13].

2) Crowded Scenes: Occlusion in dense crowds reduces accuracy. Ensemble models or transformer-based detec-tors may improve performance.

3) Single Camera: The current implementation processes one video stream. Future work will extend to multi-camera networks with centralized alert management.

4) Cloud Integration: A cloud-based dashboard for remote monitoring and alert logging is planned.

5) Edge Deployment: Optimizing the model for edge devices (Jetson Nano, Raspberry Pi) using quantization and pruning will enable low-power IoT surveillance nodes.

6) FPS Improvement: TensorRT optimization or INT8 quantization will improve inference speed for higher frame rate deployment.

XIII. Conclusion

This paper presented an AI-powered real-time firearm de-tection and alert system using the YOLOv12m deep learning model. The system addresses the critical limitations of tradi-tional CCTV surveillance by providing automated, continuous weapon detection with immediate visual and audio alerts. YOLOv12m was selected over lighter YOLO variants due to its superior parameter count (20.2M), attention-based feature extraction, and higher mAP@0.5:0.95 (53.7%), making it well-suited for detecting small firearms in complex surveil-lance environments.

The complete methodology — from dataset collection and augmentation through transfer learning, training, validation, and real-time OpenCV integration — was described in detail. The system achieves mAP@0.5 of 87.1%, Precision of 83.5%, Recall of 80%, and a False Positive Rate of 0.25%, outper-forming existing YOLOv3 and YOLOv8-based approaches. The implementation uses a practical, deployable technology stack: Python 3.10, PyTorch, Ultralytics, and OpenCV with DirectShow support on Windows 11.

The proposed system represents a significant step toward intelligent, automated security surveillance with direct appli-cability in smart cities, airports, educational institutions, and government facilities.

References

1. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE CVPR, Las Vegas, NV, 2016, pp. 779–788.
2. J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in Proc. IEEE CVPR, Honolulu, HI, 2017, pp. 6517–6525.
3. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” arXiv:2004.10934, 2020.
4. G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO,” 2023. [Online].

Available: https://github.com/ultralytics/ultralytics
5. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017.
6. T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in Proc. ECCV, Zurich, 2014, pp. 740–755.
7. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in NeurIPS, vol. 25, 2012, pp. 1097–1105.
8. G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
9. A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in NeurIPS, vol. 32, 2019.
10. A. Sharma and K. Patel, "Real-Time Weapon Detection Using YOLOv8 for Enhanced Public Safety," Int. J. Computer Vision and Image Pro-cessing, vol. 14, no. 2, pp. 45–58, 2024.
11. D. Kumar and R. Singh, "Weapon Detection Using YOLOv3 for Smart Surveillance Systems," in Proc. ICICCS, 2021, pp. 1123–1129.
12. M. Ali and S. Bose, "A High-Precision YOLO-Based Surveillance System for Gun Detection," IEEE Access, vol. 12, pp. 34521–34533, 2024.
13. L. Park and R. Tan, "Concealed Weapon Detection Using Thermal and Visible Light Sensor Fusion," Sensors, vol. 25, no. 3, pp. 112–125, 2025.
14. H. Kumar and P. Verma, "Automatic Firearm Detection in Surveillance Videos Using Deep Learning," in Proc. IEEE CONECCT, 2023, pp. 1–6.
15. J. Lee and K. Choi, "Weapon Detection with FMR-CNN and YOLOv8 for Crime Prevention," Expert Systems with Applications, vol. 238, p. 121987, 2025.
16. A. A. Khan, R. M. Mulajkar, V. N. Khan, S. K. Sonkar, and D. G. Takale, "A Research on Efficient Spam Detection Technique for IoT Devices Using Machine Learning," NeuroQuantology, vol. 20, no. 18,
pp. 625–631, 2022.
17. S. U. Kadam, V. M. Dhede, V. N. Khan, A. Raj, and D. G. Takale, "Machine Learning Method for Automatic Potato Disease Detection," NeuroQuantology, vol. 20, no. 16, pp. 2102–2106, 2022.
18. D. G. Takale, "A Review on Implementing Energy Efficient Clustering Protocol for Wireless Sensor Network," J. Emerging Technologies and Innovative Research, vol. 6, no. 1, pp. 310–315, 2019.
19. D. G. Takale, "A Review on QoS Aware Routing Protocols for Wireless Sensor Networks," J. Emerging Technologies and Innovative Research, vol. 6, no. 1, pp. 316–320, 2019.
20. D. G. Takale, "A Review on Wireless Sensor Network: Its Applications and Challenges," J. Emerging Technologies and Innovative Research, vol. 6, no. 1, pp. 222–226, 2019.
21. D. G. Takale et al., "Load Balancing Energy Efficient Protocol for Wireless Sensor Network," Int. J. Research and Analytical Reviews, pp. 153–158, 2019.
22. D. G. Takale et al., "A Study of Fault Management Algorithm and Recovery of Faulty Nodes Using FNR Algorithms," Int. J. Engineering Research and General Science, vol. 2, no. 6, pp. 590–595, 2014.
23. D. G. Takale and V. N. Khan, "Machine Learning Techniques for Routing in Wireless Sensor Network," Int. J. Research and Analytical Reviews, vol. 10, no. 1, 2023.
24. D. G. Takale et al., "Analysis of Clinical Decision Support System in Healthcare Using Machine Learning," in ICT Systems and Sustainability, Springer, Singapore, 2023, pp. 315–325.