International Journal of Safety and Security Engineering
Vol. 16, No. 6, June, 2026, pp. 1309-1321
Journal homepage: http://iieta.org/journals/ijsse

Real Time Multi-Class Weapon and Physical Aggression Event Detection Using YOLO26 in Surveillance Videos

Mohammed Raid Salman*, Kawther Thabt Saleh

Department of Computer Science, Mustansiriyah University, Baghdad 10052, Iraq

Corresponding Author Email: mohammedsalman99@uomustansiriyah.edu.iq

https://doi.org/10.18280/ijsse.160613

Received: 19 April 2026
Revised: 1 June 2026
Accepted: 11 June 2026
Available online: 30 June 2026

Keywords:
deep learning, YOLO26, real-time surveillance, weapon detection, classification, augmentation

ABSTRACT

In recent years, the need for intelligent surveillance systems to detect threats has been rising due to increasing criminal and violent activities. Traditional methods of monitoring, as known, depend entirely on human observation, which becomes inefficient when dealing with multiple screens and long monitoring periods, making early detection difficult in critical situations. This study proposes a real-time weapon detection and classification system based on the latest generation of YOLO family (YOLO26). Training and evaluation were done on two datasets: the first is a publicly available dataset used in previous studies, and a large-scale dataset constructed in this work containing multiple categories. The aim is to achieve an optimal balance between computational efficiency and accuracy of detection, making it suitable for many real-time tasks. Experimental results of YOLO26-M (Medium) achieved strong performance on both datasets with a precision of 0.954, a recall of 0.913, an F1 score of 0.933, a mean average precision mAP@50 of 0.955, and mAP @50-95 of 0.840 while maintaining low inference latency. The proposed model shows robustness in detection, also handling complex conditions that could occur in surveillance. The results highlight the potential the framework has for surveillance applications, while additional deployment-specific evaluation remains an important direction for future work. Overall, this work contributes toward improving intelligent systems associated with security by providing a scalable and efficient approach for real-time weapon detection and public safety enhancement.

1. INTRODUCTION

The rise in violence and homicide rates has been a challenge in the modern world because of the uncontrolled access to weaponry in many countries, leading to serious threats to public safety and security in general, with other consequences like psychological and economic losses. It's known that guns and other weapons are the most common form of crime incidence and violent acts [1, 2]. And as it's known, the traditional monitoring process is not very effective, especially when many monitor screens require concentration from the operator, where the personnel in charge can lose focus after short periods, leading to the possibility of missing critical events [3, 4]. All the mentioned points lead to the conclusion that human monitoring alone is not effective or accurate in the threat/danger detection process, which emphasizes the vital need for effective solutions and measures.

The recent advancement in technology and artificial intelligence (AI) specifically deep learning and computer vision, has enabled the idea of weapon detection [5]. The conventional neural networks (CNNs) proved to be very effective when it comes to object detection tasks, and when speaking about strong models in object detection many other modern frameworks stand out, like the YOLO family which gained its reputation due to its effectiveness and ability in real time with the needed balance between accuracy and speed [6, 7]. The detection task in deep learning models is highly related to many factors such as object size, lighting conditions, or occlusion, making it very difficult, especially if the images captured have complex backgrounds [8, 9]. In addition, the performance of a deep learning model depends on the dataset itself and its quality. The existing datasets are limited in size and diversity and are not accurate in representing the survival scenarios, making the model less generalizable [10, 11].

To address the limitations above, recent studies have focused on both model architecture and dataset characteristics. Newer YOLO based models enhanced the original architectures in the early versions, leading to improvement in the detection process and real-time capability [7, 8].

The current study deals with the latest progress in object detection through YOLO versions. The study contributes to enhancing the public security field in the following ways:

(1) Comparing the new YOLO26-M (Medium) against YOLOv11-M, training both versions on a dataset from a previous study.
(2) Constructing a dataset of 10 classes from different weapons categories and violence, containing 50k images (along with the annotations) in total.
(3) Building a model from training the same version

(YOLO26) on the proposed dataset capable of detecting and classifying potential threats in real time.

The sections discussed in this study are structured as follows: the literature view is summarized in Section 2, information about YOLO26 is provided in Section 3, and the methodology, containing the datasets related descriptions mentioned, the model, the environment used, and the training steps all presented in Section 4. While Section 5 discusses the results for both datasets using appropriate evaluation and consideration taken for further deployment.

2. RELATED WORK

When mentioning object detection, the origins must be considered, and the early approaches for detection all relied on machine learning (ML), which was the foundational framework with its traditional techniques and feature engineering. For example, one study employed Gradient Orientation and Laplacian Magnitude-based Histogram (GLH) descriptors that showed features that, when designed carefully, can achieve high accuracy of nearly 95% with strong metric values [12]. Despite the promising results, the traditional ML approaches remain constrained and suffer from some limitations, such as manual feature engineering, making them struggle to generalize in complex environments, featuring reduced scalability and robustness.

On the other hand, deep learning advancement has opened the way for many alternatives, such as CNNs for object detection and classification tasks. The CNN-based methods work by hierarchically learning feature representation automatically from the data, which makes them superior when compared with ML methods [13]. The faster R-CNN, which is considered a two-stage detector in some studies, showed improved accuracy due to the regional proposal mechanism and can outperform single-stage detectors like SSD, but the trade-off is inference speed [3]. Not only weapons, but also violence is concentrated on by some researchers.

The AIRTLab database has been used to apply three proposed models focusing on violence detection, counting on spatial-temporal features extracted from video streams, where recognizing aggressive behaviors is possible, although false positives remain a difficult challenge due to the similarity with non-violent actions like hugs and friendly gestures [14].

However, these systems faced many challenges related to object size and surrounding surveillance environments. A new dataset was developed for application to the proposed Faster R-CNN model. The model demonstrated superiority over the SSD model in weapon detection but showed moderate performance with mean Average Precision (mAP) values around 0.662, which indicated that the small or occluded objects were difficult to detect [15]. The MARIE (Mechanism for Realtime Identification of Firearms) model, based on the original dataset of the University of Granada and the SSD framework, integrated MobileNetV2 and InceptionV2 architectures, resulting in a system with high speed, achieving accuracies of 81.65% and 80.91%, respectively [16]. In addition, the complexity in computations and the latency of CNN architectures like the VGG-16 and Inception-based models made these models not preferred or suitable for real-time tasks.

A dataset was collected from various database sources such as Kaggle, IMFDB, UGR, and LinkSprite [17]. Some of the limitations are related directly to the low-resolution input images or complex backgrounds, occlusion, small size of weapons, or point of view from which the image is captured, even when lighting conditions are considered, which significantly degrade detection performance in real-world surveillance environments.

To overcome the challenges of real-time object detection, the YOLO family of detectors was adopted, becoming the dominant model for detection. A comparative analysis of YOLOv3 and YOLOv4 was conducted to determine which is superior [2]. A dataset of images of weapons was compiled from Google Images and other sources. The YOLOv4 has outperformed YOLOv3 in terms of performance, with mAP values of approximately 84.85% compared to v3's 77.30%.

A dataset was collected through photographing weapons, capturing CCTV footage from YouTube, Internet Movie Firearms Database (IMFDB) imfdb.org, and utilizing data from the University of Granada and other sources [18]. Several methods have been proposed, such as Faster-RCNN Inception-ResnetV2 (FRIRv2), Inception-V3, Inception-ResnetV2, SSDMobileNetV1, VGG16, YOLOv3, and YOLOv4. The results showed that the best method was using YOLOv4, which gives a mean average precision of 91.73%, with an F1-score of 91%, and it outperformed the other algorithms in terms of results. YOLO models are constantly evolving.

A system based on Scaled-YOLOv4 was developed, optimizing FPS and making it suitable for Weapon detection in all devices [19]. The Scaled-YOLOv4 model achieves an mAP value of about 92.1%, while a system that uses YOLOv8 and a dataset collected from different sources in addition to images generated by GANs enhanced further with augmentations achieves accuracies exceeding 92% [6]. A system using the YOLOv8 model for detecting handguns, shotguns, and other weapons to send an alert about an armed robbery demonstrated very strong performance in identifying anomalies with an accuracy rate of 87.50% [20].

Recent studies have demonstrated that YOLO based architictures maintain a strong balance between detection accuracy and computational efficiency, supporting their adoption in real-time surveillance and security applications [17].

Despite the mentioned improvements, some important challenges remain. Even with modern YOLO-based models, detection performance often drops below 95% in realistic conditions due to factors such as small object size, occlusion, poor lighting, and visual similarity between weapons and non-weapon objects [21]. A systematic review of AI-based weapon detection systems between 2016 and 2025 pointed out that the most recent approaches rely on deep learning, like Faster R-CNN and YOLO, achieving highly promising results and precision values up to 99.5%; however, the study also points to the challenges related to dataset inconsistency and lack of large-scale standard datasets and even standard evaluation [11], justifying the need for more robust and flexible solutions. Table 1 below demonstrates a systematic comparison between the most important of the previously mentioned studies.

The existing studies either use small datasets, limited weapon classes, or older YOLO versions. This motivates the evaluation of YOLO26 on both the benchmark and diverse large-scale datasets to find solutions to the highlighted limitations in this paper.

Table 1. Systematic comparison of previous studies

Ref.	Year	Model	Dataset	Reported Performance	Limitations
Hashmi et al. [2]	2021	YOLOv3 and YOLOv4	Images were collected from movies, Google, and CCTV	77.30% to 84.85%	Diversity of categories and size of the dataset
Mangrolia and Sheth [12]	2021	GLH, HOG with SVM and NN	Video captured from the CCTV camera	30.32 to 96.66%	In the case of active backgrounds, the effectiveness decreases. Despite its robustness, GLH produces complex features that cannot be linearly classified
Sernani et al. [14]	2021	3D CNNs and ConvLSTM	AIRTLab dataset	95.20 to 98.16%	too many false positives when evaluating short sequences from long videos
Mane [15]	2024	Faster RCNN, SSD	Image from the internet	mAP@50 = 0.837	Lower performance compared to R-CNN and sensitivity to small object detection
Abi-Nader et al. [16]	2025	SSD with MobileNetV2, SSD with InceptionV2	Dataset compiled by the University of Granada	74.87% to 75.62%	Limited precision for small objects also produces false positives
Pravesh and Sahana [17]	2025	enhancement pipeline with a state-of-the-art YOLOv11-based object detector	Collected from platforms like Link sprite, VBS3, Kaggle, UGR, IMFDB	95.78%	Performance depends on image enhancement preprocessing and dark scene simulation
Bhatti et al. [18]	2021	Use VGG, Inceptionv3, InceptionResNetv2, SSD Mobile Net, Faster RCNN Inception ResNetv2, YOLOv3 and YOLOv4	Collected from the internet, IMFDB, YouTube CCTV videos, GitHub repositories, the University of Granada	mAP = 91.73%, F1 = 91%	Have a false positive and negative values
Ahmed et al. [19]	2022	Scaled-YOLOv4 with CSPDarknet53	Collected from real home security cameras and CCTV camera	mAP = 92.1%	Requires GPU/TensorRT optimization
Reyes and Cruz [20]	2025	YOLOv8	Collected from ARMAS weapon dataset, and internet movie firearm database imfdb.org	87.50%	Dataset overfitting and false negatives

Note: Gradient Orientation and Laplacian Magnitude-based Histogram = GLH; conventional neural networks = CNNs.

3. YOLO26

Transfer learning works by letting the model reuse previously learned features; thus, it has significant performance in tasks related to computer vision. It can also be applied successfully in multi-weapon detection systems like audio-based firearm classification [22]. Focusing on real-world deployment, the YOLO models (you only look once) family proved worthy and efficient due to their high detection and fast speed. YOLO models utilize the architecture of single-stage detection, and the processing of the entire image is done as one forward pass, resulting in faster speeds than two-stage detection systems [7].

The first proposed YOLO frame work proposed by Joseph Redmon and colleagues in 2016 presented a new approach in object detection, unlike the traditional R-CNN and Faster R-CNN, which separated region proposal from classification. YOLO formulated detection as a single regression problem. By directly predicting bounding boxes and class probabilities in one forward pass through a convolutional neural network (CNN), it achieves competitive accuracy and real-time speed. Through the following years, many versions were presented, each with different architectures and suitable improvements for the specific task.

In this study the latest version of YOLO which is YOLO26, is selected as the core model for the intended work. The YOLO26 implementation used in this sudy was based on the Ultrealytics framework and the experimental configuration, software version, model size and training parameters along with the evaluation settings are reported in section 4.2 to support reproducibility. as its previous versions is also designed to deliver the needed performance and efficiency, with the capability of handling various computer vision tasks such as object detection, classification, segmentation [7]. Like the previous versions, this one comes with multiple variants, each of which can handle different tasks and deployment requirements, starting from the lightest to the heaviest in architecture nano (n), small (s), medium (m), large (l) and extra-large (x). Compared to earlier models, this one has some architectural improvements aiming to enhance real-time inference and edge deployment. This means simplified design and reduced computational complexity, all to enable as fast an inference time as possible on both CPUs and GPUs [23].

From a theoretical perspective, the proposed dataset2 contains substantial variations in object scale, illumination conditions, occlusion and visual similarity between categories, along with the complex backgrounds. Such characteristics require a capable detector that can learn robust multi-scale feature representations while maintaining efficient localization. YOLO26 is particularly suitable for this scenario because its architecture is designed to preserve detailed spatial information while simultaneously extracting high-level semantic features.

This combination is important for distinguishing visually similar categories, such as a knife and a cutter. As well as for detecting small or partially occluded objects encountered commonly in surveillance imagery.

Details about the architecture are available in the original YOLO26 publication [7], illustrating COCO mAP (50-95) versus latency performance. The more recent the model is, the

higher its associated mAP, indicating the improvements over generations as well as competitive real-time detectors (PP-YOLOE+, DAMO-YOLO, and RTMDet). On the end-to-end latency axis, YOLO26 is compared with YOLOv10 and RT-DETR variants, illustrating its advantage in overall pipeline efficiency [7].

In this work, the YOLO26-M variant is selected for training and evaluation, based on the balance it provides between detection performance and computational efficiency, especially with a large-scale dataset, which makes it the most suitable for real-time surveillance, where both speed and accuracy are crucial.

4. PROPOSED METHODOLOGY

The proposed methodology for creating and developing a real-time weapon and violence detection system is explained in this section. Illustrated in Figure 1.

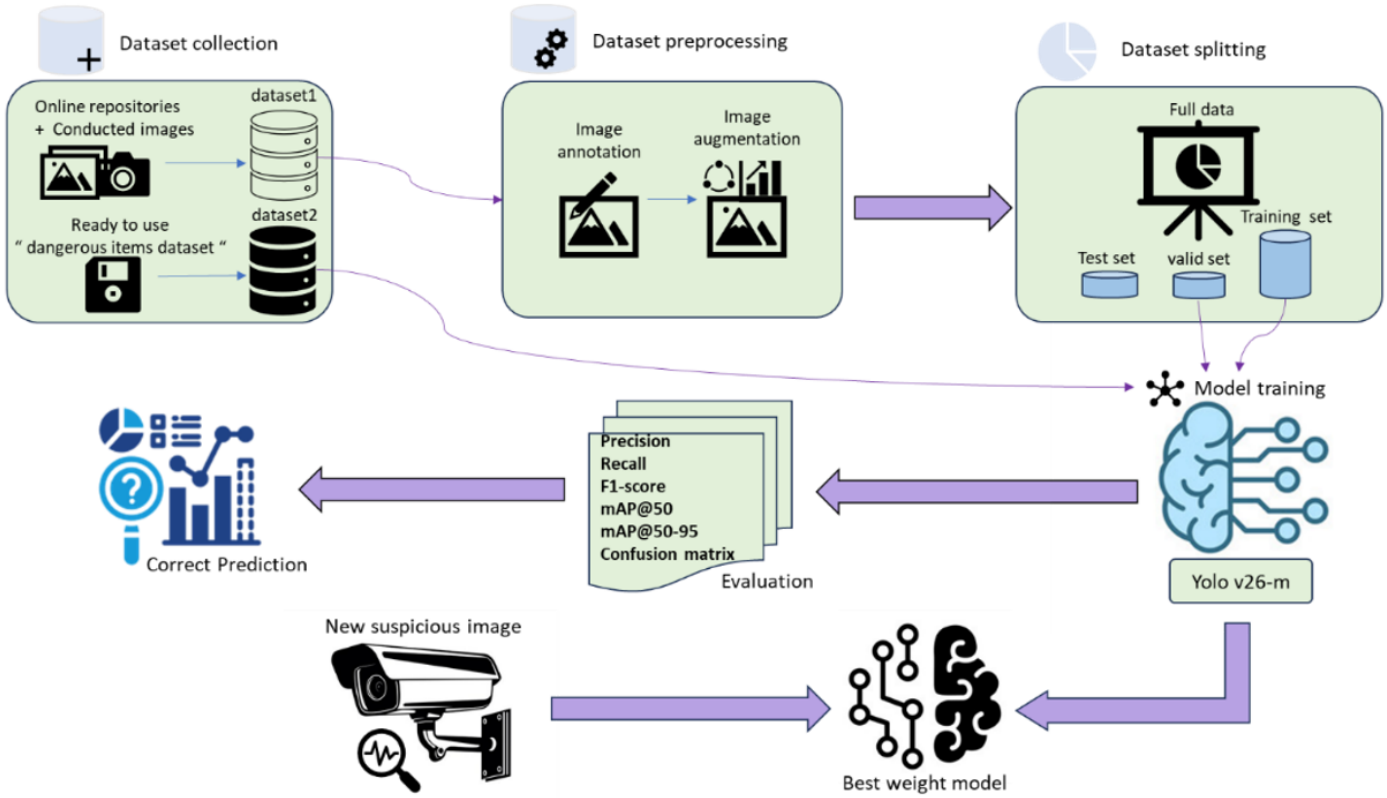

Figure 1. Proposed workflow

4.1 Datasets

The dataset used in this study was constructed and utilized following ethical data collection practices. Due to the lack of a standardized large-scale benchmark dataset for weapon detection and classification, the importance of constructing domain-specific datasets has also been highlighted in other computer vision applications, such as gesture recognition systems, where tailored datasets significantly improve model performance and real-world applicability [24].

This work employs two datasets. The first dataset is adopted from existing literature for benchmarking purposes, while the second dataset is newly constructed to address the limitations of existing datasets in terms of scale, diversity, and real-world representation.

4.1.1 Dataset description

As we mentioned, the study is using 2 datasets with different classes and sizes.

Dataset1 (dangerous items dataset) is the first dataset, adopted from a previous study, and consists of five classes: knife, machete, baseball bat, handgun, and rifle [21]. This dataset is used to evaluate and compare the performance of the proposed YOLO26 model against previously reported results.

The complete dataset consists of 8805 images created by merging self-collected images and public datasets and is split into 70% training (5934 images), while validation and test splits are both 15% (1272 images) each. The number of detected items they contain is shown in Table 2.

Table 2. Dataset1 classes and images

Classes	Train (70%)	Valid (15%)	Test (15%)	Total
Knife	1232	239	237	1708
Machete	1237	265	291	1793
Bat	1246	274	276	1796
Gun	1244	260	251	1755
Rifle	1219	261	273	1753
All	6178	1299	1328	8805

Dataset2 (proposed dataset) is the second dataset that represents the primary contribution of this study. It is a large-scale dataset consisting of 10 classes, designed to reflect realistic surveillance scenarios. The selected classes include veracity of weapons to ensure a system that can be deployed in public places like schools and hospitals and government institutions the classes depicted to show real life scenarios starting from close range violence weapons like knife and

cutter to the possible weapons an individual can carry, the classes are (knife, cutter, scissor, axe, hammer, handgun, assault rifle, shotgun, rocket launcher and violence) each class collected from previous researches that used publicly available datasets merged with images from online repositories and internet sources so each class is a mix of specific number of images chosen from many sources and datasets ensuring that the images for that class is covering real life scenarios like fog, dark and blurry images.

Particular attention was given to classes that suffer from underrepresentation in existing datasets, especially the hammer and cutter categories. For these classes, additional images were manually collected using the POCO F4 mobile camera under different viewpoints, backgrounds, and lighting conditions at different times of the day in multiple locations to better simulate realistic surveillance scenarios. These images were treated as the same dataset classes with the same annotation and quality control procedures as the remaining classes.

The selected classes were included in order to reflect prohibited weapons and ambiguous hand-held objects that may become threat-related depending on context. Some of the classes, like cutter hammer and scissor, may appear as ordinary tools; therefore, their detection should not be interpreted as a threat. Instead, the proposed system is designed to support surveillance operators by identifying any potentially relevant objects for further human verification. This distinction is important because real security decisions require contextual assessment, including location behavior, object handling, and operator confirmation.

The objective of dataset2 is not to classify objects as threats automatically, but rather to identify potentially relevant objects that may require further inspection within a surveillance workflow. Consequently, the selected categories include both explicit weapons and common handheld tools. While some of these objects may appear in legitimate contexts, they are also frequently associated with security incidents depending on the surrounding circumstances; therefore, the proposed system should be interpreted as a decision support mechanism that highlights suspicious objects for human review rather than making autonomous security decisions. Contextual information, object handling, location, and operator verification remain essential components of any practical deployment.

Unlike the weapon categories corresponding to physical objects, the violence class, which is considered in this work as a physical aggression event, represents visually observable violent interactions between individuals. The bounding boxes were annotated around the primary region where the aggressive interactions occurred. The reason for including this class is not to perform temporal action recognition but rather to identify visually evident threat-related events that may require immediate attention in surveillance environments, especially when the proposed framework is aiming at public places, including schools and hospitals, where such events need attention. Where physical aggression may occur and require rapid situational awareness, similar approaches have been explored in violence detection research, where aggressive behavior is treated as a security-relevant visual category for automated monitoring. The final class selection was designed to represent a broad spectrum of surveillance-relevant threats while maintaining diversity in object appearance, scale, and contextual complexity.

To improve dataset transparency, the source composition of dataset2 is summarized in Table 3. The dataset was constructed from multiple repositories, existing research datasets, and online sources, in addition to manually captured images for poor presentation classes. Duplicate samples were visually inspected and removed where possible before final splitting.

Unlike many existing weapon detection datasets that focus on a limited number of firearm categories or contain only a few thousand images, dataset2 was designed to represent a wider range of surveillance-relevant threat categories under diverse visual conditions. The inclusion of multiple weapon types, visually similar hand-held objects, and a dedicated physical aggression class under the name violence class increases the complexity of the detection task and enables a more comprehensive evaluation of model robustness. These characteristics provide a practical benchmark for studying the trade-off between detection accuracy, localization quality and computational efficiency in real-world scenarios.

Following duplicate filtering, quality-control inspection, class balancing and final dataset curation with the needed augmentations, in total, the dataset contains 10 classes with 50000 images and their associated labels, split as 80% for training, 10% for validation, and 10% for testing. The classes are described further in Table 4.

Table 3. Composition of dataset2 before preprocessing and augmentation

Category	Source	Selected Images
Knife	SOHAS weapon detection dataset	2786
	Robflow, Kaggle, internet sources	3800
Handgun	SOHAS weapon detection dataset	696
	Pistol computer vision dataset	698
	Armas dataset	593
	Robflow, Kaggle, and internet sources	7169
Assault rifle	Robflow, public repositories	9343
shotgun	Robflow, diverse internet sources	4443
Rocket launcher	Kaggle, internet repositories and Robflow	4966
Hammer	Self-constructed images, Roboflow, online repositories	3828
Cutter	Self-constructed images, online repositories, Robflow, Kaggle	3279
Axe	Internet sources, Robflow universe	4103
Violence	Internet sources, Robflow universe	5588
Scissor	Internet sources, Robflow universe, Kaggle	3190

Table 4. Dataset2 classes after preprocessing

Classes	Train (80%)	Valid (10%)	Test (10%)	Total
Knife	4000	500	500	5000
Cutter	4000	500	500	5000
Scissor	4000	500	500	5000
Axe	4000	500	500	5000
Hammer	4000	500	500	5000
Handgun	4000	500	500	5000
Assault rifle	4000	500	500	5000
Shotgun	4000	500	500	5000
Rocket launcher	4000	500	500	5000
Violence	4000	500	500	5000
All	40,000	5000	5000	50 k

4.1.2 Data preprocessing and augmentation

Dataset1 was used as provided by the original study, where standard augmentation techniques such as rotation, blurring, and brightness/contrast adjustments were applied to improve generalization.

Dataset2, most of the classes were collected and merged from different users from online sources each with its own preprocessing operations, and many have their own images uploaded at 640 × 640 image size. Some of the downloaded files were found to be processed and augmented with the needed standard operations (safe augmentations) like blur, noise, rotate, brightness and darkness.

Some classes needed augmentations to simulate real world footages that could be taken at different times of the day and in different weather situations. Here are the augmentations found for some of the images for the classes and some of them were applied to ensure covering all the cases and conditions of real-life demonstration and capturing footage from the surveillance cameras, all shown in Table 5.

CW / CCW means Clockwise / Counter-clockwise, and px refers to pixels. ROI (Region of Interest) for shadows restricts augmentation to the lower ~65% of the image, simulating realistic lighting conditions. Brightness/Darkness scaling values are multiplicative factors.

It should be noted that the dataset was constructed from multiple public sources, online repositories, and preexisting augmented samples uploaded by researchers to online repositories like Kaggle and RobFlow Universe. So, dataset splitting was performed before any additional augmentation operations were applied in this study. Augmented images generated during the present work were restricted to their corresponding split and were not transferred between training, validation or testing subsets. And for the samples obtained from public repositories that may already have undergone preprocessing or augmentation by their original contributors, duplicate inspection was conducted before final inclusion. Although complete elimination of visually similar images cannot be guaranteed when aggregating data from multiple public sources, these procedures were adopted to minimize potential overlap and preserve the independence of the evaluation subsets.

Table 5. Augmentations and values

Classes	Augmentation	Value Range
Handgun	Random Rotation	±15°
	Random Shear	±10°
	90° Rotation	None / CW / CCW / Upside-down
	Gaussian Blur	0–0.8 px
	Salt & Pepper Noise	0.5% pixels
	Horizontal Flip	50% probability
Violence	Random Rotation	±15°
	90° Rotation	None / CW / CCW / Upside-down
	Gaussian Blur	0–2.5 px
	Salt & Pepper Noise	1.8% pixels
Axe	Gaussian Blur	10–20 px
	Gaussian Noise	0.05–0.15
	Scale-out (Zoom Out)	0.75–0.95
	Darkness Scale	1.45–1.65
Assault Rifle	Horizontal Flip	100% probability
	Brightness Scale	0.55–0.65
Shotgun	90° Rotation	None / CW / CCW / Upside-down
	Horizontal Flip	Applied
	Brightness Scale	0.55–0.65
	Darkness Scale	1.45–1.65
Rocket Launcher	Horizontal Flip	50% probability
	Random Rotation	±15°
	Gaussian Blur	5–11 px
	Gaussian Noise	0.04–0.10
Hammer & Scissor	Brightness Adjustment	+0.10–+0.22
	Darkness Adjustment	−0.12–−0.28
	Cast Shadows	1–2 shadows, dimension = 5, ROI = (0.0, 0.35 → 1.0, 1.0)
	Random Rotation	−6° to +6°

4.1.3 Annotation and architecture

Most of the collected datasets were already available in YOLO format, while others were converted using custom Python scripts. Additionally, manually collected images (e.g., hammer and cutter classes) were annotated using the Roboflow platform, ensuring accurate bounding box labelling and class assignment. The final dataset follows the standard YOLO structure.

final Main folder :
-train
*Images (40000)
*labels
-Val
*Images (5000)
*labels
-test
*Images (5000)
*labels
-yaml.txt

4.2 Training process

The YOLO26-M version was chosen based on its balance between detection performance and computational efficiency. This version was applied to two datasets in the training process

to compare the performance and the model's generalization. The training process was done using Python scripts on Google Colab environment specifically the pro + subscription to unlock the new NVIDIA H100 GPU which is known for high computational capabilities reducing the time the training process can take for large datasets.

4.2.1 Dataset1 (benchmark dataset)

The first dataset was used to make a comparison with a previous study, and it has been proven that the YOLO26-M variant outperformed the YOLOv11-M. The dataset followed a direct structure to the YOLO standard. Executed the training process by using the following parameters [size of image: 960, Batch size: Auto (batch = -1, determined by GPU dynamically), Epochs: 200 (Equal to the referenced study), Device: GPU (device = 0), Workers: 8, Early stopping: Enabled]. The best performance model was achieved at epoch 183, where the training was completed without triggering early stopping.

4.2.2 Dataset2 (proposed dataset)

The 50,000 images were used here, considered a large-scale dataset and the same training track was followed, with minor variations to accommodate the size and complexity of the dataset. The session in the Colab environment is time-limited, and therefore, a training resume strategy was used. This allowed training to continue from the last point of interruption in case of a connection loss, thus ensuring the training process was completed fully.

To improve reproducibility, the implementation and training configuration used for dataset2. The experiments were conducted using Ultralytics version 8.4.14 with Python version 3.12.12 and PyTorch 2.9.0+u128, and CUDA acceleration on NVIDIA H100 80GB HBM3 GPU. The YOLO26-M model used in this study contained 132 fused layers, 20,357,162 parameters, and 67.9 GFLOPs. The training was performed for 300 epochs using an input size of 960×960, batch size of 64, patience of 50, cosine learning rate scheduling, AMP enabled, and deterministic training with seed 0, and additional parameters are : (optimizer = auto, initial learning rate = 0.01, final learning rate factor = 0.01, momentum = 0.937, weight decay = 0.0005, workers = 8, close mosaic = 10, and warmup epochs of 3). Evaluation was conducted using the test split with a confidence threshold = 0.25 and IoU/NMS = 0.70. The best model weight was obtained at epoch 275.

And due to the computational cost and time of training YOLO26-M for 300 epochs on a 50,000-image dataset, the process of repeating training with multiple random seeds was not performed in this study. To improve reproducibility, deterministic training was enabled where possible, a fixed seed value was used, and all training/evaluation settings were kept constant. Future work will include multi-seed training runs to report the mean and standard deviation across independent runs.

5. RESULTS AND DISCUSSION

This section explains the results and outcomes After data preparation, configuring the model and training, analyzing model performance using standard detection metrics with qualitative visualizations and comparative analysis.

5.1 Evaluation metrics

These are the metrics used to assess the performance of the model. Standard evaluation metrics commonly used in object detection were employed and widely adopted in YOLO-based systems, providing a comprehensive understanding of detection accuracy and robustness, serving as feedback to the model performance [23].

Precision (P) indicates how many predicted weapons are actually correct.

$$Precision = \frac{TP}{TP + FP}$$

Recall (R): indicator for the ability of the model to capture all weapons and danger instances.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score: assessment of overall detection by balancing precision and recall.

$$F1 = 2\frac{p.r}{p + r}$$

Mean average precision (mAP): how well the model detects and localizes objects across all classes.

$$mAP = \frac{1}{C}\sum_{C=1}^{C} APc$$

where, c is the number of classes and APc is the average precision for the class.

Intersection over Union (IoU): measurement of the overlap between the predicted bounding box and the ground truth box.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

These metrics are standard in modern object detection frameworks and are essential for evaluating real-time detection systems [6].

5.2 Experimental results

The trained YOLO26-M model was evaluated on both datasets using the best-performing epoch obtained during training. The evaluation was conducted on the test split, and results are presented both quantitatively and qualitatively.

5.2.1 Results of dataset1 (dangerous items dataset)

This section will cover the results according to the metrics explained briefly for each class and the model, as shown in Table 6, along with the visualization sample, the curves and the confusion matrix. Comparison results are shown in Table 7 between omiotek and z. Model [21] and the proposed model in this study on the same dataset.

The model achieved (0.954) overall precision, indicating a high ability to identify weapon instances correctly with minimal false positives and this is very important in surveillance systems where the false alarms must be kept as low as possible. The recall value was found to be (0.848), meaning that while the model detects most weapon instances correctly, some of the objects may still not be recognized and

may be missed, aligning with the known challenges in the weapon detection field [8]. Overall mAP@50 of 0.916 proves strong capability for detection, while the mAP@50-95 of 0.773 reflects the performance under stricter localization criteria. The range between the two is a strong indicator for further improvement in bounding box precision. Among the results, we notice some high numbers for the AR and bat in terms of performance and the reason is their large sizes and special features, but in contrast, the handgun showed lower mAP, which backs up the literature indicating small objects are harder to detect accurately [8, 9].

Figure 2 shows performance curves providing further evaluation of models' rating, starting with (a) the precision-recall curve. A strong overall performance is obvious, meaning that the model keeps high precision across a wide range of recall values. The precision-confidence curve (b) shows the increase in precision with higher confidence thresholds, reflecting the reliability of high-confidence predictions. on the contrary, the recall confidence curve (c) shows that recall is highest when the confidence is at the lowest thresholds and decreases when the threshold increases. which highlights that there is a trade-off between avoiding false positives and detecting more objects. The F1 confidence curve (d) provides an optimal balance between precision and recall at a threshold of confidence of nearly 0.5, achieving the maximum F1 score of 0.90 overall. The analysis of these curves confirms that the model is stable, effective and suitable for real-time deployment.

The confusion matrix is shown in Figure 3, providing per-class prediction performance and misclassifications, noticing diagonal values where the numbers are a good indicator for strong classification accuracy, starting from the highest values (bat = 0.92) and then (AR = 0.91) and the (handgun = 0.87). Achieved the highest correct prediction rates. Knife and machete showed lower accuracy (0.84 and 0.80, respectively), indicating difficulty distinguishing these two classes. This is due to the similarities between these objects: both have overlapping shapes and similar features. In addition, a portion of the objects is misclassified across all classes, indicating missed detections. Relatively, the low values of off-diagonal means indicate limited inter-class confusion, and most errors are not related to incorrect classification but to missed detections.

Table 7 shows a comparison of the performance metrics between the two models (previous study v11-m) [21] and the proposed YOLO26-M on the dangerous items study dataset.

Table 6. Dataset1 classes and model results

Classes	P	R	mAP50	mAP 50-95	F1
machete	0.958	0.778	0.884	0.767	0.859
knife	0.923	0.823	0.883	0.760	0.870
bat	0.947	0.906	0.949	0.853	0.926
Assault rifle	0.976	0.882	0.945	0.808	0.927
handgun	0.968	0.852	0.918	0.675	0.907
model	0.954	0.848	0.916	0.773	0.898

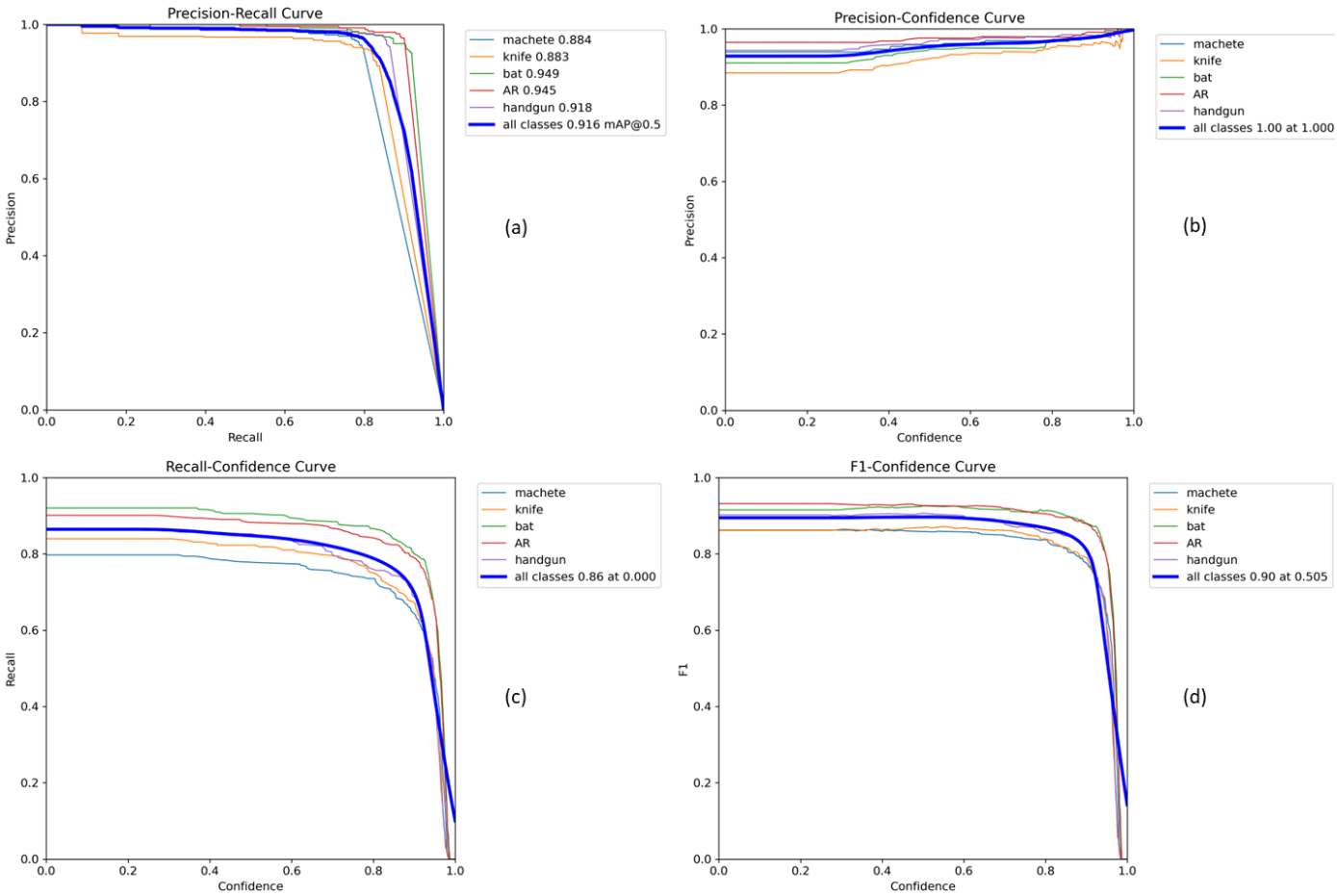

Figure 2. Performance curves for dataset1 model, (a) precision-recall curve, (b) precision-confidence curve, (c) recall-confidence and (d) F1-confidence curve

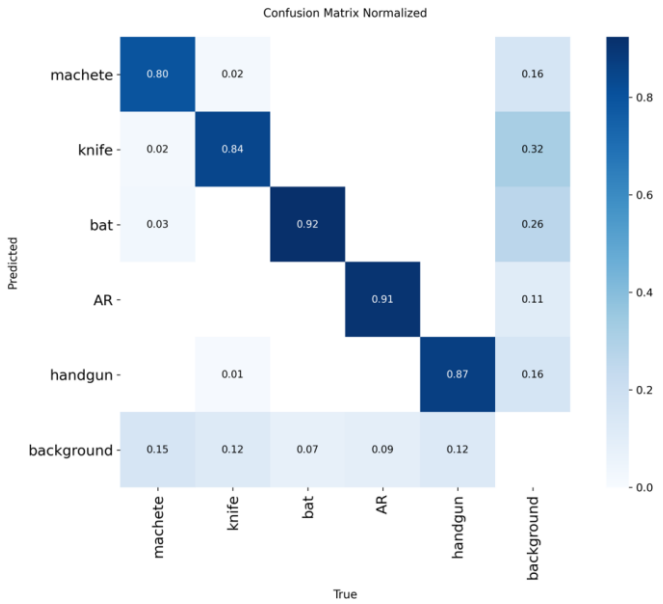

Figure 3. Confusion matrix for dataset1 model

Table 7. Comparison of YOLOv11-M and YOLO26-M

Metric	Previous YOLOv11-M	Proposed YOLO26-M
Precision	0.896	0.954
Recall	0.882	0.848
mAP@50	0.918	0.916
mAP @50–95	0.737	0.773

Table 7 demonstrates the improvement in precision (+5.8%), considered an indicator for weapon and non-weapon object discrimination, reducing false positives. Recall decreased compared with YOLOv11, indicating that YOLO26-M produced fewer but more confident detections. This trade-off may reduce false alarms but may also increase missed detections; therefore, confidence threshold tuning is necessary, depending on whether the deployment prioritizes sensitivity or precision. More importantly, noticing the improvement in mAP @50-95 highlights the enhanced localization accuracy across multiple (IoU) thresholds, reflecting the advancement of the YOLO26 architecture, particularly the ability to balance efficiency and accuracy for real-time tasks. All these results confirm that YOLO26 is a strong candidate for real-time applications.

5.2.2 Results on dataset2 (proposed dataset)

This section is concerned with the trained YOLO26-M on the proposed dataset2, consisting of 50,000 images from multiple categories and by using the metrics defined earlier, the evaluation was done on the test split set. It should be noted that the violence class is a physical aggression event category and differs conceptually from the weapon classes, as it represents a visually observable threat-related event rather than a physical object. Nevertheless, it was included to evaluate the capability of the proposed framework to detect multiple forms of security-relevant threats within a unified surveillance setting. Table 8 shows the exact results.

The model achieved an overall precision of 0.954 and recall of 0.913, resulting in an F1-score of 0.933, which indicates a strong balance between detection accuracy and completeness. In addition, both mAP values demonstrate robust performance across both thresholds.

In Figure 4, the normalized confusion matrix is considered a strong performance for the classification operation, where some true positive rates are noticed in some classes like cutter axe AR and shotgun; other classes like hammer violence and knife also show reliable detection and classes like scissor and rocket launcher have slightly lower accuracy, suggesting that these classes are challenging to distinguish. Misclassification is minimal between the classes, indicating the model learns special features for each category.

However, some of the samples is misclassified as background (up to 0.18 for scissors and 0.13 for rocket launcher). Overall, the matrix confirms that the model achieves high classification accuracy with limited confusion for inter-class. And most errors are associated with missed detections rather than incorrect class predictions.

Table 8. Results for dataset2

Classes	P	R	F1	mAP @0.50	mAP @0.50:0.95
knife	0.939	0.916	0.928	0.949	0.722
cutter	0.985	0.992	0.989	0.995	0.959
scissor	0.968	0.808	0.881	0.911	0.737
axe	0.990	0.984	0.987	0.994	0.939
hammer	0.966	0.904	0.934	0.961	0.866
handgun	0.939	0.866	0.901	0.926	0.844
Assault rifle	0.982	0.994	0.988	0.995	0.987
shotgun	0.937	0.939	0.938	0.965	0.876
Rocket launcher	0.920	0.850	0.884	0.917	0.751
violence	0.913	0.877	0.895	0.933	0.723
model	0.954	0.913	0.933	0.955	0.840

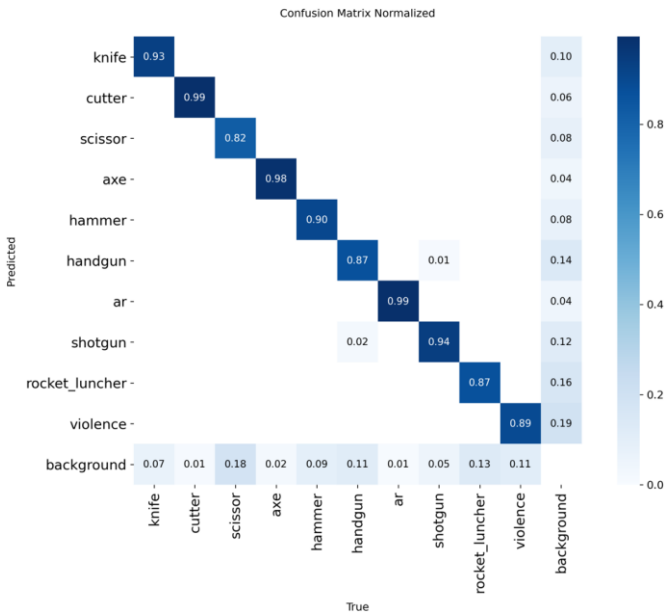

Figure 4. Confusion matrix for dataset2 model

Figure 5 shows evaluation curves where similar trends were observed in dataset1 with improved performance. The Precision–Recall curve (a) shows strong overall detection, with a high mAP@0.5 of 0.955, indicating that the model maintains high precision across most levels of recall. The Precision–Confidence curve (b) reveals that precision improves steadily along with confidence confirming the reliability of high-confidence predictions. In contrast, the recall–confidence curve (c) shows that recall is highest at

lower confidence levels (nearly 0.95) and decreases as the threshold increases. The F1–Confidence curve (d) identifies that the model achieved higher precision across confidence thresholds and a higher F1 score at a lower optimal threshold (0.335). Overall, the curves tell how model performance is considered robust and suitable for real-world applications.

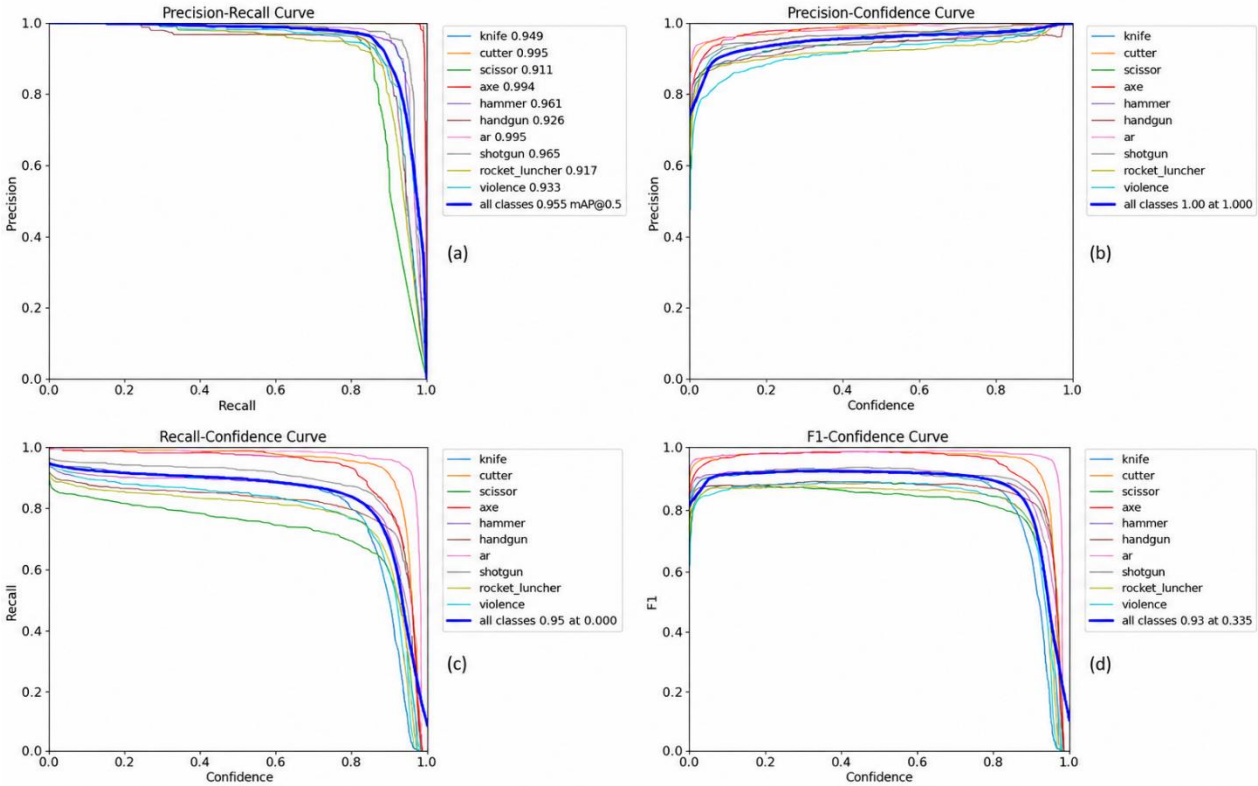

Figure 5. Performance curves for dataset2 model, (a) precision-recall curve, (b) precision-confidence curve, (c) recall-confidence and (d) F1-confidence curve

5.2.3 Cross-dataset validation using YouTube Gun Detection Dataset

Although dataset1 already provides direct comparison against previously reported YOLO family versions from the original study, an additional evaluation was conducted on the independent YouTube Gun Detection Dataset (GDD) benchmark dataset to investigate the generalization behavior of recent YOLO generations under different surveillance scenarios. This supplementary experiment included YOLOv8, YOLOv11 and YOLO26 trained and evaluated under identical settings using the official dataset protocol; the results are summarized in Table 9.

The YOUTUBE GDD benchmark provides complementary evidence rather than a direct claim of universal superiority. YOLO26 achieved the highest overall precision and the best mAP@50-95, indicating improved localization under stricter IoU thresholds. Therefore, the GDD results indicate that YOLO26 provides a competitive accuracy-efficiency trade-off, especially in localization robustness. This balance supports the use of YOLO26 as a practical detector.

Table 9. Comparative benchmark evaluation on Gun Detection Dataset (GDD)

YOLO Model	P	R	mAP @50	mAP @50-95	Latency (ms)
v8	0.801	0.718	0.767	0.618	5.421
v11	0.812	0.696	0.749	0.616	6.001
26	0.815	0.701	0.754	0.624	6.336

5.2.4 Failure cases and limitations analysis

Although a strong overall performance is achieved by the proposed framework across datasets, several challenges were identified during the qualitative inspection of the prediction results. Most of the failure cases are associated with small object size, partial occlusion and complex background. In addition to the low illumination conditions and visual similarities between classes, as illustrated in Figure 6.

Figure 6. Failure cases samples

For dataset1, some confusion was observed between the knife and the machete due to their similar geometric characteristics and features. In addition, handgun detection occasionally failed when the object occupied only a small portion of the image, which is consistent with findings reported in previous weapon-detection studies.

And for dataset2, the scissor and rocket launcher classes produced lower recall compared to the remaining classes. Scissors often appeared as thin objects with limited information, making them more difficult to localize accurately. Rocket launchers presented an additional challenge due to their diverse appearance, viewing angles and lower representation in publicly available surveillance imagery.

The violence category also exhibited greater variability than conventional weapon classes because the detection here depends on human posture and interaction patterns rather than a clearly defined physical object, leading to some misdetections occurring in the complex scenes containing multiple individuals.

These observations indicate that future improvements may be achieved through additional training samples representing small-scale objects, enhanced processing for low-light images and the incorporation of temporal information for behavior-related event detection.

5.3 Speed analysis and real-world deployment

An NVIDIA H100 GPU (80GB HBM3) was used to infer the performance of the proposed model of YOLO26-M using 960x960 pixels of input resolution, which is consistent with the training configuration. The two datasets are evaluated and considered for preprocessing, inference, and post-processing stages.

For dataset1, the measured latency was approximately 2.7 ms per image using the NVIDIA H100 environment. These results demonstrate highly efficient inference performance under high-performance computing conditions while maintaining strong accuracy for detection. These findings are considered an indicator of the suitability of the proposed framework for latency-sensitive objects, although the deployment performance is expected to vary depending on the hardware platform in use.

For dataset2, high efficiency execution was demonstrated; the time measured as follows: (Preprocessing: ~0.1 ms, Inference: ~1.7 ms, Postprocessing: ~0.1 ms). This performance indicates that the model operates well beyond conventional real-time requirements.

The experimental evaluation was conducted on high-performance data-centre hardware (NVIDIA H100), representing an upper bound of achievable performance. To take the assessment of deployment feasibility beyond the data-center environment, the trained YOLO26 model on dataset2 was additionally evaluated on consumer-grade hardware (NVIDIA GeForce RTX 3050 Laptop GPU, 6 GB VRAM) using the same split and input resolution (960 × 960). The model maintained comparable detection performance, achieving a precision of 0.953, a recall of 0.913, a mAP@50 of 0.954, and a mAP@50-95 of 0.840, as shown in Table 10. These results indicate that the proposed framework preserves its detection capability on lower-resource hardware, although inference throughput is naturally reduced compared with the H100 platform. In practical deployment scenarios, the system is expected to operate on more resource-constrained platforms such as NVIDIA RTX-series GPUs and embedded systems (e.g., Jetson family). On such platforms, an increase in inference latency is expected due to reduced computational resources. However, the achievable inference speed remains dependent on computational resources that are available, deployment configuration and optimization strategy adopted for the target platform.

The deployment benchmark demonstrates that the proposed framework preserves its detection performance across different hardware platforms. While inference latency increased from approximately 1.9 ms on the NVIDIA H100 to 98.6 ms on the RTX 3050, the rest of the values remained unchanged, indicating that the reported H100 measurements represent an upper-bound performance estimation, where the RTX provides a more realistic deployment scenario for simple consumer-grade hardware.

Figure 7 illustrates a practical deployment scenario for the proposed surveillance framework, in which the YOLO26-M detector processes the inputs captured by the cameras. Predictions exceeding a predefined confidence threshold are treated as potential threats and generate alerts for further inspection. and to reduce the impact of false detections, human verification is recommended before initiating any security-related actions. Once the threat or danger is confirmed, the system can assist security personnel by providing rapid notification and situational awareness for appropriate response measures.

Table 10. Hardware results comparison

Hardware	P	R	mAP @50	mAP 50-95	Latency (ms)	FPS
H100 80GB	0.954	0.913	0.955	0.840	1.9	526
RTX 3050 Laptop 6GB	0.953	0.913	0.954	0.840	98.6	10.1

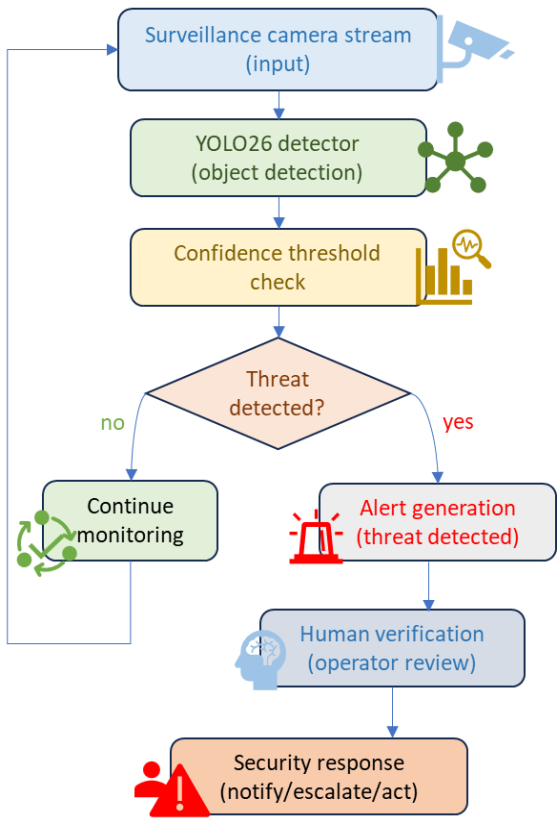

Figure 7. Proposed deployment workflow

For practical deployment, additional optimization techniques can be applied, like Quantization (FP16 / INT8) or TensorRT acceleration and even Batch size tuning. These

techniques are widely used to reduce latency and improve throughput without significant degradation in detection performance.

From an architectural perspective, the use of high input resolution (960 × 960) improves detection of small objects and partially occluded weapons. However, it also increases computational cost. Therefore, deployment configurations could be taken with considerations such as reducing input resolution and/or selecting lighter model variants (e.g., YOLO26-S or YOLO26-N) to achieve a better balance between speed and accuracy depending on hardware constraints.

5.4 Ethics and privacy considerations

The development of intelligent systems requires careful considerations related to ethical, privacy and security implications. The proposed framework is designed to improve public safety through early detection, but to prevent misuse and protect individual rights, its deployment must be accomplished with appropriate safeguards.

All the datasets used in this study were collected from public sources, existing research, online repositories, and manually acquired images intended exclusively for research and academic purposes. Most of the dataset information focuses on weaponry from different angles to make the recognized personnel less identifiable, and personally identifiable information was not intentionally collected and the focus of the annotation process was restricted to threat-related objects and activities. In practical deployment scenarios, some additional privacy-preserving measures should be considered, such as video anonymization and face blurring to minimize unnecessary exposure of personal information.

Furthermore, the proposed system should be viewed as a decision-support tool rather than a fully autonomous security solution. The outcomes of the detection process may be affected by environmental conditions, object occlusion, lighting variations, or even visual similarities between similar classes. Therefore, human-in-the-loop is heavily recommended before initiating security-related actions or interventions. This approach can reduce the impact of false alarms and improve operational reliability.

Also, it should be noted that access to surveillance data and detection results should be controlled through appropriate cybersecurity and data protection mechanisms, preventing unauthorized access or misuse. Future deployments should also comply with applicable privacy regulations, institutional policies, and ethical guidelines governing the use of AI in public environments.

By taking these considerations seriously and incorporating them, intelligent weapons detection systems can contribute to public safety while maintaining transparency, accountability and respect for individual privacy.

6. CONCLUSION

The increasing rate of violent incidents has pushed the need for intelligent surveillance systems with the capability of early threat detection and notification, along with categorizing the threat. In this context, deep learning-based approaches have demonstrated significant potential in enhancing public safety by enabling automated and real-time monitoring of critical environments.

This work proposes and evaluates a real-time weapon detection and classification framework based on the YOLO26 architecture (*m* variant) trained across multiple weapon categories. The model results show a balance between detection accuracy and computational efficiency for object detection ,especially in difficult conditions such as complex backgrounds, inconspicuous, and small sizes of objects. These are essential for practical application. In addition to weapons, the framework was evaluated on a threat-related physical aggression category (violence), demonstrating the flexibility of the proposed approach in identifying different types of security-relevant events within surveillance imagery.

The model demonstrated a strong, balanced detection accuracy and computational efficiency across the evaluated datasets, including successful testing on both data-center and consumer-grade hardware. These results suggest the potential applicability of the proposed framework in surveillance-oriented scenarios such as public places and surveillance-oriented public environments. The ability of the model to detect multiple weapon categories and frame-level physical aggression events suggests its potential as a decision-support component for future early warning surveillance systems. The results suggest the feasibility of future deployment in surveillance environments; further validation is required on real CCTV streams and edge hardware.

Overall, this work demonstrates the effectiveness of the YOLO26-M architecture for multi-class weapon detection and frame-level physical aggression event detection under experimental surveillance-oriented conditions. The proposed framework achieved strong detection performance while maintaining computational efficiency under experimental conditions.

Additional validation on the independent YouTube GDD benchmark further demonstrated the ability of YOLO26 to maintain competitive performance relative to recent generations across different datasets. Future work will focus on deployment optimization, evaluation on embedded and edge platforms and the integration of multimodal information sources such as audio events and contextual scene understanding, to further improve robustness in complex surveillance environments.

REFERENCES

[1] Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C., Anjum, M.R. (2021). Weapon detection using YOLO V3 for smart surveillance system. Mathematical Problems in Engineering, 2021(1): 9975700. https://doi.org/10.1155/2021/9975700

[2] Hashmi, T.S.S., Haq, N.U., Fraz, M.M., Shahzad, M. (2021). Application of deep learning for weapons detection in surveillance videos. In 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), Islamabad, Pakistan, pp. 1-6. https://doi.org/10.1109/ICoDT252288.2021.9441523

[3] Quyyum, M.E.E., Abdullah, M.H.L. (2023). Weapon detection in surveillance videos using deep neural networks. In Proceedings of the Multimedia University Engineering Conference (MECON 2022), pp. 183-195. https://doi.org/10.2991/978-94-6463-082-4_19

[4] Salido, J., Lomas, V., Ruiz-Santaquiteria, J., Deniz, O. (2021). Automatic handgun detection with deep learning in video surveillance images. Applied Sciences, 11(13):

6085. https://doi.org/10.3390/app11136085
[5] Brahmaiah, M., Madala, S.R., Mastan Chowdary, C. (2021). Artificial intelligence and deep learning for weapon identification in security systems. Journal of Physics: Conference Series, 2089(1): 012079. https://doi.org/10.1088/1742-6596/2089/1/012079
[6] Thakur, A., Shrivastav, A., Sharma, R., Kumar, T., Puri, K. (2024). Real-time weapon detection using YOLOv8 for enhanced safety. arXiv preprint arXiv:2410.19862. https://doi.org/10.48550/arXiv.2410.19862
[7] Sapkota, R., Cheppally, R.H., Sharda, A., Karkee, M. (2026). YOLO26: Key architectural enhancements and performance benchmarking for real-time object detection. arXiv preprint arXiv:2509.25164. https://doi.org/10.48550/arXiv.2509.25164
[8] Berardini, D., Migliorelli, L., Galdelli, A., Marín-Jiménez, M.J. (2025). Edge artificial intelligence and super-resolution for enhanced weapon detection in video surveillance. Engineering Applications of Artificial Intelligence, 140: 109684. https://doi.org/10.1016/j.engappai.2024.109684
[9] Gu, Y., Liao, X., Qin, X. (2022). YouTube-GDD: A challenging gun detection dataset with rich contextual information. arXiv preprint arXiv:2203.04129. https://doi.org/10.48550/arXiv.2203.04129
[10] Qi, D., Tan, W., Liu, Z., Yao, Q., Liu, J. (2021). A dataset and system for real-time gun detection in surveillance video using deep learning. arXiv preprint arXiv:2105.01058. https://doi.org/10.48550/arXiv.2105.01058
[11] Murugan, T., Badusha, N.A.N.M., Semaihi, A.R.O.A., Alkindi, M.M.R., Alnaqbi, E.M.R., Alketbi, G.H.T. (2025). AI-based weapon detection for security surveillance: Recent research advances (2016–2025). Electronics, 14(23): 4609. https://doi.org/10.3390/electronics14234609
[12] Mangrolia, J., Sheth, R. (2021). Detection and classification of weapon using gradient orientation and laplacian magnitude-based histogram. PIMT Journal of Research, 13(4): 140-145.
[13] Sabri, M.M., Hoommod, H.K., Hussein, K.A. (2025). Security surveillance systems based on deep learning and Blockchain techniques: A review. Mustansiriyah Journal of Pure and Applied Sciences, 3(3): 173-193. https://doi.org/10.47831/mjpas.v3i3.326
[14] Sernani, P., Falcionelli, N., Tomassini, S., Contardo, P., Dragoni, A.F. (2021). Deep learning for automatic violence detection: Tests on the AIRTLab dataset. IEEE Access, 9: 160580-160595. https://doi.org/10.1109/ACCESS.2021.3131315
[15] Mane, S.B. (2024). Weapon detection and classification using deep learning. ITEGAM-Journal of Engineering, Technology and Industrial Applications, 10(47): 19-26. https://doi.org/10.5935/jetia.v10i47.1039
[16] Abi-Nader, D., Harb, H., Jaber, A., Mansour, A., Osswald, C., Mostafa, N., Zaki, C. (2025). MARIE: One-stage object detection mechanism for real-time identifying of firearms. Computer Modeling in Engineering & Sciences, 142(1): 279-298. https://doi.org/10.32604/cmes.2024.056816
[17] Pravesh, R., Sahana, B.C. (2025). Robust firearm detection in low-light surveillance conditions using YOLOv11 with image enhancement. International Journal of Safety and Security Engineering, 15(4): 797-809. https://doi.org/10.18280/ijsse.150416
[18] Bhatti, M.T., Khan, M.G., Aslam, M., Fiaz, M.J. (2021). Weapon detection in real-time CCTV videos using deep learning. IEEE Access, 9: 34366-34382. https://doi.org/10.1109/ACCESS.2021.3059170
[19] Ahmed, S., Bhatti, M.T., Khan, M.G., Lövström, B., Shahid, M. (2022). Development and optimization of deep learning models for weapon detection in surveillance videos. Applied Sciences, 12(12): 5772. https://doi.org/10.3390/app12125772
[20] Reyes, A.L.E., Cruz, J.C.D. (2025). Anomalous weapon detection for armed robbery using Yolo V8. Engineering Proceedings, 92(1): 85. https://doi.org/10.3390/engproc2025092085
[21] Omiotek, Z. (2025). Effectiveness of modern models belonging to the YOLO and vision transformer architectures in dangerous items detection. Electronics, 14(17): 3540. https://doi.org/10.3390/electronics14173540
[22] Valliappan, N.H., Pande, S.D., Reddy Vinta, S. (2024). Enhancing gun detection with transfer learning and YAMNet audio classification. IEEE Access, 12: 58940-58949. https://doi.org/10.1109/ACCESS.2024.3392649
[23] Saleh, K.T., Karim, A.A. (2025). Enhancing communication with elderly and stroke patients based on sign-gesture translation via audio-visual avatars. Open Engineering, 15(1): 20240068. https://doi.org/10.1515/eng-2024-0068
[24] Saleh, K.T., Karim, A.A. (2024). Building a dataset of pointing gestures for elderly people in Iraqi nursing homes. Ingénierie des Systèmes d'Information, 29(6): 2455-2465. https://doi.org/10.18280/isi.290632