Digital Signal Processing 132 (2023) 103790

Contents lists available at ScienceDirect

Digital Signal Processing

journal homepage: www.elsevier.com/locate/dsp

Fighting against terrorism: A real-time CCTV autonomous weapons detection based on improved YOLO v4 ☆

Guanbo Wang[a,b], Hongwei Ding[a,b,*], Mingliang Duan[c], Yuanyuan Pu[a,b], Zhijun Yang[a,b], Haiyan Li[a,b]

[a] *College of Information Technology, Yunnan University, Chenggong District, Kunming, Yunnan Province, China*
[b] *The Key Laboratory of Internet of Things Technology and Application in Yunnan Province, China*
[c] *Yunnan Provincial Highway Network Toll Management Limited Company, China*

ARTICLE INFO

Article history:
Available online 21 October 2022

Dataset is released in https://github.com/Philharmy-Wang/Fighting-against-terrorism

Keywords:
Weapon detection
Object detection
Synthetic data
YOLO
Security detection

ABSTRACT

In recent years, deep learning has demonstrated tremendous potential in the real world. Object detection is a critical real-world task for deep learning. You Only Look Once (YOLO) object detection model recognizes interesting regions in images with impressive accuracy and real-time performance. The objective of this paper is to apply object detection to the field of security and counter-terrorism. Individuals are protected from violence by recognizing and locating the guns on closed-circuit television (CCTV). This paper presents a real-time detection approach for CCTV autonomous weapons based on YOLO v4. For the characteristics of CCTV scenarios, we propose the YOLO v4 backbone with Spatial Cross Stage Partial-ResNet (SCSP-ResNet). Meanwhile, the receptive field enhancement module is shown to capture fine semantic features of high-dimensional small objects. The Fusion-PANet (F-PaNet) module has been used to fuse multi-scale information to improve the model's perceptive power on the region of interest. Furthermore, we merge synthetic and real-world datasets to comprehensively investigate the effects of synthetic datasets on detectors. The experimental results reveal that our suggested detection model improves mAP (mean Accuracy Precision) and inference time by 7.37% and 4.2%, respectively. The model's parameter is reduced by 0.349 BFLOP/s(billion floating point operations per second). The proposed detector outperforms the baseline model in terms of accuracy, real-time, and robustness.

1. Introduction

In recent years, with the advancement of hardware technology and the combined efforts of academics from around the world, deep learning's superior performance has led to its widespread application in everyday life. Object detection is a crucial task in the areas of deep learning and computer vision, which have applications in autonomous driving, intelligent cities, and smart transportation. In the field of closed-circuit television (CCTV), the detection of strangers is a highly developed technology. However, the system is incapable of estimating the threat level by only detecting intruders, which prevents accurate and timely response plans from being created. In densely populated areas, such as schools and retail malls, terrorists armed with automatic weapons are incredibly destructive and frequently cause catastrophic damage. Therefore, the development of a rapid, accurate, and effective automatic weapon detection system is crucial for preserving social stability and saving lives.

CCTV-based security detection plays a crucial function in a variety of scenarios. However, detection in particular scenarios often faces the challenges of complex backgrounds and small objects. For example, in a complex background, the handgun in the hands of the holder only takes up less than 3% of the CCTV screen. Enríquez, Fernando et al. [1] identified the limitations of the current security system and designed a security system for the rapid detection of weapons. When a potential threat is detected, this system alerts security staff so that they can commence preparing an immediate response. Therefore, building detectors must consider both accuracy and real-time.

☆ **Funding information**. The National Natural Science Foundation of China, Grant/Award Numbers: 61461053, 61461054, 62166048, 62066047, 61861045. "Famous teacher of teaching" of the Yunnan 10000 Talents Program, the Major Science. The Major Science and Technology Project of Yunnan Province, Grant/Award Numbers: 2018ZF017.

* Corresponding author at: College of Information Technology, Yunnan University, Chenggong District, Kunming, Yunnan Province, China.
E-mail address: wgb@mail.ynu.edu.cn (H. Ding).

https://doi.org/10.1016/j.dsp.2022.103790
1051-2004/© 2022 Elsevier Inc. All rights reserved.

The cameras of CCTV devices are typically installed at a great height to capture a wide range of images. Therefore, the size of objects in CCTV is smaller. González, Jose L. Salazar et al. [2] mentioned that the handgun occupies less than 3% of the image in the CCTV. In addition, the quality of CCTV images can be affected by ambient lighting. Inadequate lighting conditions can cause video stream noise and reduce the contrast between objects and backgrounds. In this paper, we propose an enhanced YOLO v4 object detection method in order to enhance the perception of small objects in CCTV.

The collection and labeling of datasets present a challenge for applying object detection to specific applications. In reality, there are many difficulties with data collection. On the one hand, the environment and equipment affect the quality of the data collected, requiring manual selection. On the other hand, the data collection process could be legally liable (for example, when someone's portrait rights were violated). Utilizing synthetic images to augment the dataset is not a novel practice. In 2007, Taylor et al. [3] employed Half-Life 2 virtual simulation data for video surveillance design and evaluation. Wang et al. [4] implemented a rendering pipeline to generate realistic head images in order to address the issue of insufficient datasets in human-computer interaction head pose estimation. González et al. [2] developed Unity Game Engine to generate CCTV datasets simulating real weapons, and combined them with real-world datasets to test the efficacy of synthetic data on Faster R-CNN. Therefore, we combine the real-world dataset and the synthetic dataset from Unity (2020) at various scales and employ them in a complex algorithm for detecting small objects in the background. In addition, we investigate the optimal dataset combination and training scheme for models using transfer learning and other techniques.

To develop a fast and effective CCTV automatic weapon detection model, we propose the following two design objectives:

1. We must (i) verify whether the synthetic dataset can improve the model performance and (ii) determine the optimal training scheme for the model.
2. We need a faster and more accurate object detection model to detect automatic weapons in CCTV images.

Consequently, based on the objectives of the research, our contribution is summarized as follows:

1. We increase the dataset size to address the issue that the CCTV gun dataset is constrained and challenging to gather. The results of the experiments in Section 5 show that the approach works.
2. We employ transfer learning and different dataset combination schemes to analyze the optimal fusion method and the training scheme of the dataset to investigate the model's performance further.
3. The proposed model employs pruning to reduce Neck's high-dimensional redundant convolutional layers. It improves the model's real-time performance with a small loss of accuracy.
4. We propose the SCSP-ResNet structure, which is implemented in the backbone portion of YOLO v4. By employing a residual-spatial mechanism, it is capable of removing the complex background information while simultaneously increasing the spatial dimensional information of object features.
5. We present the receptive field enhancement module and implement it on the YOLO v4 backbone deep layer. This module instructs the model to acquire finer, higher-dimensional features of small objects prior to feature fusion.
6. To improve the robustness of the model's perception of small objects, we propose the F-PaNet module, which could merge three different scales of anchors from the YOLO v4 detection layer.

2. Related work

2.1. Object detection in security scenarios

CCTV surveillance systems are becoming more effective as smart cities and intelligent transportation become increasingly popular. CCTV monitoring is a vital component of the security industry. However, CCTV surveillance's disadvantage is that an observer must simultaneously watch multiple video streams. Velastin et al. [5] mentioned that even a trained professional observer cannot focus on surveillance video for long periods of time. After 20-40 minutes, the observer becomes "video blind". Moreover, CCTV surveillance relies solely on human observers for video surveillance, which is inefficient as the surveillance area expands and human attention is diverted [1].

Currently, researchers are focused on applying object detection algorithms to various real-world scenarios, and object detection algorithms in the field of security is one of the most popular research topics. Samet Akcay et al. [6] investigated X-Ray in security and conduct a comprehensive investigation of object detection based on traditional methods and deep learning, which demonstrates the auxiliary role of object detection for manual screening in security. Olmos et al. [7] proposed an automatic handgun detector based on video streams and verified the effectiveness of the Region Proposal Networks (RPN) method by comparing it with the sliding windows method. Castillo et al. [8] introduced a real-time detector for surveillance video based on cold weapons (kitchen knives, daggers, machetes, etc.), which enhanced the robustness of the model to different lighting conditions by adjusting the brightness. Pang, Lei, et al. [9] presented a millimeter wave weapon image detection system based on YOLO v3 and employed a small sample of millimeter wave weapon images for security screening with satisfactory accuracy and real-time performance. Pérez-Hernández et al. [10] constructed a surveillance camera dataset with six categories and applied it to a two-stage object detector, which effectively reduce the number of false positives in surveillance videos.

2.2. Small object detection in complex backgrounds

The majority of object detection algorithms are designed to detect objects with regions of interest that are larger than 32 × 32 pixels. However, some practical applications require the detection of objects with a smaller proportion of pixels within the image. The MS COCO dataset proposed by Microsoft defines the absolute scale of small objects as follows [11]:

1. A region of interest is deemed to be a small object when its area is less than 32 × 32 pixel values.
2. A region of interest is regarded as being small object when its height and width are less than 0.1 of the size of the original image.

Unlike large and medium-sized objects, small objects have issues like less pixels and semantic information, resulting in undesired detection effects of small object detection on existing models. The following characteristics are exhibited by small objects in complex scenes:

1. **Local prominence.** Small objects may not be the most significant in the overall image, but local features are typically more prominent.

2. **Global sparsity.** Due to the fact that small objects make up a negligible portion of the image, objects can be considered globally sparse even if there are many of them.

Therefore, the detection of small objects in complex scenes faces the following technical challenges:

1. There is no large-scale dataset of small objects. MS COCO [11], ImageNet [12], Pascal VOC [12], and other large-scale public datasets frequently employed in the field of object detection are focused on the detection of commonly scaled objects. Existing publicly accessible small object datasets have particular application scenarios. There are still relatively few large-scale generalized small object datasets.

2. The detection of small objects could be hindered by complex backgrounds. Current small object detection has specific application scenarios, including remote sensing [13] and other complex domains. In complex backgrounds, the information of small objects can be obscured by other larger objects or blended with the background due to insufficient image contrast. In addition, in scenes such as CCTV surveillance, noise is also generated due to insufficient ambient light. A study by Ravichandran A et al. [14] pointed out that the noise of images has a great negative impact on small objects in object recognition. KaiShuang et al. [15] mentioned that the small objects contained less information and had a high possibility of confusing themselves with the background.

3. Convolutional neural networks have limitations for feature extraction of small objects. Regular-scale object detection employs convolutional neural networks for multiple down-sampling to enlarge the receptive field of the model [16]. However, the edge information and semantic information of small objects are blurred. Moreover, multiple down-sampling operations result in the loss of feature information for small objects, which reduces the performance of the detector.

In an effort to address the technical challenges of small object detection, numerous academics and research institutions have devoted themselves to the study of small object detection enhancement techniques. On the one hand, data augmentation can be applied to enhance the performance of models for detecting small objects. Traditional data augmentation enlarges the training dataset by rotating, inverting, cropping at random, scaling, and altering the color of the input image. Expanding small object data in images (Mosaic data augmentation by YOLO v4 [17], MixUP data augmentation technique by PP-YOLO [18], etc.) and increasing image resolution can be utilized to provide more detailed information for small object detection when these image processing-based data augmentation methods fail to produce meaningful results. On the other hand, multi-scale feature fusion can also improve the performance of small object detection. The convolutional neural network-based small object detection method is well-suited for detecting small objects with smaller receptive fields and higher spatial resolution in the shallow network. However, shallow layer features have weak semantic information and a low recall rate [19].

The semantic information of the deeper layer of the neural network is more abundant, but the direct extraction could result in the loss of small object information after multiple down-sampling operations. Several object detection models (YOLO v4 [17], SSD [20], etc.) have adopted feature fusion in order to fuse the multi-scale feature maps and address the bottleneck in convolutional neural network feature extraction. Brais Bosquet et al. [21] proposed an end-to-end spatio-temporal convolutional neural network for small object detection in video that improved detection accuracy by correlating the highest ranked regions with regions of interest containing small objects over time.

2.3. Synthetic datasets in deep learning

Expanding a model's training data with synthetic datasets is a reliable data augmentation strategy that can effectively enhance model performance. In the field of computer vision, such as license plate recognition [22] and medical image segmentation and detection [23], the scheme has been widely employed for deep network training. Hattori H et al. [24] proposed that synthetic data can compensate for the absence of real training data. Liu WX et al. [25] utilized a large number of annotated synthetic aircraft images to reduce texture distortion and noise, thereby enhancing the performance of aircraft detection in remote sensing images. Kim, JH et al. [26] adopted generative adversarial network to convert RGB images to infrared images, which enlarged the size of the dataset and the detection network achieved better performance. He et al. [27] generated a large-scale synthetic dataset that can be applied to the recognition and analysis of ship objects in aerial imagery, demonstrating the utility and potential of synthetic data in image recognition and comprehension tasks. To demonstrate the effectiveness of using synthetic datasets in deep learning, Öhman W et al. [28] produced a synthetic weapon dataset using the military simulator VBS3, combined it with the real dataset in various ways, and trained it on the Faster-R CNN with various training strategies. González et al. [2] generated a dataset for weapon detection utilizing Unreal Engine and merge it with a dataset from realistic surveillance to train a Faster R-CNN, demonstrating the impact of using synthetic data in weapon detection systems. In addition, the research compared synthetic data with actual surveillance data, which significantly advanced the field. Consequently, synthetic data can enhance the performance of models with insufficient datasets.

3. Datasets characteristics

We build the gun detection dataset, which consists of three components: 1. synthetic dataset, 2. dataset of rifles and handguns from Openimg [29], 3. CCTV gun detection dataset provided by González et al. [2]. Table 1 contains the details of the dataset. The instance box of the dataset is depicted in Fig. 1. In Section 4, we conduct a comprehensive evaluation of this dataset utilizing multiple object detection algorithms to determine whether the synthetic dataset can improve the model's performance and which training methods can optimize the model's performance.

3.1. Synthetic datasets

The Unity Game Engine generates synthetic datasets by simulating outdoor and indoor scenes. Multiple screen captures are installed in virtual scenes designed to simulate real-world camera positions. In addition, the dataset is made more realistic by multiple cameras capturing multiple characters, actions, and scenes with four types of handguns, five types of rifles, a smartphone, and a knife. Compared to the actual dataset, the synthetic dataset still contains gaps. However, synthetic data can help the network better focus on important regions during training, and we demonstrate how helpful synthetic data is in the experiments in Section 5.3.

With the intent of capturing images and completing the collection of datasets, the camera is initially positioned in a solitary area and the armed and unarmed virtual characters are allowed to move through the scene along a predetermined path, permitting them to turn around, stop, and perform other actions. After annotating the captures, the annotated category and location information are recorded to an *.xml* file in the PASCAL VOC format. In addition, we convert the data to *.json* files in the MS COCO format in the event that training could be performed on detectron 2

Table 1
Details of the dataset.

Abbreviations	Dataset	Description	Number of images	Total images
S	Synthetic dataset	Spilt-2500	2500	10231
R	Real-world simulation attack dataset	Cam 1	607	
		Cam 5	3511	
		Cam 7	1031	
O	Openimg public dataset	Rifle(include Rifle and Shotgun)Handgun	2582	

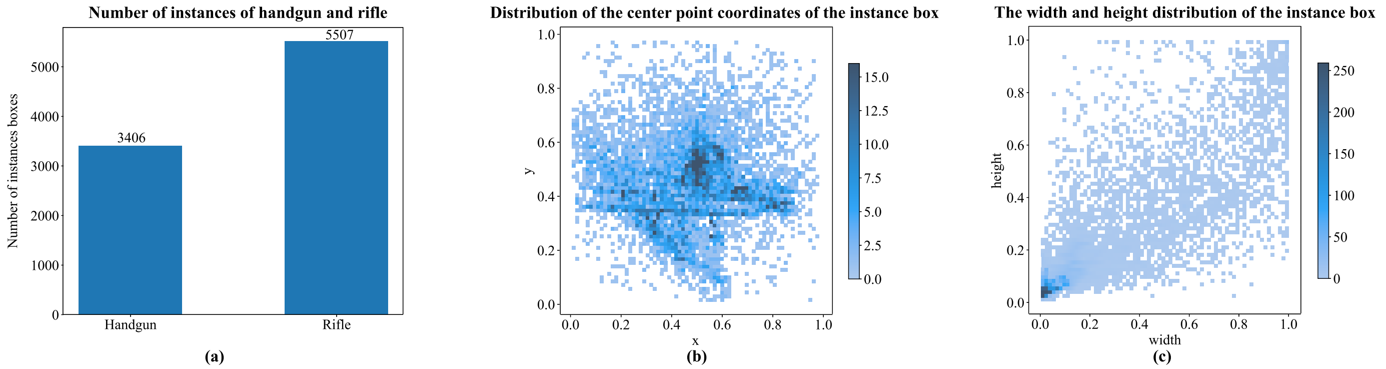

Fig. 1. Distribution of dataset instance boxes. (a) displays the number of instance boxes for handguns and rifles, with 0.87 instance boxes per image on average. (b) indicates that the majority of the example boxes' centers are located in the image's center. (c) explains that the width and height of the majority of the image's example boxes are less than 0.1 pixels.

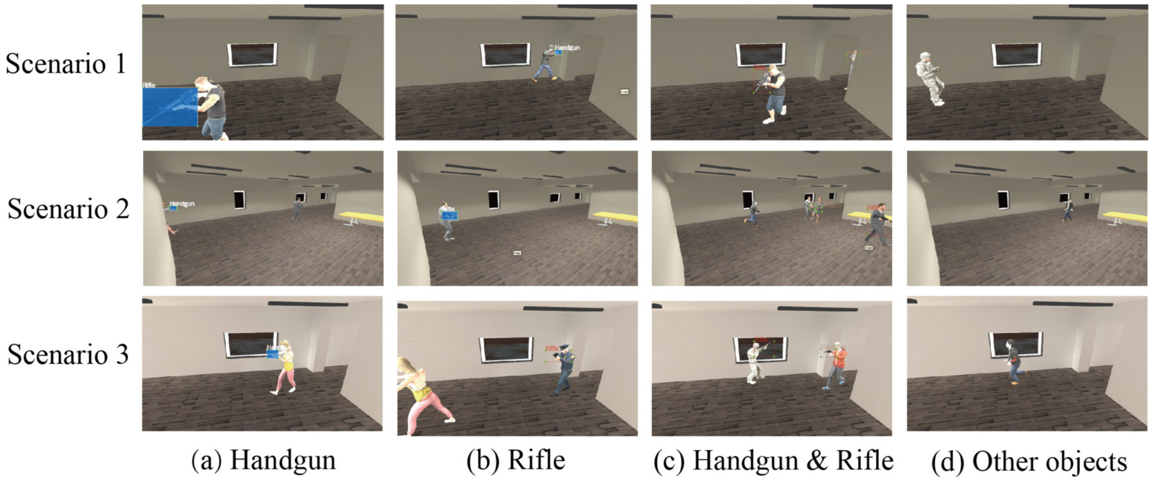

Fig. 2. Partial synthesis of images from the dataset. We chose synthetic images of three distinct scenes, totaling 2500 images, including four types of images: handgun only, rifle only, both handgun and rifle, and no object detected.

[30]. Fig. 2 illustrates a section of the synthetic dataset. Fig. 3 displays the distribution of the instance boxes within the synthetic dataset. Fig. 3 (b) demonstrates that the instance boxes of the synthetic dataset are concentrated in the image's center and bottom. The majority of the instance boxes have width and height dimensions that are less than 0.1.

3.2. Real-world simulation attack dataset

The real-world attack simulation dataset produced by González et al. [2] consists of images from three distinct real-world scenarios within a university. CAM 1, CAM 5, and CAM 7 are the camera codes for the three cameras acquiring the dataset, which are located in distinct locations within the same area. CAM 1 and CAM 7 are located in corridors with adequate lighting and objects similar to those detected, such as doors, trash cans, and fire extinguishers. Fig. 4 depicts the images of real-world attack simulation datasets. The following is the specific information about each camera.

CAM 1: This camera captured 40 minutes and 25 seconds of surveillance video. We select five videos with people's movement, extracted 2 frames per second, and annotated them manually, resulting in 607 images.

CAM 7: This camera captures 1 hour, 3 minutes, and 34 seconds of surveillance video similar to Cam1, and we select videos with task movement by manually annotating 2 frames per second, for a total of 3511 images.

CAM 5: This camera is positioned at an entrance with an uneven distribution of light and a high contrast between light and dark. 39 minutes and 7 seconds of video were captured, and 8 minutes and 36 seconds of video with human movement were selected, 2 frames per second are extracted, and manual annotation is used, for a total of 1031 images.

The distribution of the instance frames from the dataset for the real-world simulation attack is shown in Fig. 5. Fig. 5 (b) demonstrates that the majority of instance boxes from the synthetic dataset are located in the lower half of the image. Fig. 5 (c)

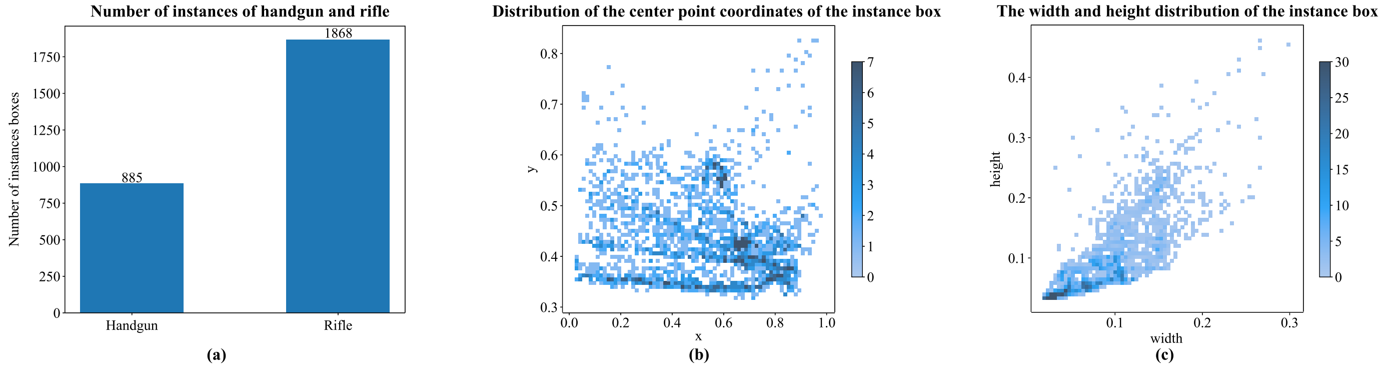

Fig. 3. Distribution of instance boxes for synthetic datasets.

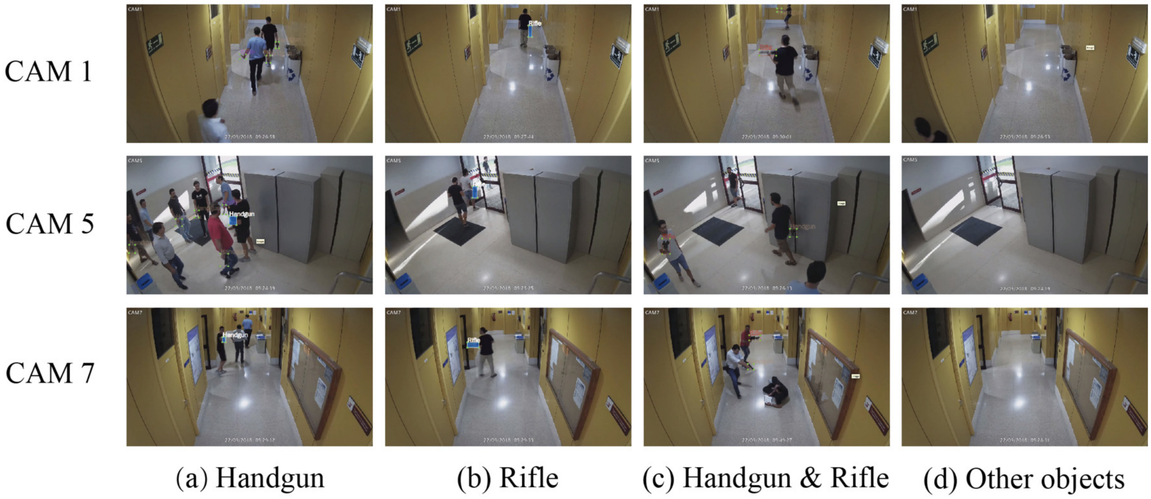

Fig. 4. Some images from the real-world attack simulation database. The dataset contains 5149 images and three scenarios, CAM 1, CAM 5, and CAM 7, as well as handgun and rifle detection objects.

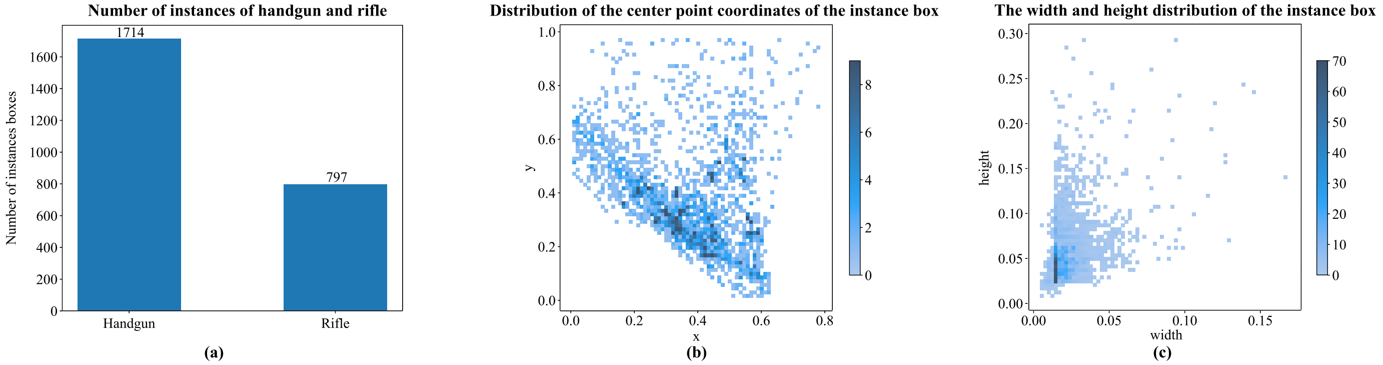

Fig. 5. Distribution of instance boxes for real-world simulation attack datasets.

shows that the majority of instance boxes are less than 0.03 inches in width and 0.02–0.07 inches in height, which is appropriate for small objects.

3.3. Openimg dataset

The synthetic dataset and the real-world simulated attack dataset are both small object datasets under surveillance videos. Therefore, training the model merely on these two datasets is insufficient for the model to learn the real features of the detected objects. This improves the model's ability to detect small objects against complex backgrounds. Experiments in Section 4 prove this observation as well. Consequently, we augment the publicly available dataset with 2582 Openimg images of automatic weapons, including 2277 in the training set (507 handgun images and 1770 rifle images) and 305 in the test set (70 in handgun images and 235 in rifle images). In addition, we apply data augmentation to this incomplete dataset to enable the model to acquire multi-scale and multi-angle object information. Fig. 6 depicts the gun images from a portion of the Openimg public dataset and the augmented images. The distribution of the instance boxes of the Openimg dataset is depicted in Fig. 7. Fig. 7 (b) demonstrates that the majority of instance boxes in the Openimg dataset are located in the image's center. Fig. 7 (c) shows that the distribution of instance boxes in this dataset

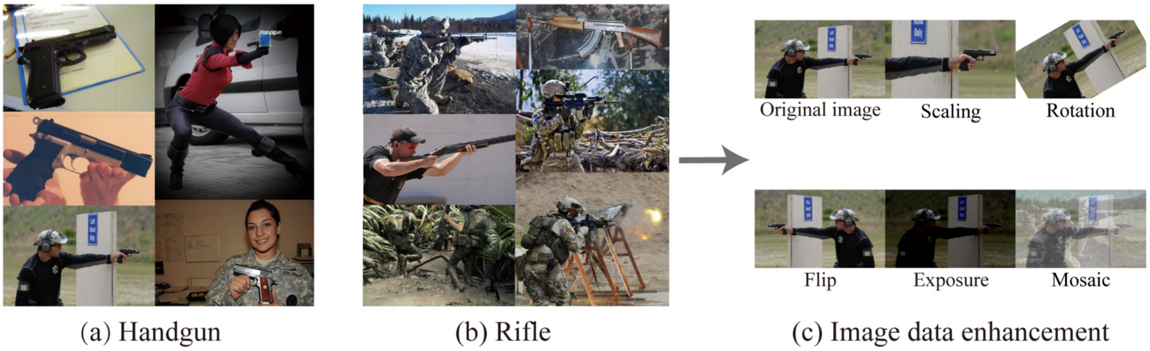

Fig. 6. Several Openimg images. (a) and (b) are example images from the Handgun and Rifle sections, respectively. The data augmentation scheme to expand the dataset is indicated by (c).

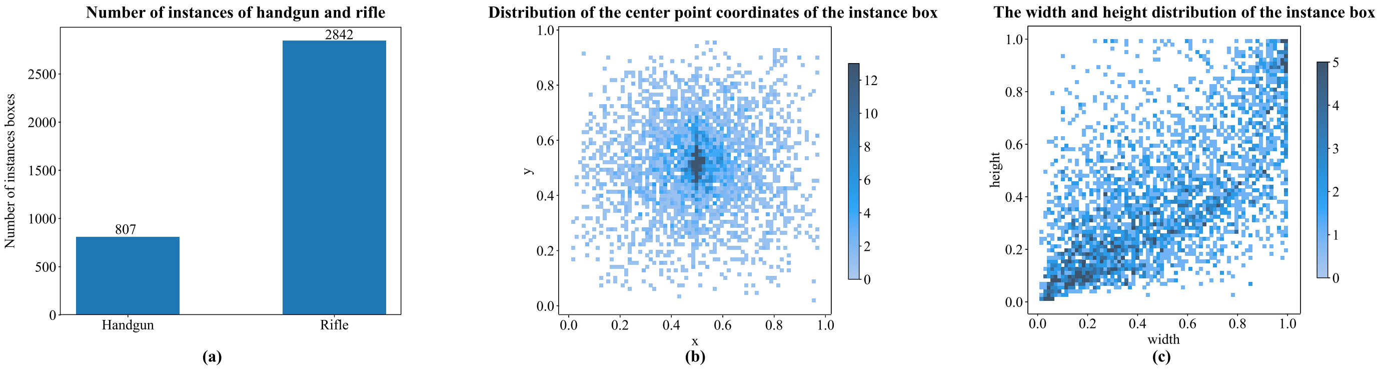

Fig. 7. Distribution of Openimg dataset instance boxes.

is relatively broad, which is consistent with the sizes of actual objects.

4. The proposed detector method

4.1. YOLO v4 model

Unlike two-stage object detection algorithms such as Faster R-CNN, YOLO transforms object detection into a regression problem. YOLO employs a regression approach to directly generate prediction frames and the probabilities for each class without candidate regions, thereby significantly enhancing the speed of detection. YOLO v1 [31] is based on global image information for prediction and has great generalization ability. However, YOLO v1 has drawbacks such as poor positional accuracy and the inability to reliably detect small and dense objects. YOLO v2 [32] improves recall by introducing a feature fusion module and K-means clustering. However, YOLO v2 only employs a single detection head, and small object recognition performance is poor. YOLO v3 [33] resolves the issues that plagued YOLO v1 and YOLO v2. It is a network for object detection that focuses on the detection of small objects and has balanced speed and accuracy. However, YOLO v3 still has drawbacks, such as inaccuracy in recognizing the positions of objects. The YOLO v4 (Bochkovskiy et al. [17]) network is an evolution of the YOLO v3 algorithm for single-stage object detection. YOLO v4 utilizes techniques such as mosaic data augmentation, mish activation function, cross-stage partial connections (CSP), and SPP (spatial pyramid pooling) to achieve more accurate and faster results than YOLO v3. Fig. 8 illustrates the network architecture of YOLO v4.

YOLO v4 outperforms YOLO v3 in terms of accuracy and real-time performance, but its network structure is based on general-purpose object detection, which results in larger detection errors when dealing with small objects in surveillance videos with complex environmental backgrounds. Therefore, we redesign the network structure to enhance the detection accuracy of the model for small objects while maintaining real-time performance.

4.2. Improved YOLO v4 for small object

4.2.1. SCSP-ResNet

YOLO v4 utilizes five cascades of CSP-ResNet53 at various scales in the backbone to gradually extract high-dimensional feature information from the input image. Fig. 9 shows the structure of the YOLO v4 backbone. Each CSP-ResNet structure consists of two major parts, one of which extracts deep features through n-layer ResNet and the other part extracts shallow features through cascaded convolutional neural networks. If x_1 is the input feature map and F_1 is the output feature map, then the feed-forward pass of the CSP-ResNet can be represented by Equation (1):

$$\begin{aligned} w_1 &= x_1 * B_1 \\ F_1 &= w_1 * P_1 \\ F_2 &= w_1 * P_2 * R_n \\ F &= F_1 \circ F_2 \end{aligned} \tag{1}$$

where $*$ is the convolution operation, B_1 is the base layer convolution layer, w_1 is the feature matrix of input features x_1 and B_1 after convolution calculation, P_1 is the convolution layer of CSP-ResNet Part 1, P_2 is the convolution layer of CSP-ResNet Part 2, R_n is the ResNet module with n cascades, and $\circ$ is the feature map stitching operation, which combines multiple feature matrices of the same dimension into a new feature matrix by concatenation. The residual structure of CSP-ResNet optimizes the repetitive gradient information in the network and reduces the computational effort. In addition, CSP-ResNet integrates the feature maps at the beginning and the end of the network phase, which improves the

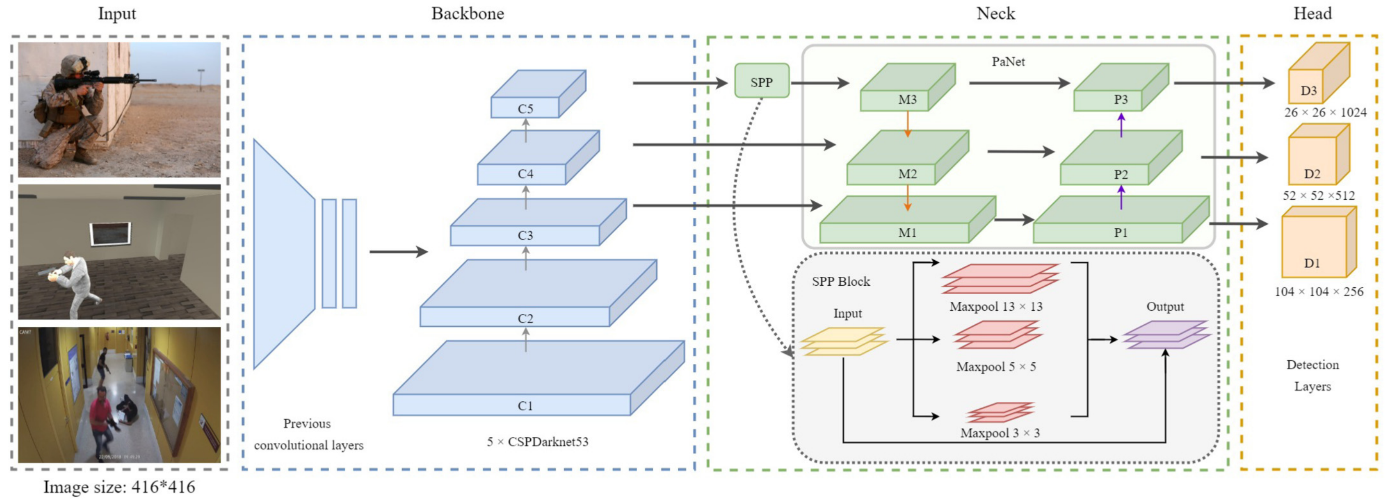

Fig. 8. The network architecture of YOLO v4. The Backbone of YOLO v4 utilizes five different scales of CSP-Darknet53, the Neck section employs SPP and PaNet modules, and the Head section detects extracted features via three different scales of YOLO layers.

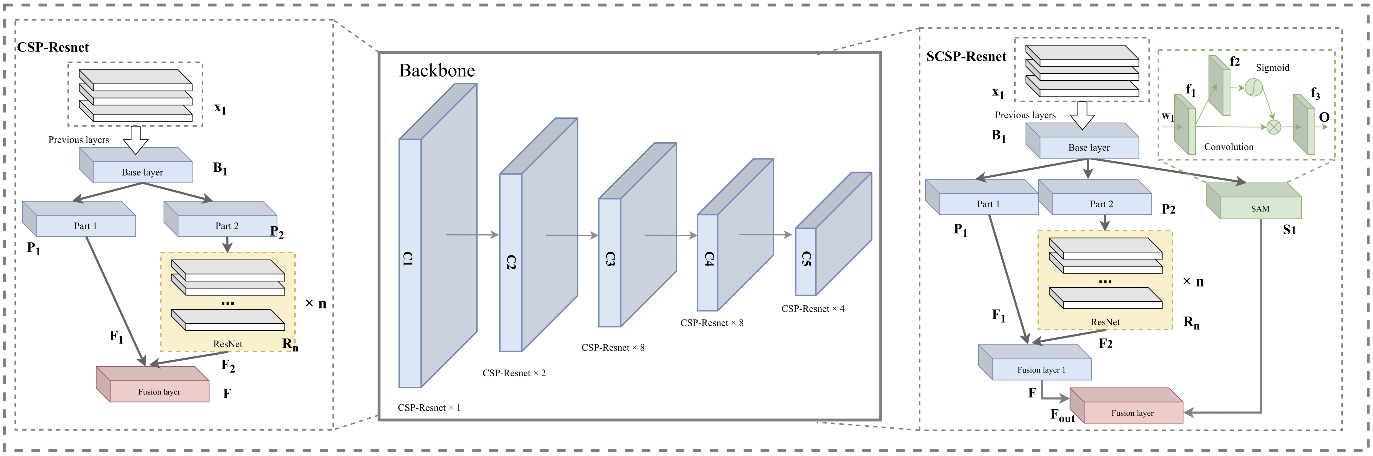

Fig. 9. The backbone of YOLO v4 consists of four CSP-ResNet of different scales. According to the number of ResNet in CSP-ResNet. CSP-ResNet can be divided into CSP-ResNet × 1, CSP-ResNet × 2, CSP-ResNet × 4, CSP-ResNet × 8, etc. The C1 layer of Backbone is CSP-ResNet × 1, C2 layer is CSP-ResNet × 2, C3 and C4 layers are CSP-ResNet × 8, and C5 layer is CSP-ResNet × 8. The right side shows the modified SCSP-ResNet structure.

gradient variability of the network and thus provides excellent performance when dealing with multi-class and large scale objects. When convolutional neural networks detect small objects, several down-sampling operations are applied to expand the perceptual field of the model in order to detect small objects more effectively, and the features are down-sampled to reduce the feature map size, which causes the feature map to lose spatial information.

Therefore, SCSP-ResNet is proposed to supplement the spatial information of feature maps by SAM. SCSP-ResNet augments CSP-ResNet with an enhanced Spatial Attention Mechanism (SAM) module, which improves the model's focus on key features in candidate regions. We propose SAM to first extract the original features of the base layer input, then extract the spatial information of the features by utilizing a convolution layer and a Sigmoid activation function, followed by a fusion layer with CSP-ResNet features for channel fusion, which increases the spatial information of the output features. Equation (2) and Equation (3) describe the forward pass operations of SAM and SCSP-ResNet, respectively.

$$O\left(I\right) = \left[I * f_1^{(m\times m)}; I * f_1^{(m\times m)} * f_2^{(n\times n)} \left(\frac{1}{1+e^{-x}}\right)\right] * f_3^{(q\times q)} \tag{2}$$

where I is the input feature matrix. f_1, f_2 and f_3 are the three cascaded convolutional layers of the SAM module. f_1 and f_2 have a convolutional kernel size of $m \times m$, and f_3 has a convolutional kernel size of $n \times n$. O is the output feature matrix.

$$F_{out} = F \circ S_1 \tag{3}$$

where S_1 is the output feature matrix of w_1 after SAM layer operation and F is the feature matrix of SCSP-ResNet output.

4.2.2. *Receptive field enhancement module*

The backbone of YOLO v4 employs multiple cascaded CSP-ResNet to extract model features with multiple down-sampling, and thus the feature maps of the network's deeper layers have a greater dimension. Multiple down-sampling suppresses a portion of the background information and has less effect on the object region applied to large scale objects. However, for small-scale objects, multiple down-sampling results in the loss of feature map information and a decrease in the model's detection performance for small objects.

In a convolutional neural network, the receptive field can determine the region size of the output feature map of the convolutional layer that corresponds to the input layer. In the field of object detection, RFB (Receptive Field Block) [34] and TridentNet

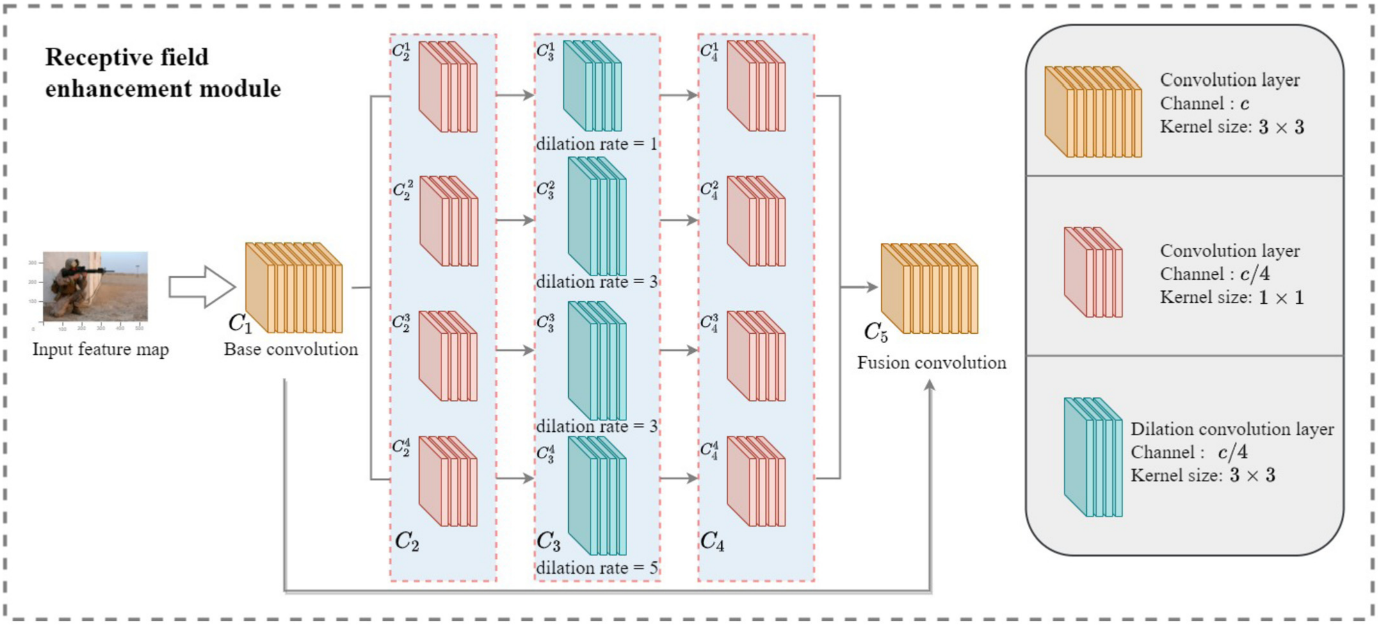

Fig. 10. Structure of the receptive field enhancement module. C_1 is the base convolution layer, which preprocesses the input feature maps. C_2 and C_4 are the convolution layers of 1×1. These two layers can both reduce the number of parameters of the model and achieve cross-channel information integration by adjusting the number of channels of the feature map. C_3 is the convolution layer with three different scales of expansion to obtain multi-scale perceptual fields by adjusting the expansion factor. C_5 is the feature fusion layer with convolution kernel size 3×3.

[35] enhance model performance by expanding the model's perceptual field in order to improve the model's ability to extract features. Multiple down-sampling operations can be applied to obtain a multi-scale receptive field and acquire effective implicit spatial information, allowing for more accurate detection of small objects against complex backgrounds. Fig. 10 depicts the design of the receptive field enhancement module, inspired by GoogLeNet [36] and RFB [34].

The Receptive Field Enhancement module adopts the GoogLeNet architecture, and preprocesses the input feature maps in the C_1 and C_2 layers. Dilated convolution with different dilation factors is adopted in layer C_3 to obtain higher resolution features. Dilated convolution takes both the convolutional filtering function of regular convolutional layers and the generalization effect of pooling layers, which does not reduce the feature map size as the stride increases. The dilation convolution supports exponentially expanding acceptance domain, and the operation of dilation convolution in the model does not lose resolution and coverage. Let F_{dia} be a discrete function satisfying $\mathbb{Z}^2 \rightarrow \mathbb{R}$, $\Delta_r = [-r, r]^2 \cap \mathbb{Z}^2$, and let k_f be a discrete filter of size c satisfying $\Delta_r \rightarrow \mathbb{R}$, and let l be the expansion coefficient of the expansion convolution, then the expansion convolution operator $*_l$ can be defined as Equation (4).

$$(F_{dia} *_l k_f)(\mathrm{p}) = \sum_{\mathrm{s}+l\mathrm{t}=\mathrm{p}} F_{dia}(\mathrm{s}) k_f(\mathrm{t}) \tag{4}$$

When the dilation factors $*_l$ is 1, it is a regular convolution operation. From Equation (4), it can be observed that the size of the receptive field grows exponentially and the parameters number of the model grows linearly. The extracted multi-scale receptive field features are downscaled in layer C_4 to further reduce the parameters number. Finally, the multi-scale feature maps are fused at C_5 to obtain multidimensional fine features.

4.2.3. F-PaNet

PaNet [17], [37] is implemented in the Neck section of YOLO v4. Comparatively to the Feature Pyramid Network (FPN) of YOLO v3, PaNet facilitates the flow of feature information and shortens the information path between the bottom and top layers by means of bottom-up path enhancement. Both YOLO v3 and YOLO v4 have three YOLO layers in the Head, which generate different-scaled feature maps. On CCTV, firearms are depicted as miniature objects with little size variation. With three different scales of YOLO layers, a number of extremely small objects will be lost during the down-sampling process from Neck to Head. Furthermore, Employing the same small-scale YOLO layer throughout the Head section reduces the model's robustness.

Therefore, we propose F-PaNet, which aims to improve the model's detection performance for small objects without compromising the model's robustness. The structure of PaNet and F-PaNet is illustrated in Fig. 11. The bottom network contains spatial data, while the top network contains semantic data. In the information flow from the bottom to the top layer, PaNet's down-sampling will remove some spatial information, which prevents the detection of objects with sparser feature information. As the number of network layers increases, it becomes more difficult to access precise spatial information for features from the lowest layer to the highest layer. In addition, since each prediction layer is based on a feature-level feature pooling network, the dropped information may contribute to the model's final detection. F-PaNet expands upon PaNet by providing additional opportunities to collect diverse data. First, the feature flow path is shorter. F-PaNet fuses the three feature layers at the bottom and top levels of the Neck segment independently, thereby improving the mobility of spatial and semantic data. Second, the fusion of feature information from various feature levels provides diverse information for the final detection layer, which helps in the localization and identification of small objects in complex scenarios.

4.2.4. Other enhancements

Model Purning To maintain the model's accuracy while minimizing the number of parameters, we employ a structured model pruning scheme. Structured pruning is a flexible pruning scheme that includes the pruning of the number of convolutional channels and the number of convolutional filters. This method completely prunes the convolution kernel and feature map, which reduces the number of model parameters, simplifies the operation, and makes

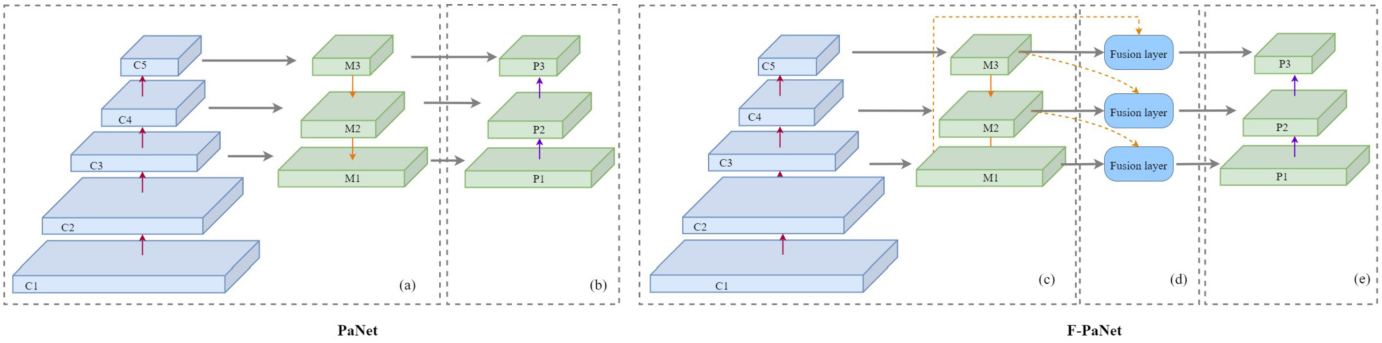

Fig. 11. Illustration of PaNet and F-PaNet. (a),(c) Feature pyramid network module. (b),(e) Bottom-up path augmentation. (d) Feature Fusion and Information Flow Enhancement Module.

Table 2
Details of model pruning.

Neck Module	Convolutional layer/ Stride	Original convolution layers		Modified convolution layers	
		Filter shape	Parameters (GFLOP/s)	Filter shape	Parameters (BFLOP/s)
P1	Conv 1 / s1	1*1*128	0.177	1*1*128	0.177
	Conv 2 / s1	1*1*128	0.177	1*1*128	0.177
	Conv 3 / s1	3*3*256	1.595	3*3*256	1.595
	Conv 4 / s1	1*1*128	0.177	1*1*256	0.177
	Conv 5 / s1	3*3*256	1.595	3*3*256	1.595
	Conv 6 / s1	1*1*128	0.177	–	–
P2	Conv 1 / s2	3*3*256	0.399	3*3*256	0.399
	Conv 2 / s1	1*1*256	0.177	1*1*256	0.177
	Conv 3 / s1	3*3*512	1.595	3*3*512	1.595
	Conv 4 / s1	1*1*256	0.177	1*1*256	0.177
	Conv 5 / s1	3*3*512	1.595	1*1*512	1.595
	Conv 6 / s1	1*1*256	0.177	–	–
P3	Conv 1 / s2	3*3*512	1.595	–	–
	Conv 2 / s1	1*1*512	0.399	–	–
	Conv 3 / s1	3*3*1024	1.595	–	–
	Conv 4 / s1	1*1*512	0.177	1*1*1024	0.177
	Conv 5 / s1	3*3*1024	1.595	1*1*512	0.177
	Conv 6 / s1	1*1*512	0.177	1*1*1024	0.177
Total parameters		–	13.556	–	8.195

the model deployable. In this paper, the filters in the Neck section of YOLO v4 are pruned in accordance with the scheme presented in Table 2. Table 4 shows the experimental results before and after model pruning.

Anchor frame size matching based on k-means YOLO v4 utilizes the K-means clustering algorithm to obtain nine anchor frames and apply them to three different scales of YOLO layers in the Head portion of the network in order to localize and classify detection objects. YOLO v4 anchor frames are intended for large and medium-sized objects in the PASCAL VOC dataset. Therefore, it is not applicable to the detection of small objects. Equation (5) and Equation (6) are the calculation of the distance of K-means clustering algorithm in YOLO v4.

$$d(box, central) = 1 - IoU(box, central) \tag{5}$$

$$IoU = \frac{\left|B \cap B^{gt}\right|}{\left|B \cup B^{gt}\right|} \tag{6}$$

where, *box* is the size of the prediction frame, *central* is the clustering center, B is the area of the real frame, B^{gt} is the area of the prediction frame, and IoU (Intersection Over Union) is the intersection ratio of the prediction frame and the real frame. When IoU is applied to reflect the detection effect of the prediction frame and the true frame, if the two frames do not intersect, IoU=0, indicting it cannot correctly reflect the actual distance between the prediction frame and the true frame. To address the shortcomings of IoU, Hamid Rezatofighi et al. [38] proposed $GIoU$ (Generalized IoU), and expressed by Equation (7).

$$GIoU = IoU - \frac{\left|C - B \cup B^{gt}\right|}{|C|} \tag{7}$$

In Equation (7), C is the smallest box area that contains both B and B^{gt}. $GIoU$ could not only measure the overlap between the prediction frame and the real frame, but also correctly measure the non-overlap case, which can better reflect the overlap between the prediction frame and the real frame.

Therefore, in order to better measure the overlap between the prediction frame and the real frame, we adopt $GIoU$ instead of IoU. Equation (8) is the improved distance calculation method of K-means clustering algorithm.

$$d_{GIoU}(box, central) = 1 - GIoU(box, central) \tag{8}$$

5. Experimental results

5.1. Experimental setup

We have trained the model on Geforce RTX-2080Ti to validate the performance of the model proposed in this paper. The YOLO series algorithm is trained on the Darknet platform, which is an open-source object detection framework written in C++ with excellent support for the YOLO series algorithm. Faster R-CNN and Retinanet are trained on the detectron 2 platform [30], which is an open source object detection and instance segmentation platform provided by the Facebook research team and supports a wide

Table 3
Division scheme of training and test sets.

Data set division scheme	Dataset	Abbreviations	Number of images		Proportion
Train	Synthetic dataset	STR	1840	7925	77.5%
	Real-world simulation attack dataset	RTR	3808		
	Openimg	OTR	2277		
Test	Synthetic dataset	STE	660	2306	22.5%
	Real-world simulation attack dataset	RTE	1341		
	Openimg	OTE	305		

Table 4
Experimental results before and after model pruning.

	Parameters (BFLOP/s)	Size of the weight file	mAP (mean Accuracy Precision)	FPS (Frame Per Second)
Original model	59.570	256 Mb	**74.38%**	76.7
Pruned model	**53.224**	**190 Mb**	61.2%	**86.6**

Table 5
Model performance after modifying the anchor frame.

	Anchor frame size	TP	FP	FN	mAP	AP (Handgun)	AP (Rifle)	F1 score	recall	precesion
Original model	(12, 16) (19, 36) (40, 28) (36, 75) (76, 55) (72, 146) (142, 110) (192, 243) (459, 401)	1202	219	391	80.26%	71.5%	89.02%	0.8	0.75	0.85
Improved model	(11, 23) (38, 34) (60, 73) (155, 68) (81,166) (233,124) (162,230) (310,194) (359,334)	1226	230	367	81.75%	74.61%	88.9%	0.8	0.77	0.84

range of State of the Art (SOTA) algorithms that are widely applied in computer vision research projects and production applications.

Larger size images will input more detailed information to the network, which can improve the model detection accuracy, but it would lose the real-time performance of the model. The improved model adopts the hyper-parameters of YOLO v4. In addition, we set the image input size to 416 × 416 for all experiments. Furthermore, we adopt random gradient descent (SGD) optimizer with batch size set for 64, momentum set for 0.9, iterations set for 10000, initial learning rate set for 0.00261, and the learning rate decreases at training up to 8000 and 9000 times, respectively, with the learning rate 0.1 of the initial learning rate.

Three datasets are employed for our experiments, which are detailed and abbreviated as shown in Table 3. The annotation files of the dataset anchor frames are in *.xml* format. For training in Darknet and detectron 2 [30], we convert the files in this format to *.txt* files supported by darknet and *.json* files in MS COCO format. All experiments are performed with an input image size of 416 × 416.

The experimental results are evaluated by using the following metrics.

1. Detection Precision. Including mean Average Precision (mAP), Precision per category (AP), small Average Precision (APs), medium Average Precision (APm), large Average Precision (APl), can directly reflect the accuracy of the model detection, which is our main challenge goal.

2. The number of parameters (BFLOP/s), which can reflect the size of the model parameters, and the computational power of the required hardware.

3. Weight file size, which evaluates the disk space occupied by the model.

4. Inference time and FPS (Frame Per Second) for a single image (which can reflect the real-time nature of the model, which is our main challenge goal).

5. True Positives (TP) are the number of correct predicted answers, False Positives (FP) are the number of wrong predictions of other classes into this class, and False Negatives (FN) are the number of predictions of this class into other classes.

6. Recall and Precision. Recall is the proportion of positive cases in the total number of predicted cases, which reflects the accuracy rate of the model. Precision is the proportion of positive samples in the positive cases determined by the classifier, which reflects the accuracy rate of the model. The equations for calculation are as Equation (9).

$$\begin{aligned} precision_k &= \frac{TP}{TP+FP} \\ recall_k &= \frac{TP}{TP+FN} \end{aligned} \tag{9}$$

7. F1 score, which is often used to measure classification problems. It is the summed average of the precision and recall rates.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{10}$$

Fig. 12 shows the experimental setup for six sets of experiments in this paper. In Section 5.2, we validate the efficacy of model pruning and anchor frame modification. In Section 5.3, six sets of experiments are designed to determine whether synthetic data can enhance the performance of a model. In Section 5.4, the optimal model training scheme is determined employing transfer learning. The ablation experiments are displayed in Section 5.5, and the comparison experiments are illustrated in Section 5.6.

5.2. Experiment of other enhancements

The comparison of experimental results before and after model pruning is shown in Table 4. After pruning the model, the parameters are reduced by 6.346 BFLOP/s and the weight size decreases by 66 Mb. Compared to the result of before pruning, the FPS improves by 13%. However, the model's mAP decreases by 13.18%, and there is still considerable room for growth.

Taking the input image size of 416*416 as an example, the nine selected anchor frames are (11, 23) (38, 34) (60, 73) (155, 68) (81,166) (233,124) (162,230) (310,194) (359,334) after we adopt the modified K-means clustering algorithm for distance calculation. Table 5 shows the comparison of model related parameters before and after modifying the anchor frame size.

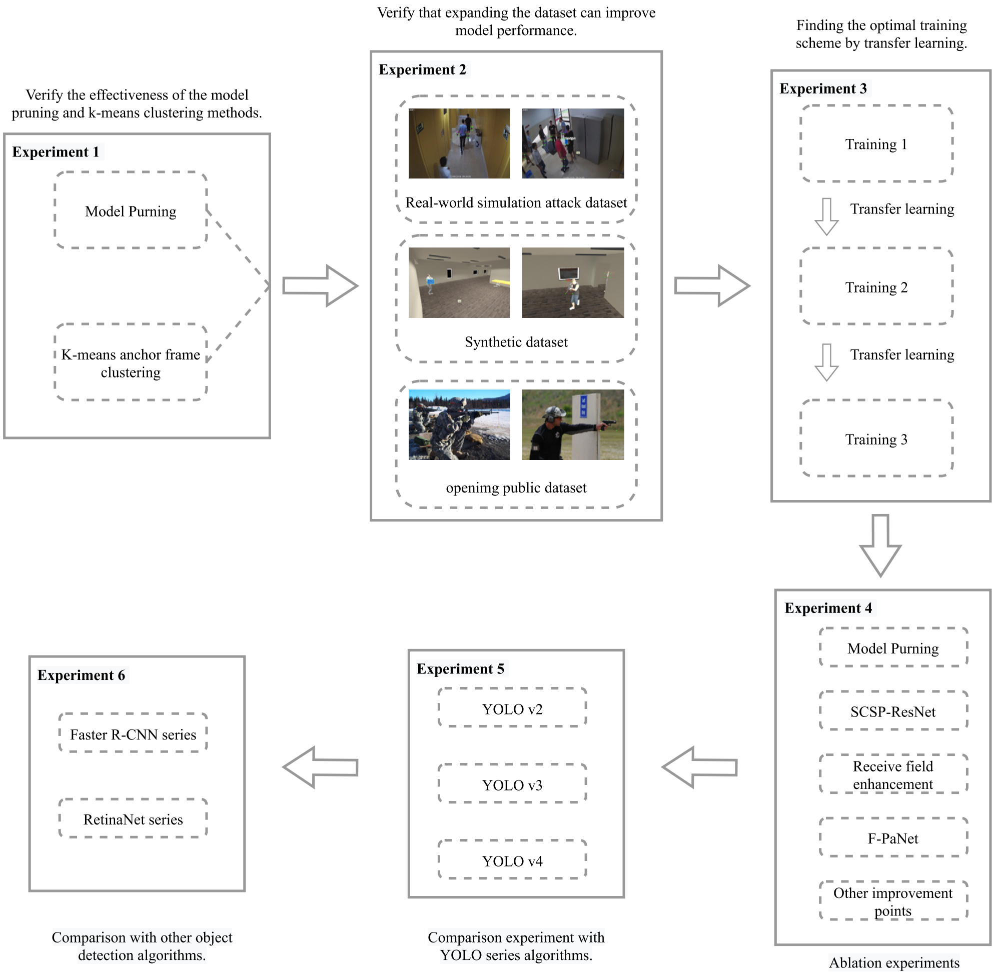
Verify that expanding the dataset can improve model performance.
Finding the optimal training scheme by transfer learning.
Verify the effectiveness of the model pruning and k-means clustering methods.
Experiment 1
Model Purning
K-means anchor frame clustering
Experiment 2
Real-world simulation attack dataset
Synthetic dataset
openimg public dataset
Experiment 3
Training 1
Transfer learning
Training 2
Transfer learning
Training 3
Experiment 4
Model Purning
SCSP-ResNet
Receive field enhancement
F-PaNet
Other improvement points
Ablation experiments
Experiment 5
YOLO v2
YOLO v3
YOLO v4
Comparison experiment with YOLO series algorithms.
Experiment 6
Faster R-CNN series
RetinaNet series
Comparison with other object detection algorithms.

Fig. 12. The flow of the experiments.

5.3. Experiments with different datasets

These experiments employ our modified YOLO v4 as the baseline model and a Table 6 to divide the dataset in order to train the model and investigate whether the synthetic dataset could improve the performance of the model. All experiments are conducted on the Darknet platform, without transfer learning or pretrained weights, and the initial parameters are maintained. The experimental results are shown in Table 6. Experiments 1 to 6 are trained on YOLO v4, and experiments 7 to 12 are trained on the improved YOLO v4. The experimental results presented in Table 6 demonstrate that the improved YOLO v4 model outperforms the original algorithm in both the synthetic dataset with simple scenarios and the real-world attack simulation dataset with complex backgrounds.

Experiment 1 and Experiment 7 display the results of training and testing using only the simulated attack dataset, which contains 5149 images. The mAP of the improved model is only 66.75% due to the lack of pre-training weights and the addition of more distractors (fire hydrant, garbage bin, scene light and dark contrast, etc.) and negative samples (images without the object to be detected). The anchor box size of handguns is typically smaller than that of rifles, making handguns more difficult to detect. In all experiments, therefore, the AP values of the Rifle are greater than those of the Handgun. In Experiment 2 and Experiment 8, 2500 synthetic images are applied for training. Since the scenes of the training and test sets are more similar as a result of the simplified scenes of the synthetic images and the continuous video frame extraction utilized to generate all datasets, the accuracy rate is higher. Experiments 3 and 9 are designed due to the eventual application of the model in the real world. Experiments 3 and 9 are trained with the synthetic dataset and evaluated with the actual dataset. Experiment 9 demonstrates that the model trained on the synthetic dataset cannot be directly applied to the real world. Experiments 4 and 10 utilize only images from the real world, including the simulated attack dataset and the Openimg public dataset. Extending the 2582 real-world weapon datasets improves the overall performance of the model compared to Experiment 7. Among them, mAP is enhanced by 3.72%, and the model's check accuracy is the same as in Experiment 7, but recall is enhanced by 0.05 percent. Experiments 5 and 11 are trained with synthetic and simulated attack data sets, respectively. The experimental results demonstrate that synthetic data can improve the performance of the model, and the model's overall performance is enhanced compared to several previous experiments. Experiments 6 and 12 add the Openimg dataset, and the resulting model has the highest TP value and the second-highest precision and recall after experiment 8. This portion of the experiment demonstrates that adding synthetic images to the dataset improves the performance of the model. Consequently, the addition of synthetic images to the dataset is effective.

5.4. Training scheme

In this experiment, the dataset is divided into two parts: training set and test set, and each part has three different categories

Table 6
Experimental results in different datasets.

Experiment number	model	Training set	Test set	TP	FP	FN	mAP	AP (Handgun)	AP (Rifle)	F1 score	Recall	Precision
1	YOLO v4	RTR	RTE	347	176	302	56.94%	54.06%	59.81%	0.59	0.53	0.66
2		STR	STE	706	51	32	96.66%	96.35%	97.96%	0.94	0.96	0.93
3		STR	RTE+OTE	499	246	546	52.59%	46.44%	58.74%	0.56	0.48	0.67
4		RTR+OTR	RTE+OTE	673	281	372	65.51%	61.55%	69.46%	0.67	0.64	0.71
5		STR+RTR	STE+RTE	434	168	215	70.02%	67.05%	73.00%	0.69	0.67	0.72
6		STR+RTR+OTR	STE+RTE+OTE	1286	313	474	74.38%	70.17%	78.59%	0.77	0.73	0.81
7	Improved YOLO v4	RTR	RTE	415	148	234	66.75%	62.18%	71.33%	0.68	0.64	0.74
8		STR	STE	704	28	34	97.14%	96.94%	97.34%	0.96	0.95	0.96
9		STR	RTE+OTE	390	127	465	54.74%	45.75%	63.73%	0.57	0.46	0.75
10		RTR+OTR	RTE+OTE	721	247	324	70.47%	66.41%	74.53%	0.74	0.69	0.74
11		STR+RTR	STE+RTE	474	131	175	75.7%	72.84%	78.56%	0.76	0.73	0.78
12		STR+RTR+OTR	STE+RTE+OTE	1226	230	367	81.75%	74.61%	88.9%	0.8	0.77	0.84

Table 7
Experimental results of the research on the optimal training scheme.

Experiment number	Training 1	Training 2	Training 3	TP	FP	FN	mAP	AP (Handgun)	AP (Rifle)	F1 score	Recall	Precision
1	O	S	R	477	145	172	76.36%	74.4%	78.32%	0.75	0.73	0.77
2		S+R	–	954	191	433	74.15%	63.59%	84.72%	0.75	0.69	0.83
3	S	O	R	472	125	177	74.51%	73.86%	75.15%	0.75	0.73	0.79
4	R	O	STR+STE	714	28	24	97.55%	96.58%	98.53%	0.96	0.97	0.96
5	S+R	–	–	832	162	555	66.58%	53.75	79.41%	0.7	0.6	0.84
6	S+R+O	–	–	1226	230	367	81.75%	74.61%	88.9%	0.8	0.77	0.84

(synthetic dataset, simulated attack dataset, and Openimg dataset). Fig. 13 shows the distribution of the dataset. Fig. 13 (a) shows the overall information of the dataset with 3407 handgun images, 5520 rifle images and 1304 negative sample images (without any object). Fig. 13 (b) shows the division scheme of the training set and the test set, the training set includes 7925 images and the test set includes 2306 images. Fig. 13 (c) and Fig. 13 (d) show the distribution of the images number of handgun and rifle in the training and test sets, respectively.

In this experiment, the dataset is divided into two parts: the training set and the test set, with each part containing three distinct categories (synthetic dataset, simulated attack dataset, and Openimg dataset). The distribution of the data set is depicted in Fig. 13. Fig. 13 (a) depicts the overall data of the dataset, which consists of 3407 handgun images, 5520 rifle images, and 1304 negative sample images (without any object). Fig. 13 (b) depicts the division scheme of the training set and the test set, with the training set containing 7925 images and the test set containing 2306 images respectively. Fig. 13 (c) and (d) illustrate the distribution of the number of handgun and rifle images in the training and test sets, respectively.

We propose to employ transfer learning to research the optimum data training scheme. As in Section 5.3, our enhanced model is utilized as a training model on various datasets. Once the model of the object detection algorithm has been constructed, the number of model parameters, the size of the weight file, and the real time cannot be altered. Therefore, we only select parameters that contribute to the model's accuracy. The real-time performance is contrasted in Sections 5.5 and Section 5.6. The experimental outcomes are displayed in Table 7.

Experiment 1 starts with pre-training on the Openimg dataset, followed by training on the synthetic dataset using transfer learning, and finally on the simulated attack dataset. After applying transfer learning, the mAP of the model is improved by 9.61% over training on the simulated dataset only (see Experiment 1 in Section 5.3 for details), and the smaller scale Handgun and Rifle are also improved by 12.22% and 6.99%, respectively, which are significant performance improvements for small object detection. In Experiment 2, pre-training is first performed on the Openimg dataset, and then the synthetic dataset is mixed with the simulated attack dataset afterwards. Compared with Experiment 1, the TP value of the model is significantly higher than that of Experiment 1 because of the expanded images in the test set. However, the mAP of the model declines slightly, and the detection accuracy of Handgun is 10.81% lower than that of Experiment 1, while the F1 score is consistent with Experiment 1.

Experiment 3 is trained by adopting the order of S → O → R. The F1 score of the model remained the same as Experiment 1 and Experiment 2, and the overall performance did not improve significantly. Experiment 4 employs the order of R → O → S for training. The mAP of the model can reach 97.55% because of the simple background of the synthetic dataset. In addition, these scenes are extracted from consecutive video frames with a high background similarity. But in real scenes, the accuracy of the model is relatively low. The experiments in Section 5.3 have proved the conclusion. Experiment 5 is trained after merging the synthetic dataset and the real-world simulated attack dataset. The experimental results show that there is still more room for improvement in this way of data combination. Experiment 6 merges Openimg, synthetic dataset and real-world simulated attack dataset for training, the model has the highest TP value, and the accuracy of Handgun and Rifle is also optimal. In addition, Recall and Precision also show the effectiveness of this data combination scheme. Therefore, the scheme of Experiment 6 is optimal. The ablation experiments and comparison experiments are trained by this scheme.

5.5. *Ablation experiments*

In this section, we demonstrate the incremental impact of each module on the model. The results of Section 5.5 and Section 5.6 are used to train the experiments using a combination of virtual and real-world data. The experiments are conducted in accordance with our exploration strategy for improving the model's functionality, and the outcomes demonstrate the viability of our suggested areas for improvement. Table 8 displays the outcomes of ablation experiments.

A→B: First, we attempt to develop a basic model for real-time weapon detection. Due to the large number of parameters in YOLO v4, we first prune the convolutional layer of the Neck model. Section 4.2.4 describes the specific model pruning scheme. There is a

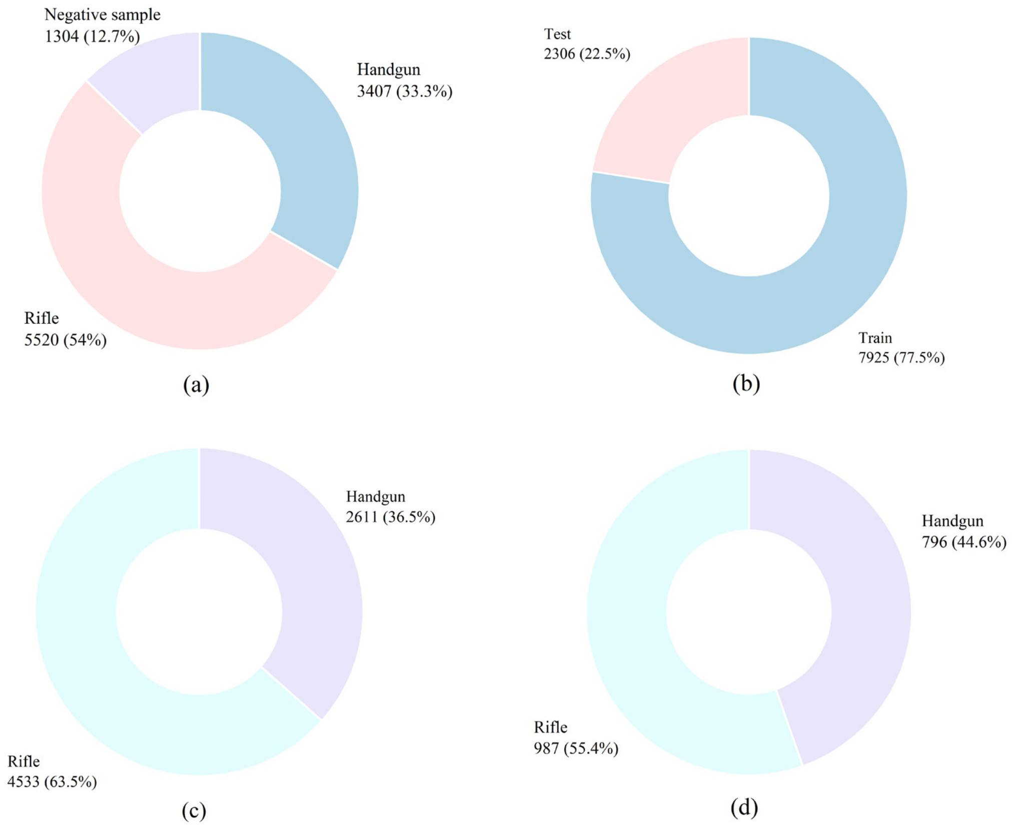

Fig. 13. Distribution of the dataset (including synthetic dataset, simulated attack dataset, and Openimg dataset).

Table 8
Results of ablation experiments.

	Method	mAP	Parameters (GFLOP/s)	Size of the weight file	FPS
A	Darknet 53 YOLO v4	77.3%	59.570	256.0 Mb	76.7
B	A + Model Pruning	61.2%	52.432	190.5 Mb	86.6
C	B + SCSP-ResNet	71.9%	53.224	190.9 Mb	84.2
D	C + Receptive field enhancement	76.6%	54.47	214.7 Mb	80.1
E	D + F-PaNet	80.3%	59.221	252.1 Mb	77.3
F	E + Other improvement points	81.75%	59.221	252.1 Mb	77.1

greater improvement in the number of parameters and weight size in the pruned model. However, it is not possible to extract deeper features from the pruned model network due to the reduced number of layers. Consequently, the model's mAP is significantly diminished.

B→C: We augment the CSP-ResNet component of the model. This module adds spatial information to the model's features, increasing the number of parameters by only 0.792 BFLOP/s and improving the model's mAP by 10.7%. Moreover, the model's FPS decreases by only 2.4% compared to experiment B. The model's overall performance is significantly enhanced.

C→D: Next, we consider using a perceptual field enhancement module deep in the backbone to add more fine features to the high-dimensional feature maps. Compared with experiment C, the accuracy of the model is improved by 4.7% and the number of parameters is increased by 1.246. But the weight size is increased by 23.8 Mb due to the additional 15 convolutional layers of this module. Next, we consider implementing a perceptual field enhancement module deep in the backbone to add more fine-grained features to the high-dimensional feature maps. Compared to experiment C, the model's accuracy has increased by 4.7%, while the number of parameters has increased by 1.246. However, the module's additional 15 convolutional layers result in a weight increase of 23.8 Mb.

D→E: We propose F-PaNet, which is also an effective solution for enhancing the performance of models. Adding the F-PaNet module to the model increases the mAP by 3.7%. The module increases the number of parameters and size of the model's weight, but the model's FPS remains better compared to that of the baseline model.

E→F: It is difficult to improve the mAP of the model without increasing model parameters in order to ensure the model's real-time nature. In this section, we primarily employ the k-means-based anchor frame calculation method described in Section 4.2.4 to enhance the model's ability to locate the anchor frames. Without adding additional parameters or computational effort, the model's performance is improved further.

5.6. Comparison with other object detection algorithms

We have compared the improved model with other object detection algorithms, including the two-stage Faster R-CNN, the single-stage Retinanet, and the YOLO family of algorithms. The YOLO series algorithms are trained on Darknet with batch size set for 64 and iterations set for 8000. The Retinanet and Faster R-CNNs are trained on detectron 2 with batch size set for 24 and iterations set for 33000. The learning rate of all experiments is 0.002, and no pre-training weights are used in any of them. Infer time is the

Table 9
Comparison experiment with YOLO series algorithms.

model	mAP	Parameters (GFLOP/s)	Size of the weight file	Infer time	FPS
YOLO v2	38.94%	29.371	202.7 Mb	5.69 ms	175.7
YOLO v3	67.62%	65.312	246.3 Mb	17.46 ms	57.2
YOLO v4	74.38%	59.570	256.0 Mb	15.10 ms	66.2
Improved YOLO v4	81.75%	59.221	252.1 Mb	14.47 ms	69.1

Table 10
Comparison with other object detection algorithms.

model	mAP (%)	AP (%)	AP75 (%)	APs (%)	APm (%)	APl (%)	Size of the weight file	Infer time
Faster R-CNN-R50-FPN-1x	53.54	28.68	27.744	5.786	23.40	36.02	333.3 Mb	0.11 s
Faster R-CNN-R50-FPN-3x	54.21	28.93	28.52	1.902	23.84	35.61	333.3 Mb	0.15 s
Faster R-CNN-R101-FPN-3x	53.23	29.65	28.97	2.87	24.04	36.58	464 Mb	0.16 s
RetinaNet-R50-FPN-1x	37.80	21.54	21.35	0.58	9.66	36.04	290 Mb	0.10 s
RetinaNet-R50-FPN-3x	38.84	22.11	22.28	1.04	11.32	36.68	290 Mb	0.10 s
RetinaNet-R101-FPN-3x	37.23	21.88	22.35	0.92	10.67	36.89	436 Mb	0.12 s
Improved YOLO v4	81.75	–	45.4%	–	–	–	252.1 Mb	0.019 s

inference time of a single image, which is influenced by the detection object in the image. FPS is the number of video frames per second processed by the model. The experimental results are shown in Table 9.

We first compare the algorithms of the YOLO series. YOLO v2 is an earlier version of the YOLO series algorithm. Since the number of network layers is less, the model has less number of parameters with the best FPS for the YOLO series. However, the accuracy of the model is the worst. YOLO v3 is an improved version of YOLO v2 with optimization schemes such as FPN. The number of parameters is increased by 35.941, but the accuracy of the model is significantly improved. YOLO v4 is our baseline model. Compared with YOLO v3, the algorithm uses improved schemes such as PaNet, CSP-ResNet, etc. The weight size of the model is improved by 9.7 Mb, the video real-time processing power is decreased by 0.6, but the infer time is improved by 2.36 ms, and the accuracy of the model is improved by 6.76%. The FPS of our improved model is reduced by 0.2 compared with YOLO v3 (which is almost negligible in real applications), but the mAP of the model reaches the optimal of YOLO series, and the experimental results prove the effectiveness of our improved scheme.

In addition, we also have compared the improved model with Faster R-CNN and RetinaNet. We train with three different structures of Faster R-CNN and retinanet which are provided by detectron 2 [30], and the experimental structures are shown in Table 10. The experimental results show that it is not the case that the deeper the network structure is, the better the model performance is. The performance of the model with 101 layers of ResNet is inferior to that of ResNet with 50 layers. In addition, benefiting from the excellent real-time performance of the YOLO algorithm, the inference time of our proposed improved YOLO v4 is only 17% of that of the Faster R-CNN, and the mAP is much better than that of the Faster R-CNN and Retinane. Fig. 14 shows the detection effect.

On the Openimg test images, the detection results reveal that YOLO v4, Faster R-CNN, and RetinaNet are all susceptible to false and missing detections. Specifically, YOLO v4 fails to correctly identify the handgun in the first test image and identifies the headset as a rifle in the second image, which also fails to identify the actual rifle. On the first image, both Faster R-CNN and RetinaNet detect false objects. Due to the simplified scenario presented by the synthetic dataset, all algorithms accurately detect the object. On the dataset of simulated real-world attacks, YOLO v2 and Faster R-CNN suffer from severe misses, and neither detects any weapons in the scenario. YOLO v3 detects only one of the four objects, while RetinaNet detects two but erroneously identifies the wall as a weapon. YOLO v4 and our current improved algorithm detect all weapons in the scene with precision. Our algorithm clearly has the advantage in terms of detection accuracy.

6. Conclusion

Due to their diminutive size, low resolution, and susceptibility to complex backgrounds, detecting small objects with CCTV is a challenging task. To address these issues, we propose a CCTV weapon detection system based on the YOLO algorithm that operates in real time. The system is trained to use a dataset that merges synthetic and real images of weapons. The experimental results indicate that the synthetic dataset can enhance the performance of the model. Moreover, we propose an optimization scheme for the real-time detection of small objects against complex backgrounds for CCTV surveillance. In particular, we augment the spatial attention module in the backbone network in order to suppress complex background information and increase the spatial dimensionality of information for objective features. Moreover, we employ multi-scale dilation convolution in the Neck region to provide the model with a greater number of objectively fine, high-dimensional details. And we also propose the F-PaNet module, which improves the flow of position information in the deep Neck portion of the network and supplies diverse feature information for the final detection layer. In section 4, we examine the improved scheme and the training scheme, and the experimental results demonstrate the effectiveness of the improved scheme. Our proposed model enhancements are applicable to other computer vision tasks as well. However, whether synthetic datasets can improve model performance indefinitely and whether synthetic data with complex backgrounds are more effective in improving model performance are not thoroughly investigated. In the future, we will continue to investigate the effect of synthetic data on object detection tasks by blending and training real-world images with synthetic images of varying scales, and evaluating the experimental results. In addition, our future work will apply the synthetic data approach to image segmentation, object tracking, and other areas to determine if the approach can be applied to other computer vision tasks.

CRediT authorship contribution statement

Guanbo Wang: Conceptualization, Software, Writing – original draft. **Hongwei Ding:** Supervision, Writing – original draft. **Mingliang Duan:** Data curation, Visualization. **Yuanyuan Pu:** Data curation, Investigation. **Zhijun Yang:** Software, Supervision, Validation. **Haiyan Li:** Validation, Writing – review & editing.

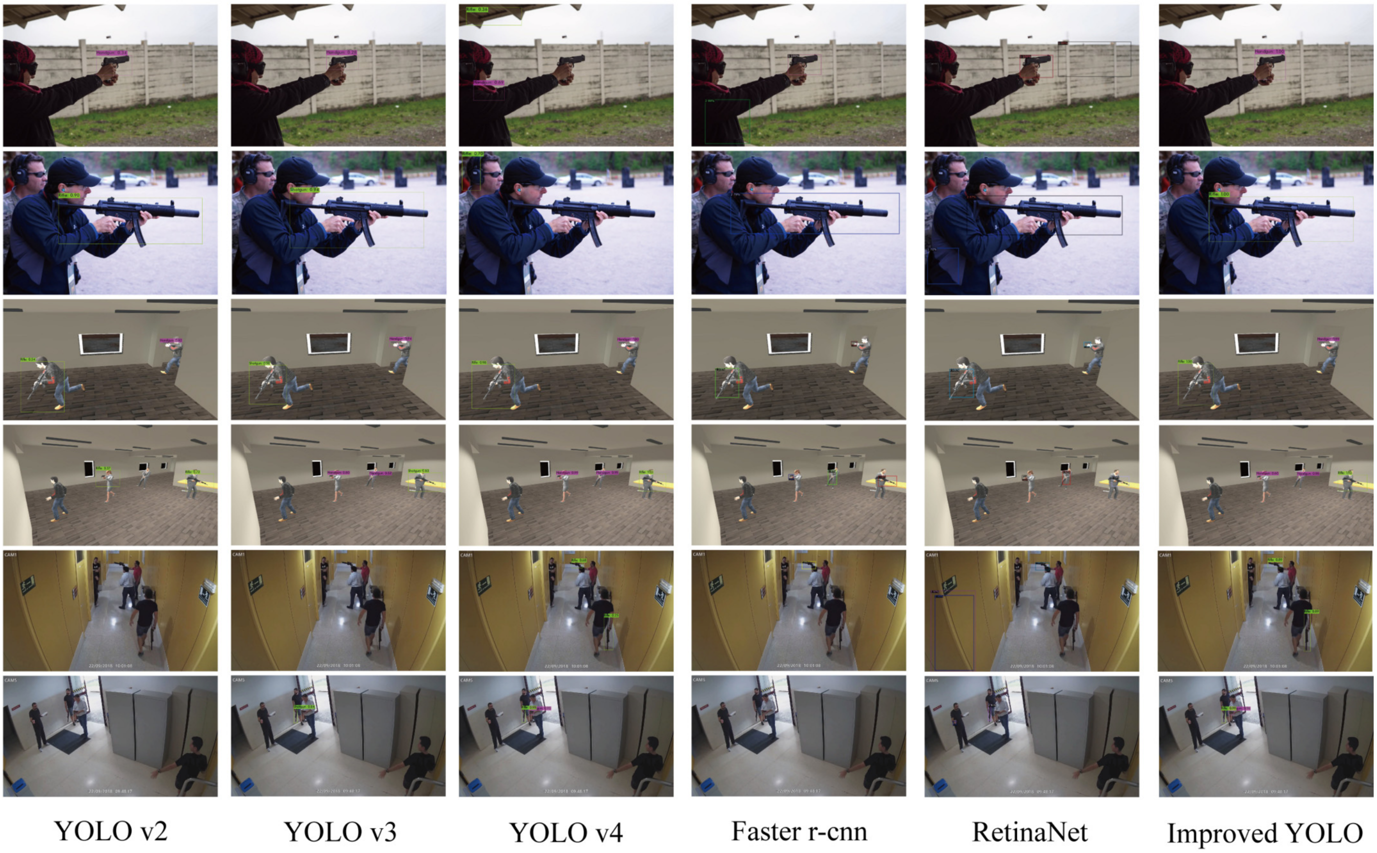

Fig. 14. Detection results of different algorithms.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] F. Enríquez, L.M. Soria, J.A. Álvarez-García, F.S. Caparrini, F. Velasco, O. Deniz, N. Vallez, Vision and crowdsensing technology for an optimal response in physical-security, in: International Conference on Computational Science, Springer, 2019, pp. 15–26.

[2] J.L.S. González, C. Zaccaro, J.A. Álvarez-García, L.M.S. Morillo, F.S. Caparrini, Real-time gun detection in cctv: an open problem, Neural Netw. 132 (2020) 297–308.

[3] G.R. Taylor, A.J. Chosak, P.C. Brewer, Ovvv: using virtual worlds to design and evaluate surveillance systems, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[4] Y. Wang, W. Liang, J. Shen, Y. Jia, L.-F. Yu, A deep coarse-to-fine network for head pose estimation from synthetic data, Pattern Recognit. 94 (2019) 196–206.

[5] S.A. Velastin, B.A. Boghossian, M.A. Vicencio-Silva, A motion-based image processing system for detecting potentially dangerous situations in underground railway stations, Transp. Res., Part C, Emerg. Technol. 14 (2) (2006) 96–113.

[6] S. Akcay, T. Breckon, Towards automatic threat detection: a survey of advances of deep learning within x-ray security imaging, Pattern Recognit. 122 (2022) 108245.

[7] R. Olmos, S. Tabik, F. Herrera, Automatic handgun detection alarm in videos using deep learning, Neurocomputing 275 (2018) 66–72.

[8] A. Castillo, S. Tabik, F. Pérez, R. Olmos, F. Herrera, Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning, Neurocomputing 330 (2019) 151–161.

[9] L. Pang, H. Liu, Y. Chen, J. Miao, Real-time concealed object detection from passive millimeter wave images based on the yolov3 algorithm, Sensors 20 (6) (2020) 1678.

[10] F. Pérez-Hernández, S. Tabik, A. Lamas, R. Olmos, H. Fujita, F. Herrera, Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: application in video surveillance, Knowl.-Based Syst. 194 (2020) 105590.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.

[12] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The Pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.

[13] Z. Cui, X. Wang, N. Liu, Z. Cao, J. Yang, Ship detection in large-scale sar images via spatial shuffle-group enhance attention, IEEE Trans. Geosci. Remote Sens. 59 (1) (2020) 379–391.

[14] A. Ravichandran, B. Yegnanarayana, Studies on object recognition from degraded images using neural networks, Neural Netw. 8 (3) (1995) 481–488.

[15] K. Shuang, Z. Lyu, J. Loo, W. Zhang, Scale-balanced loss for object detection, Pattern Recognit. 117 (2021) 107997.

[16] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European Conference on Computer Vision, Springer, 2014, pp. 818–833.

[17] A. Bochkovskiy, C.-Y. Wang, H.-Y.M. Liao, Yolov4: optimal speed and accuracy of object detection, arXiv preprint, arXiv:2004.10934.

[18] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding, et al., Pp-yolo: an effective and efficient implementation of object detector, arXiv preprint, arXiv:2007.12099.

[19] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3626–3633.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.

[21] B. Bosquet, M. Mucientes, V.M. Brea, Stdnet-st: spatio-temporal convnet for small object detection, Pattern Recognit. 116 (2021) 107929.

[22] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, E. Magli, Robust license plate recognition using neural networks trained on synthetic images, Pattern Recognit. 93 (2019) 134–146.

[23] I.K. Kallel, S. Almouahed, B. Solaiman, É. Bossé, An iterative possibilistic knowledge diffusion approach for blind medical image segmentation, Pattern Recognit. 78 (2018) 182–197.

[24] H. Hattori, N. Lee, V.N. Boddeti, F. Beainy, K.M. Kitani, T. Kanade, Synthesizing a scene-specific pedestrian detector and pose estimator for static video surveillance, Int. J. Comput. Vis. 126 (9) (2018) 1027–1044.
[25] W. Liu, B. Luo, J. Liu, Synthetic data augmentation using multiscale attention cyclegan for aircraft detection in remote sensing images, IEEE Geosci. Remote Sens. Lett. 19 (2021) 1–5.
[26] J.-H. Kim, Y. Hwang, GAN-based synthetic data augmentation for infrared small target detection, IEEE Trans. Geosci. Remote Sens. (2022).
[27] B. He, X. Li, B. Huang, E. Gu, W. Guo, L. Wu, Unityship: a large-scale synthetic dataset for ship recognition in aerial images, Remote Sens. 13 (24) (2021) 4999.
[28] W. Öhman, Data augmentation using military simulators in deep learning object detection applications, 2019, Master's thesis.
[29] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, The open images dataset v4: unified image classification, object detection, and visual relationship detection at scale, Int. J. Comput. Vis. 7 (2020).
[30] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, R. Girshick, Detectron2, https://github.com/facebookresearch/detectron2, 2019.
[31] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[32] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
[33] J. Redmon, A. Farhadi, Yolov3: an incremental improvement, arXiv preprint, arXiv:1804.02767.
[34] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 385–400.
[35] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6054–6063.
[36] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[37] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8759–8768.
[38] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 658–666.

Guanbo Wang, male, is a PhD candidate in School of Information Technology, Yunnan University. His research areas are deep learning, image processing and object detection. E-mail: wgb1018@gmail.com

Hongwei Ding, male, is a doctoral student supervisor in the School of Information Science, Yunnan University. His research interests are deep learning, computer vision. E-mail: wgb@mail.ynu.edu.cn

Mingliang Duan, male, The Key Laboratory of Internet of Things Technology and Application in Yunnan Province. His research interests are cloud computing, image processing. E-mail: rcod1024@163.com

Yuanyuan Pu, female, is a doctoral student supervisor in the School of Information Technology, Yunnan University. Her research interests are image processing, computer vision. E-mail: 15565315076@163.com

Zhijun Yang, male, is a doctoral student supervisor in the School of Information Science, Yunnan University. His research interests are blockchain, cloud computing and computer vision. E-mail: 1271578900@qq.com

Haiyan Li, female, is a doctoral student supervisor in the School of Information Technology, Yunnan University. Her research interests include image restoration, medical image processing, and object detection. E-mail: leehy@ynu.edu.cn