IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 July 2024, accepted 29 July 2024, date of publication 1 August 2024, date of current version 13 August 2024.

*Digital Object Identifier 10.1109/ACCESS.2024.3436709*

RESEARCH ARTICLE

# A Fine-Grained Detection Network Model for Soldier Targets Adopting Attack Action

**YU YOU, JIANZHONG WANG, ZIBO YU, YONG SUN, YIGUO PENG, SHENG ZHANG, SHAOBO BIAN, ENDI WANG, AND WEICHAO WU**
School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China

Corresponding author: Weichao Wu (wuweichao@bit.edu.cn)

This work was supported by the Defense Industrial Technology Development Program under Grant JCKY2021602B029.

**ABSTRACT** Owing to its ability to provide more accurate and detailed battlefield situational information, fine-grained detection research on soldier targets is of significant importance for military decision-making and firepower threat assessment. To address the issues of low detection accuracy and inaccurate classification in the fine-grained detection of soldier targets, we propose a fine-gain soldier target detection model based on the improved YOLOv8 (You Only Look Once v8). First, we developed a multi-branch feature fusion module to effectively fuse multi-scale feature information and used a dynamic deformable attention mechanism to help the detection model focus on key areas in deep-level features. Second, we proposed a decoupled lightweight dynamic head to extract the position and category information of soldier targets separately, effectively solving the problem of misclassification of soldier targets' attack actions under different poses. Finally, we used the Inner Minimum Points Distance Intersection over Union (Inner-MPDIoU) to further improve the convergence speed and accuracy of the network model. The proposed improvements are evaluated through comparative experiments conducted in published twenty-six test groups, and the effectiveness of the proposed method is demonstrated. Compared with the original model, our method achieved a detection precision of 78.9%, a 6.91% improvement; the mAP@50 (mean Average Precision at 50) was 79.6%, a 3.51% increase; and an mAP@50-95 of 63.8%, a gain of 5.28%. The proposed method achieves high precision and recall while reducing the computational complexity of the model, thereby enhancing its efficiency and robustness for fine-grained soldier target detection.

**INDEX TERMS** Fine-grained detection, soldier targets, YOLOv8, attack action, deep learning.

## I. INTRODUCTION

In order to achieve an accurate search and precision strike of targets, advanced target detection and recognition technology that is especially capable of fine-grained detection of targets is required in the fields of security monitoring, military reconnaissance, and automatic weapons. Fine-grained target detection technology can provide more accurate and detailed information about target objects for military decision-making, thereby improving operational efficiency and precision strike capabilities. Therefore, it is important to conduct research on fine-grained detection technologies for military targets in complex environments.

As an important national defense application, military target detection using traditional target detection methods has been studied by scholars in many countries. However, traditional object detection technology based on machine learning requires human-designed features, which enable the comprehensive, rapid, and accurate acquisition of target information in complex battlefield environments [1], [2]. In recent years, target detection methods based on deep learning using deep neural networks have surpassed classical machine learning target detection methods in terms of detection accuracy and speed, and have been widely applied in military target detection, automatic weapons, and many

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

other fields. Compared with traditional methods, deep learning methods can autonomously extract abstract image features that are beneficial for computer detection, have a greater ability to express complex structures, and have higher robustness and generalization.

At present, deep-learning target detection algorithms can be divided into anchor-based methods [3], [4] and anchor-free methods [5], [6]. The anchor-based target detection method generates multiple rectangular boxes with different sizes and proportions centered on each pixel in the image in advance to cover the entire image as much as possible. According to the generation stage of candidate boxes, anchor-based detection methods can be divided into two categories: one-stage detection and two-stage detection. The one-stage detection approach treats detection as a regression problem, and directly regresses the class and location of a target. The two-stage detection approach detects a target in two stages. In the first stage, the algorithm produces candidate regions, refines localization, and classifies these candidate regions in the second stage. The algorithm produces candidate regions in the first stage, refines the positions, and classifies the candidate regions in the second stage. Qin et al. [7] used an improved two-stage algorithm with a multi-level attention mechanism to perform the fine-grained detection of ship targets. This method adds attention mechanisms and residual connections [8] to the backbone network to extract multi-layer features, then uses deep separable convolutions [9] to deepen the network, and finally uses the ReLU activation function to reduce the computation amount of the overall network. Although this method has achieved significant improvements in accuracy, it uses a large number of residual connections, resulting in a large number of parameters and a high computational complexity. Azam et al. [10] compared the detection performance of algorithms such as RCNN (Region-based Convolutional Neural Networks), Fast RCNN, and Faster RCNN on aircraft targets. Despite the high accuracy of two-stage detection methods, their detection speed is slow and they cannot be applied to real-time detection scenarios. Therefore, faster single-stage algorithms are widely used in object detection tasks. Although the two-stage detection method offers higher precision, its slower detection speed prevents its application in real-time monitoring. Thus, the one-stage approach with a high detection frame rate was adopted more widely than the two-stage approach.

Kong et al. [11] employed a one-stage deep learning algorithm, YOLOv3 (You Only Look Once v3), to detect military targets such as tanks and soldiers. They improved the feature extraction and fusion capabilities of the network model by introducing GhostNet (Ghost Network) [12] and coordinate attention mechanism [13]. In addition, they redesigned the loss function of the detection model to enhance the detection accuracy of military targets further. Despite the effectiveness of the method discussed for detection, a considerable computational performance is required. Consequently, it is not feasible to implement this method on embedded devices with inferior computing capabilities. Wang et al. [14] improved YOLOv4 (You Only Look Once v4) by using a 3X-FPN (Three-channel Feature Pyramid Network) feature fusion network architecture for the fast detection of infrared military targets. They achieved data weighted balanced fusion through adaptive network parameter optimization to improve target detection accuracy. Infrared images can effectively avoid the influence of factors such as lighting and camouflage on detection performance, but they also lose rich texture features at the same time, making it very difficult to conduct fine-grained detection of military targets. Du et al. [15] proposed an improved YOLOv5 (You Only Look Once v5) detection algorithm to detect military targets such as ships, helicopters, tanks, and soldiers. This method replaces the focus module with a stem block and then embeds a coordinate attention module based on MobileNetV3 (Mobile Network V3) [16] blocks into the backbone network, which improves the average detection accuracy of the model. This improved approach can effectively distinguish different types of military targets; however, it has a relatively long inference time, which may lead to real-time performance issues for devices with limited hardware resources. However, both the one-stage and two-stage algorithms mentioned above require predefined anchor boxes, which can potentially introduce additional errors and have weak generalization capabilities.

The anchor-free algorithm does not require predefined sizes and positions of anchor boxes, and can adapt to targets with large-scale variations. Wang et al. [17] proposed a NAS-YOLOX (Neural Architecture Search You Only LooK Once X) algorithm with an anchor-free mechanism to detect ship targets in synthetic aperture radar images. The method they proposed improves the fusion performance of multi-scale feature information in the model throug a neural architecture search feature pyramid network and inserts an atrous convolution feature enhancement module into the backbone network to improve the receptive field and target information extraction capabilities of the network. This method increases the number of parameters and the computation complexity of the basic method, and both the inference and training times are longer. Shan et al. [18] introduced an UAVPNet (Unmanned Aerial Vehicle Pose Network) algorithm to detect UAV (Unmanned Aerial Vehicle) targets with different attitudes. In this study, multi-scale features of the target were extracted using the ResNet-50 (Residual Networks 50) backbone network. A BFP (Balanced Feature Pyramid) structure was then used for feature fusion, and a VarifocalNet (Varifocal Network) detection head was used to minimize information loss. However, the computation of this method reaches 139 Gflops, which is not suitable for deployment on embedded devices, and real-time performance is poor. Li et al. [19] addressed the problem of vehicle detection in aerial images and introduced Bi-FPN (Bidirectional Feature Pyramid Network) into YOLOv8s (You Only Look Once v8 Small). By fully considering and reusing the multi-scale features, they improved the feature fusion capability of the algorithm. Specifically, this method replaces part of the C2f module in the backbone with the GhostblockV2 structure to reduce the loss of feature information. Simultaneously, Wise-IoU loss was used to improve the overall performance of the detection task. In some cases, this method may misclassify similar backgrounds as targets and overfitting may be a problem. In addition, there are also detection algorithms based on transformer architecture; however, such algorithms require a much higher computing performance of the hardware, which is not discussed here.

In the above research on various types of military target detection, most studies focused on differentiating between different types of military targets with distinct appearances, functions, and large inter-class gaps, but did not delve into the fine-grained division of the same type of military targets with small intra-class gaps. Without fine-grained classification, dividing ordinary vehicles and armored vehicles into one category and military aircraft and civilian aircraft into one category results in significant challenges for the practical application of target detection technology in battlefield scenarios. Because military targets of different types and attributes have different functions and firepower threat levels, roughly grouping these targets together for detection will produce inaccurate results that would affect subsequent military decisions and even lead to misidentification and loss of strike capability. These military targets share similar characteristics, which make difficult to distinguish between them. For example, in medical ships, passenger ships, military transport aircraft, civilian aircraft, armed soldiers, and medical soldiers, these targets have similar appearance features but a very small percentage of distinguishing features. These similarities make it extremely difficult to classify specific military targets. This is one of the reasons why there is relatively little research on the fine-grained classification of military targets.

Currently, fine-grained detection methods for military targets face various challenges such as insufficient samples, imbalanced data distribution, and complex background environments. In these cases, the feature extraction and fusion capabilities of the network model are critical for the detection performance. To improve the practical application of network models, issues such as efficiency and speed must be considered to ensure that network models can efficiently detect images. Furthermore, existing research on military target detection mainly focuses on targets with significant inter-class differences, such as tanks, ships, and aircraft, whereas relatively little research has been conducted on the fine-grained detection of targets with smaller intra-class differences and different attributes within the same category. Existing methods have problems, such as low detection accuracy, high computational complexity, and overfitting to specific parts. Owing to limited time and resources, this study focuses on fine-grained detection and classification of soldier-type military targets. The specific contributions of this study are as follows.

- A fine-grained dataset was created for model training that considered soldiers' attack actions and the types of firearms they held.
- We introduced a dynamic deformable attention mechanism at the last layer of the backbone network to further improve the feature extraction capability of the network model.
- To enhance the fusion ability of network models for different weapon features, we proposed a multi-branch feature fusion module with dynamic snake convolution and atrous convolution to improve detection accuracy without significantly increasing computation.
- To address the problem of classification errors in the attack action of soldier targets under different poses, we propose a lightweight dynamic head and verify its effectiveness.
- By combining the Inner-IoU and MPDIoU loss functions, our method accelerates the convergence rate of network training and improves the detection performance of small objects.
- The experimental results demonstrate that our method has good detection efficiency and robustness when detecting soldier targets adopting attack actions using guns and rocket launchers in the untrained data.

The remainder of this paper is organized as follows. Section II describes the dataset and composition of the YOLOv8-AD (You Only Look Once v8 Attack Detection) network. Section III presents experiments and analyses. Finally, Section IV concludes the study.

## II. DATASET AND METHOD

### A. DATASET

This section introduces the production and distribution of the experimental dataset. We created a fine-grained detection dataset of soldier targets through data collection methods such as photo and video frame extraction from the Internet, war movies, and other channels, and the targets were marked using the Lable Image tool. The spatial interaction between soldiers and their weapons in the battlefield environment is complex, and the weapons they hold may be severely obstructed by the soldier's body. Moreover, the behavior of soldiers attacking while holding weapons is a fast and continuous action, and the same soldier may carry multiple different weapon payloads. In addition, static photos in the dataset contain temporal information. These issues pose significant difficulties for data annotation and detection.

Based on the types of weapons held by the soldier targets, soldiers' current posture, and the global information of the entire image, we divide the soldier targets into five categories: S represents ordinary soldiers who do not hold weapons, G represents soldiers who hold guns but do not exhibit obvious attacking behavior, R represents soldiers who hold a rocket launcher but do not exhibit obvious attacking behavior, RP represents soldiers who hold rocket launchers and exhibit obvious attacking action, and GP represents soldiers who

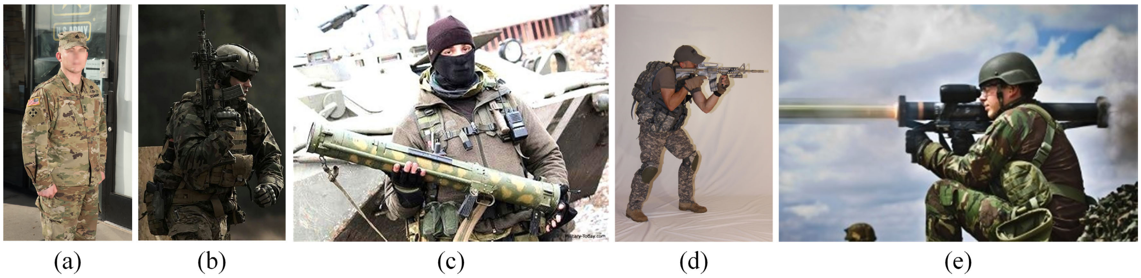

**FIGURE 1.** Examples for all categories of soldiers in the dataset; (a) an example of ordinary soldiers; (b) an example of soldiers with guns but no apparent attack action; (c) an example of soldiers who hold rocket launchers but do not exhibit obvious aggressive behavior; (d) an example of soldiers shooting with a gun; (e) an example of soldiers firing with rocket launchers.

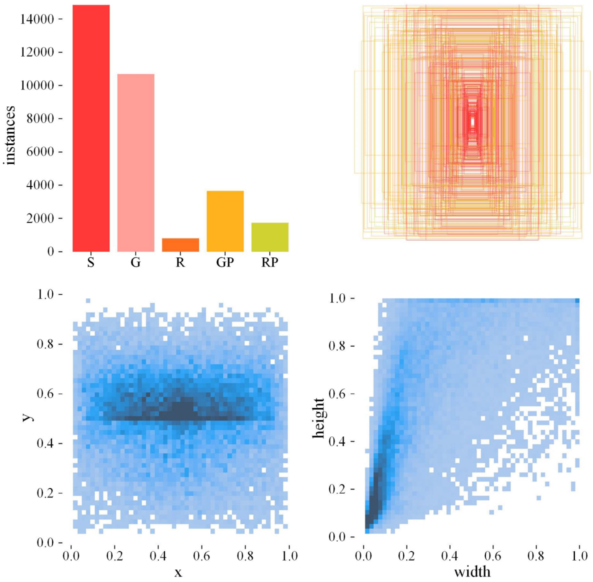

**FIGURE 2.** Visualization and distribution of the dataset. The top right is the visualization of bounding box. The top left is the number of annotations per class. The bottom left is statistical distribution of the center point of bounding box. The bottom right is statistical distribution of the bounding box sizes.

hold guns and exhibit obvious attacking action. Figure 1 shows actual examples for each category of soldiers.

Specifically, whether soldiers engage in shoulder fire shooting movements, whether their hands engage in pre-pulling the trigger, and whether they engage in aiming movements are used to determine soldiers' attack behavior. In addition, when there is severe obstruction between the soldier and the weapon they hold, making it impossible to determine whether they hold the weapon, priority should be given to categorizing them as non-weapon types. When soldiers with multiple firearms appear, priority should be given to classifying them based on the types of weapons in their hands. Finally, the dataset we created contains 13,329 images; the distribution of each category, label box, center point, and pixel size in the dataset is shown in Figure 2. During training, the built dataset was divided in a ratio of 8:1:1.

## B. THE PROPOSED METHOD

In this study, a network model YOLOv8-AD using YOLOv8 as a framework was established for the fine-grained detection of soldier targets adopting an attack action. The multi-branch C3DSA (Concentrated-Comprehensive Convolution block with a Dynamic Snake and Atrous convolution) module with snake convolution [20] and atrous convolution [21] is proposed to improve the feature expression capability of the network for soldiers' weapon holdings, and a deformable multi-head attention mechanism [22] is inserted into the backbone network. A lightweight dynamic head that combines a dynamic head [23] and ghost conv is proposed to enhance the accuracy of network models for detecting and classifying soldier attack actions, and the Inner-MPDIoU is used to further improve the convergence speed of the network. The YOLOv8-AD network structure diagram is shown in Figure 3, where the red border represents the improved modules in the YOLOv8 network. The specific details of each improvement module can be found in the corresponding sections.

### 1) MULTI-BRANCH FEATURE FUSION MODULE

One of the biggest differences between YOLOv8 and YOLOv5 is that the C3 module was replaced with the C2f module by comparing the network structures. The structures of C3 and C2f are shown in Figure 4. The C3 module combines the idea of CSPNet and residual connection, which has the advantage of fewer parameters and fewer computations, but has the problem of limited expressive ability. To overcome the shortcomings of the C3 module, YOLOv6 (You Only Look Once v6) [24] proposes to improve the C3 module with a reparameterization module RepVGG (Reparameterization Visual Geometry Group) block to obtain a more efficient backbone network, whereas YOLOv7 (You Only Look Once v7) [25] uses ELAN (Effective Layer Aggregation Network) block instead of a bottleneck to obtain more gradient flow information. The C2f module used in YOLOv8 [26] adds a split operation to the C3 module and uses a more flexible structure with rich gradient flow information, but higher computational complexity.

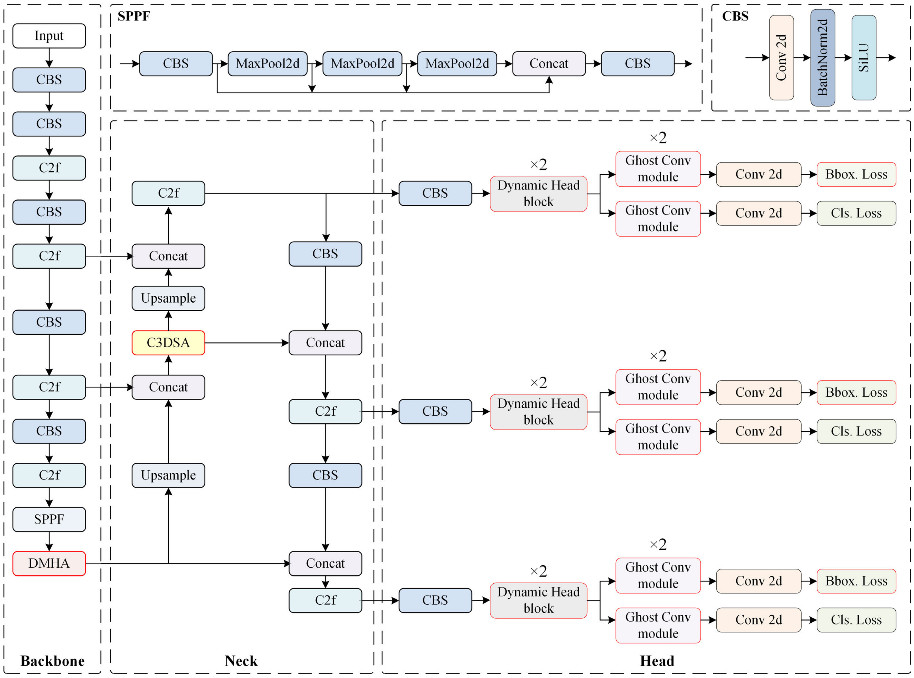

**FIGURE 3.** The structure of proposed YOLOv8-AD network.

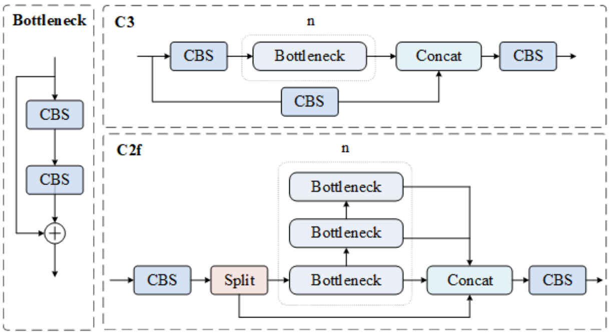

**FIGURE 4.** The comparison of structure between C3 module and C2f module.

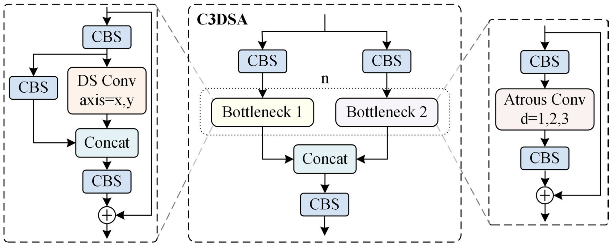

**FIGURE 5.** The structure of proposed C3DSA module, the left branch combines dynamic snake convolution and the right branch integrates multiple scales atrous convolutions.

To enhance the feature extraction and fusion capability without significantly increasing the number of parameters, we optimized the C3 module and proposed an improved multi-branch feature fusion module named C3DSA, as illustrated in Figure 5. We replace one branch bottleneck on the basis of the C3 structure, and add different types of bottlenecks on the other branch to further enrich the gradient information. One of the branches introduces Dynamic Snake Convolution (DS Conv) to improve its ability to express the tubular structural features of the weapons. DS Conv uses an iterative strategy to straighten the standard convolution kernel on the x-axis and y-axis for convolution operations, and selects the following positions to be observed for each target to be processed, thereby ensuring the continuity of observation. Taking a convolution kernel of size 7 as an example, in the x-axis direction, the specific position of each grid in $K$ is expressed as: $K_{i\pm p} = (x_{i\pm p}, y_{i\pm p})$, where $p = \{0, 1, 2, 3\}$ represents the horizontal distance from the center grid. Starting from the center position $K_i$, each grid position $K_{i\pm p}$ in the convolution kernel $K$ depends on the position of the previous grid: compared with $K_i$, $K_{i+1}$ incremented by the offset $\Delta = \{\delta | \delta \in [-1, 1]\}$. In a word, DS Conv can adaptively adjust the shape and size of the convolution kernels to better capture the characteristics of the target's weapon at different scales and poses. Meanwhile, atrous convolutions of different sizes are added to the other branch, allowing it to expand its information receptive field while adapting to targets of different scales. The calculation process of C3DSA is shown in Algorithm 1.

The C3DSA module further enriches the gradient flow information based on the C3 module, with more efficient feature representation capabilities, which can improve the detection accuracy of network models for soldiers holding different weapons in different poses. The expressive ability

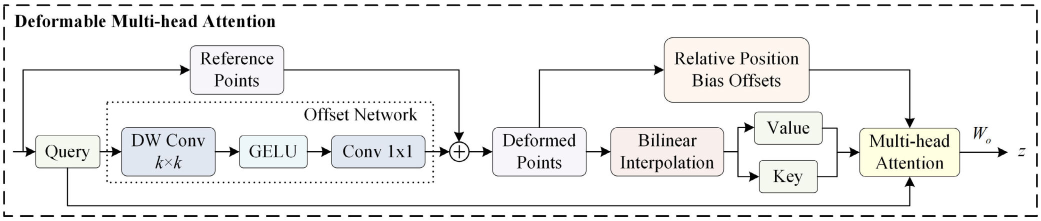

**FIGURE 6.** The processing flow of DMHA.

```
Algorithm 1 Pseudocode for C3DSA Module
Input: $x$ Input feature map
Output: $y$ Output feature map after processing
Input channel $c1$, output channel $c2$, number of repetitions
of the bottleneck block $n$, $shortcut$, $e$
1.Calculating hide channels: $c\_ = c2 \cdot e$

2.Define CBS processing layer:
$y\_cv1 = \mathrm{CBS}(x, c1, c\_)$
$y\_cv2 = \mathrm{CBS}(x, c1, c\_)$

3.Define DS Conv sequence processing bottleneck:
for $1 to n$ do
    $y1 \leftarrow \mathrm{Bottleneck1}(y\_cv1, c\_, c\_, shortcut)$
end
4.Define Atrous Conv sequence processing bottleneck:
for $1 to n$ do
    $y2 \leftarrow \mathrm{Bottleneck2}(y\_cv2, c\_, c\_, e)$
end
5.Calculate the output result:
$y\_cat = Concatenate(y1, y2, axis = 1)$
$y = \mathrm{CBS}(y\_cat, 2 \cdot c\_, c2)$
return $y$
```

of C3DSA modules for deep and shallow features was investigated by adding C3DSA modules to different locations in the YOLOv8 network; the results are presented in Table 3.

### 2) DEFORMABLE MULTI-HEAD ATTENTION

When processing high-resolution target images, the semantic information of low-level and mid-level features may not be sufficient to provide accurate classification and localization information for soldier attack actions, which could reduce the detection precision in complex scenarios. During the process of detecting soldier attack actions, it is crucial to consider multiple local details, including the head, hands, and shoulders, as well as global information related to the soldier's body posture and the surrounding environment. Over-focusing on either the local or global context may be affected by irrelevant parts outside the area of interest, thereby affecting the inference speed of the network model. Deformable Multi-Head Attention (DMHA) is inserted into the backbone network to dynamically learn the key regions of the target to address these issues. The processing flow of DMHA is shown in Figure 6. Dynamic sampling points were adopted in the DMHA, enabling the model to focus more intently on the information that is most important for the current task.

In particular, the DMHA first generates reference points and a query map based on the feature map and then inputs the query into the offset network to produce offsets. Multiple sets of deformation sampling points were obtained, based on the reference points and offsets, and the key and value were calculated through bilinear interpolation and the projection matrix. Finally, the focus area was determined by concatenating the query, key, value, and relative position bias offsets using a multi-head attention module. The output can be expressed by the following formula:

$$z = Concat\left(z^{(1)}, \ldots, z^{(m)}\right) W_o, m = 1, \ldots, M \qquad (1)$$

$$z^{(m)} = \sigma\left(\frac{q^{(m)}\tilde{k}^{(m)\mathrm{T}}}{\sqrt{d}} + \phi(\hat{B}; R)\right)\tilde{v}^{(m)} \qquad (2)$$

where $z^{(m)}$ denotes the embedding output from the $m$-th attention head, and $W_o \in \mathbb{R}^{C\times C}$ are the projection matrices. $\sigma(\cdot)$ denotes the softmax function and $d = C/M$ is the dimension of each head. $q^{(m)}$, $\tilde{k}^{(m)}$, $\tilde{v}^{(m)} \in \mathbb{R}^{N\times d}$ denote the query, key, and value embedding, respectively. $\phi(\cdot;\cdot)$ is the sampling function set obtained using bilinear interpolation. $\hat{B}$ denotes the relative position bias table, and $R$ denotes the relative displacement. In Table 4, we compare the contributions and effects of the six different attention mechanisms on model performance.

### 3) LIGHTWEIGHT DYNAMIC HEAD

The final output of a deep neural network depends on the detection head. Thus, the classification and localization capabilities of the detection head are crucial to the performance of the network model. To improve the perceptual ability of the network model in different dimensions, we introduced a dynamic head block into the original YOLOv8n detection head framework. The re-scaled feature pyramid is treated as a 3-dimensional tensor $x \in R^{N^*S^*C}$, where the $N$ represents the number of levels in the pyramid and $C$ and $S$ represent the number of channels and feature size respectively. The feature size can be calculated using height and width as $S = H \times W$. The cascade attention operation of tensor $x$ is performed separately for each of the three dimensions in the dynamic

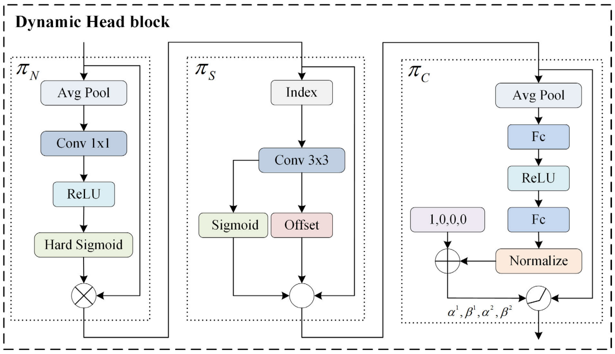

**FIGURE 7.** **The dynamic head block.**

head block, as represented by the following formula:

$$F(x) = \pi_C\Big(\pi_S\big(\pi_N(x)\cdot x\big)\cdot x\Big)\cdot x \tag{3}$$

In this equation, $\pi(\cdot)$ is an attention function, and the attention functions for different dimensions are designed according to the characteristics of each dimension. In the scale-aware dimension, a dynamic fusion approach was adopted to perceive information through $1 \times 1$ convolution, followed by average pooling, activation with ReLU and Hard Sigmoid, and splicing with the initial input to produce the output result. Spatial-aware attention can be decomposed into two steps: first, making the attention learning sparse by using deformable convolution and then aggregating features across levels at the same spatial locations. Meanwhile, task-aware attention favors different tasks by dynamically switching the channels of the features ON and OFF. The details of the attention mechanism in different dimensions within the dynamic head block are shown in Figure 7, which was implemented in a cascade manner from scale-aware $\pi_N$, spatial-aware $\pi_S$, and task-aware $\pi_C$.

In YOLOv8, the parameter count of the original decoupled detection head was relatively large, accounting for approximately 28% of the parameters of the entire network. Within the dynamic head, a multi-dimensional attention mechanism was implemented in a cascading manner. Simply combining the two modules would result in a significant increase in the number of parameters and computational complexity and potentially slow down the network's training convergence rate. To address this issue, we propose a lightweight dynamic head, as shown in Figure 8. We introduced Ghost Conv on the decoupled branches of the detection head to significantly reduce its computation and speed up model inference while improving the detection accuracy. Table 5 presents the results of several ablation experiments conducted to verify the effectiveness of the method.

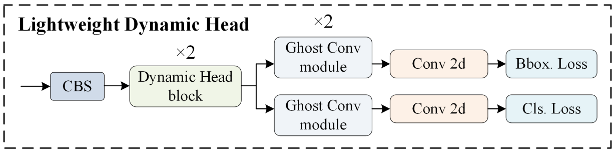

**FIGURE 8.** **The structure of proposed lightweight dynamic detection head that combines dynamic head and ghost convolution.**

### 4) INNER-MPDIOU LOSS

An advanced decoupled head improves the convergence speed but also causes misalignment issues between the classification and regression tasks. To solve this misalignment, YOLOv8 employed a task alignment learning technique [27] to enhances the alignment consistency between the classification and regression tasks. Specifically, for classification loss, YOLOv8 employs simple BCE (Binary Cross-Entropy) and SiLU [28] activation function to calculate the probabilities for each class. For regression loss, the DFL (Distribution Focal Loss) [29] is combined with the CIoU [30] loss to calculate the regression loss between the predicted and ground truth bounding boxes. Although CIoU considers factors such as IoU, center distance, and aspect ratio between the ground truth and predicted bounding boxes, the computation of CIoU is complex and does not consider the mismatching directions between predicted the and ground-truth bounding boxes. This results in the predicted bounding boxes not being able to continuously regress during training, particularly affecting small targets. To tackle this challenge, our network introduces MPDIoU [31] to enhance regression efficiency and precision and incorporates Inner-IoU [32] to further improve its detection performance on small target samples. The formulas for the MPDIoU and loss are as follows:

$$L_{MPDIoU} = 1 - MPDIoU \tag{4}$$

$$MPDIoU = IoU - \frac{d_1^2 + d_2^2}{w^2 + h^2} \tag{5}$$

$$\begin{aligned} d_1^2 + d_2^2 &= (x_1^{prd} - x_1^{gt})^2 + (y_1^{prd} - y_1^{gt})^2 \\ &\quad + (x_2^{prd} - x_2^{gt})^2 + (y_2^{prd} - y_2^{gt})^2 \end{aligned} \tag{6}$$

where $w$ and $h$ are the width and height of the input image, $(x_1^{prd}, x_2^{prd}, y_1^{prd}, y_2^{prd})$ are the coordinates of the prediction bounding box; and $(x_1^{gt}, x_2^{gt}, y_1^{gt}, y_2^{gt})$ are the coordinates of the ground truth bounding box. As shown in Equation (5), MPDIoU extends IoU by incorporating a distance term between the top-left and bottom-right corner points of the bounding box. This enables MPDIoU to capture the disparities between the predicted and ground-truth bounding boxes more accurately, particularly when distinguishing between boxes with the same aspect ratio but different sizes or positions. However, the effect of MPDIoU on small target samples requires further improvement. Inner-IoU based on auxiliary bounding boxes demonstrates that using larger auxiliary bounding boxes to calculate the loss has a significant effect on the regression of low IoU samples, while high IoU samples have the opposite effect. The Inner-MPDIoU loss was proposed to improve the performance of the original MPDIoU on small target samples by incorporating Inner-IoU. The Inner-MPDIoU loss function

**TABLE 1.** Experimental platform environment configurations.

| Experimental platform | Configurations |
|---|---|
| System | Ubuntu 22.04 |
| GPU | Nvidia Titan V (12G) |
| CPU | Intel Core i7-10700 |
| Framework | PyTorch 2.1.0 |
| Programming environment | Python 3.8 |
| Parallel computing architecture | CUDA 12.1 |

is calculated as follows:

$$x_1^{gt} = x_c^{gt} - \frac{w^{gt} * ratio}{2}, x_2^{gt} = x_c^{gt} + \frac{w^{gt} * ratio}{2} \tag{7}$$

$$y_1^{gt} = y_c^{gt} - \frac{h^{gt} * ratio}{2}, y_2^{gt} = y_c^{gt} + \frac{h^{gt} * ratio}{2} \tag{8}$$

$$x_1^{pre} = x_c^{pre} - \frac{w^{pre} * ratio}{2}, x_2^{pre} = x_c^{pre} + \frac{w^{pre} * ratio}{2} \tag{9}$$

$$y_1^{pre} = y_c^{pre} - \frac{h^{pre} * ratio}{2}, y_2^{pre} = y_c^{pre} + \frac{h^{pre} * ratio}{2} \tag{10}$$

$$\begin{aligned} inter &= (min(x_1^{gt}, x_1^{pre}) - max(x_2^{gt}, x_2^{pre})) \\ &\quad * (min(y_2^{gt}, y_2^{pre}) - max(y_1^{gt}, y_1^{pre})) \end{aligned} \tag{11}$$

$$union = w^{gt} * h^{gt} * ratio^2 + w^{pre} * h^{pre} * ratio^2 - inter \tag{12}$$

$$IoU_{inner} = \frac{inter}{union} \tag{13}$$

$$L_{inner-MPDIoU} = L_{MPDIoU} + IoU - IoU_{inner} \tag{14}$$

In Equations (7)-(10), $(x_c^{pre}, y_c^{pre})$, $(x_c^{gt}, y_c^{gt})$ are the center points of the predicted and ground-truth bounding boxes, respectively. In subsequent experiments, the inner ratio was set at 1.05. Table 6 reports the results of several ablation experiments to validate the effectiveness of the method.

## III. EXPERIMENT AND ANALYSIS

Our network was based on the PyTorch framework, and an NVIDIA Titan V was used for model training on the Ubuntu 22.04 system. The remaining configurations are listed in Table 1.

The specific hyperparameter settings used for training were as follows: the input image size was 640, IoU threshold was 0.5, optimizer was SGD, batch size was set to 32, initial learning rate was 0.01, final learning rate was 0.0001, training was conducted for 300 epochs, and the momentum decay and weight decay parameters were set to 0.937 and 0.0005, respectively. The Precision-Recall curve of YOLOv8-AD training is shown in the Figure 9.

### A. ABLATION EXPERIMENTS

#### 1) EXPERIMENTS WITH C3DSA AT DIFFERENT POSITIONS

The positions and numbers of layers of the C2f module in the YOLOv8n network are listed in Table 2. An improved network that replaces only a single C2f module with a C3DSA module at the corresponding position was tested on the self-made soldier target fine-grained detection dataset. The experimental results are presented in Table 3.

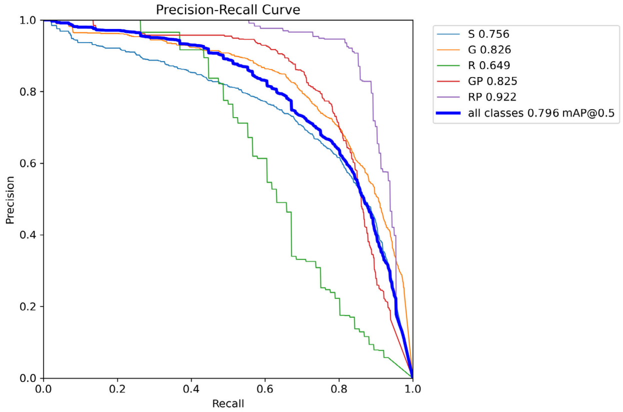

**FIGURE 9.** The Precision-Recall curve of YOLOv8-AD.

**TABLE 2.** The position of C2f module in the original YOLOv8n Network.

| Position | Layer | Repeat | Module |
|---|---|---|---|
| Backbone | 2 | 1 | C2f |
| | 4 | 2 | C2f |
| | 6 | 2 | C2f |
| | 8 | 1 | C2f |
| Neck | 12 | 1 | C2f |
| | 15 | 1 | C2f |
| | 18 | 1 | C2f |
| | 21 | 1 | C2f |

**TABLE 3.** Performance comparison of C3DSA modules at different positions in the YOLOv8n network.

| Layer | mAP50 | mAP50-95 | Parameters | GFLOPs | FPS |
|---|---|---|---|---|---|
| None | 76.9% | 60.6% | 2.87 M | 8.1 | 125 |
| 2 | 78.6% | 61.4% | 2.87 M | 8.2 | 105 |
| 4 | 78.0% | 61.6% | 2.89 M | 8.2 | 102 |
| 6 | 77.3% | 60.9% | 2.96 M | 8.1 | 107 |
| 8 | 77.6% | 61.2% | 3.04 M | 8.1 | 113 |
| 12 | 78.5% | 61.7% | 2.91 M | 8.1 | 110 |
| 15 | 78.9% | 61.8% | 2.88 M | 8.1 | 110 |
| 18 | 77.8% | 61.3% | 2.91 M | 8.1 | 114 |
| 21 | 78.2% | 61.3% | 3.04 M | 8.1 | 109 |

The same hyperparameters were used to ensure fairness. The effects of adding C3DSA modules at different positions on the model parameters and computation can be ignored. However, the impact on accuracy is significant: replacing any C2f module with a C3DSA module improves the performance of the network model on the dataset to different degrees. Moreover, replacing the shallow layers in the backbone and neck networks with C3DSA seems to yield better results, such as at layers 2, 4, 12, and 15. This is likely due to information loss occurring when the C3DSA module processes deep features. Compared to used for feature extraction in the backbone network, C3DSA demonstrates a more powerful feature fusion capability in the neck network. The best result was achieved by replacing the C2f module at layer 15 in the neck, which improved the mAP50 by 2.6% and mAP50-95 by 1.98% compared to the original detection model.

**TABLE 4. Performance comparison of different attention modules.**

| Attention | mAP50 | mAP50-95 | F1 | Parameters | GFLOPs |
|---|---|---|---|---|---|
| None | 76.9% | 60.6% | 72.7% | 2.87 M | 8.1 |
| +MHSA | 77.5% | 61.1% | 73.5% | 3.06 M | 8.3 |
| +SimAM | 77.7% | 61.5% | 74.2% | 2.87 M | 8.1 |
| +MLCA | 77.8% | 61.3% | 73.9% | 2.87 M | 8.1 |
| +MSCA | 78.1% | 60.8% | 73.7% | 2.96 M | 8.2 |
| +CPCA | 78.3% | 61.6% | 75.2% | 2.99 M | 8.3 |
| +DMHA | 78.4% | 61.8% | 75.0% | 3.12 M | 8.3 |

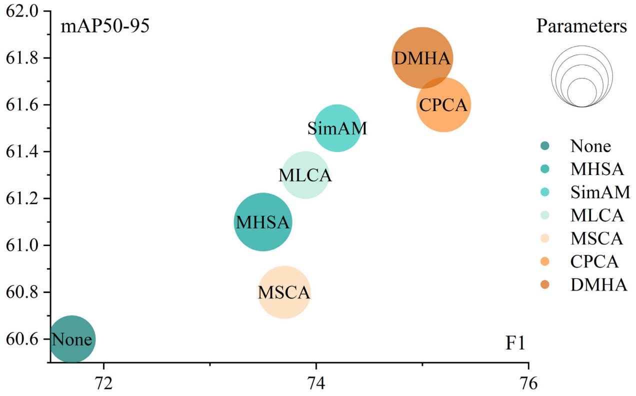

**FIGURE 10. The performance of different attention mechanism on YOLOv8n.**

### 2) EXPERIMENTS WITH DIFFERENT ATTENTION MECHANISMS

In Table 4, we test the performance of the YOLOv8n detector built with six different attention mechanisms on our self-made soldier target fine-grained detection dataset. The methods tested were MHSA [33], SimAM [34], MLCA [35], MSCA [36], CPCA [37] and DMHA.

The results demonstrate that although the different attention modules increase a small number of parameters and computations, there is a significant improvement in the accuracy and inference speed. Among them, DMHA achieved the best performance, with an increase of 1.95% in mAP50 and 1.98% in mAP50-95 compared to the detection model without the attention mechanism, and an increase of 3.16% and 17.6% in the F1 score and processing speed, respectively. Figure 10 shows the performance of adding each attention module to YOLOv8n, where the horizontal axis represents the F1 score, the vertical axis represents mAP50-95, and the circle size represents the number of normalized network model parameters. Figure 11 shows a heatmap before and after DMHA. After the DMHA, the attention of the network is increasingly focused on the target.

### 3) EXPERIMENTS WITH DIFFERENT HEAD AND LOSS FUNCTIONS

Additionally, we conducted comparative tests on different loss functions and detection heads to examine how the mAP and inference time are influenced by the CIoU and Inner-MPDIoU loss functions, as well as the Original Head (OH) and Lightweight Dynamic Head (LDH) detection heads. The experimental results for the dataset are listed in Table 5. Using Inner-MPDIoU and LDH increased mAP50 by 1.27% compared with using CIoU and LDH, and it increased mAP50-95 by 2.74% compared with using Inner-MPDIoU and the OH.

### 4) COMPARISON OF VARIOUS IMPROVED COMBINATIONS

To visually evaluate the effectiveness of each improvement, we validated the effectiveness of the adopted methods by using YOLOv8n as the baseline. The results of the ablation experiments are listed in Table 6.

In Table 6, the original model is denoted as Group 0. Groups 1, 2, and 3 represent the models constructed by adding different modules to the original model. Groups 4, 5, and 6 tested the combinations of more than two components simulaneously to evaluate the performance differences. According to Group 1, the addition of the C3DSA module significantly improved the precision, mAP50, and mAP50-95 of the model without significantly increasing the parameter and computational load, but the recall rate slightly decreased. Group 2 showed that the model with DMHA had the similar recall rate as the original YOLOv8n, while improving in precision, mAP50, mAP50-95, F1, and FPS (Frame Per Second). Among the three added improvements, the LDH with the Inner-MPDIoU loss function offered the most significant enhancement to the original YOLOv8n, achieving a balance between precision and recall of the model, but the FPS has decreased. In Group 5, the model that incorporates C3DSA and LDH with Inner-MPDIoU loss function achieved the highest accuracy of 81.0%. Finally, compared to the original model, the YOLOv8-AD model in Group 7 achieved the best mAP50 and mAP50-95, with improvements of 3.51% and 5.28% respectively. YOLOv8-AD also improved the performance in other indicators: the precision increased by 6.9%, the recall rate increased by 1.81%, and the F1 score increased by 4.26%. Despite the relatively low processing speed of the network model, it remains within an acceptable range for applications.

### 5) COMPARISON BETWEEN MODELS

Finally, we compared the YOLOv8-AD model with other YOLO series models on the self-built dataset, as presented in Table 7. Figure 12 shows a radar chart of the fine-grained detection performance of soldier target for each model. YOLOv8-AD achieved the best mAPs and F1 scores with a small increase in the parameters and computation.

To study the scope of application and effectiveness of the YOLOv8-AD model, we further conducted a comparison on public dataset Pascal VOC 2012. The results presented in Table 8 show that YOLOv8-AD still performs better than the original YOLOv8n on the Pascal VOC 2012 dataset. Although the training time of YOLOv8-AD has increased, the accuracy and recall of all categories in the Pascal VOC 2012 dataset have been improved.

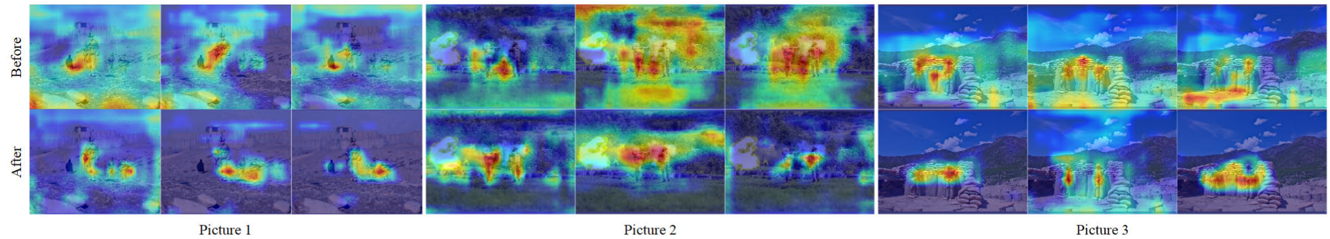

**FIGURE 11.** The heatmap of three different pictures before and after DMHA.

**TABLE 5.** The effect of CIoU/Inner-MPDIoU and OH/LDH on the detector.

| Model | Loss | Head | mAP50 | mAP50-95 | Parameters | GFLOPs |
|---|---|---|---|---|---|---|
| YOLOv8n+C3DSA+DMHA | CIoU | OH | 78.6% | 61.9% | 3.14 M | 8.3 |
| | CIoU | LDH | 78.6% | 62.6% | 3.28 M | 8.1 |
| | Inner-MPDIoU | OH | 79.4% | 62.1% | 3.14 M | 8.3 |
| | Inner-MPDIoU | LDH | 79.6% | 63.8% | 3.28 M | 8.1 |

**TABLE 6.** Ablation study of three tools: block, attention mechanism, and head with loss functions.

| Group | YOLOv8n | C3DSA | DMHA | LDH(IM) | P | R | mAP50 | mAP50-95 | F1 | Parameters | GFLOPs | FPS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ✓ | | | | 73.8% | 71.7% | 76.9% | 60.6% | 72.7% | 2.87 M | 8.1 | 125 |
| 1 | ✓ | ✓ | | | 78.6% | 70.9% | 78.9% | 61.8% | 74.6% | 2.88 M | 8.1 | 111 |
| 2 | ✓ | | ✓ | | 78.4% | 71.6% | 78.4% | 61.8% | 74.8% | 3.12 M | 8.3 | 147 |
| 3 | ✓ | | | ✓ | 78.6% | 72.4% | 78.9% | 62.9% | 75.4% | 3.01 M | 7.8 | 95 |
| 4 | ✓ | ✓ | ✓ | | 80.5% | 70.0% | 78.6% | 61.9% | 74.9% | 3.14 M | 8.3 | 100 |
| 5 | ✓ | ✓ | | ✓ | 81.0% | 70.4% | 78.4% | 62.6% | 75.3% | 3.03 M | 7.9 | 98 |
| 6 | ✓ | | ✓ | ✓ | 79.6% | 72.1% | 79.2% | 63.1% | 75.7% | 3.27 M | 8.0 | 96 |
| 7 | ✓ | ✓ | ✓ | ✓ | 78.9% | 73.0% | 79.6% | 63.8% | 75.8% | 3.28 M | 8.1 | 94 |

**TABLE 7.** The results of five detection models.

| Model | Type | Input | mAP50 | mAP50-95 | F1 | Parameters | GFLOPs | FPS |
|---|---|---|---|---|---|---|---|---|
| YOLOv3-Tiny | anchor based | 640x640 | 71.1% | 50.4% | 59.4% | 11.6 M | 18.9 | 171 |
| YOLOv5n | anchor based | 640x640 | 74.5% | 56.5% | 70.8% | 2.39 M | 7.1 | 162 |
| YOLOv6n | anchor free | 640x640 | 75.0% | 59.5% | 72.0% | 4.04 M | 11.8 | 191 |
| YOLOv8n | anchor free | 640x640 | 76.9% | 60.6% | 72.7% | 2.87 M | 8.1 | 125 |
| YOLOv8-AD(ours) | anchor free | 640x640 | 79.6% | 63.8% | 75.8% | 3.28 M | 8.1 | 94 |

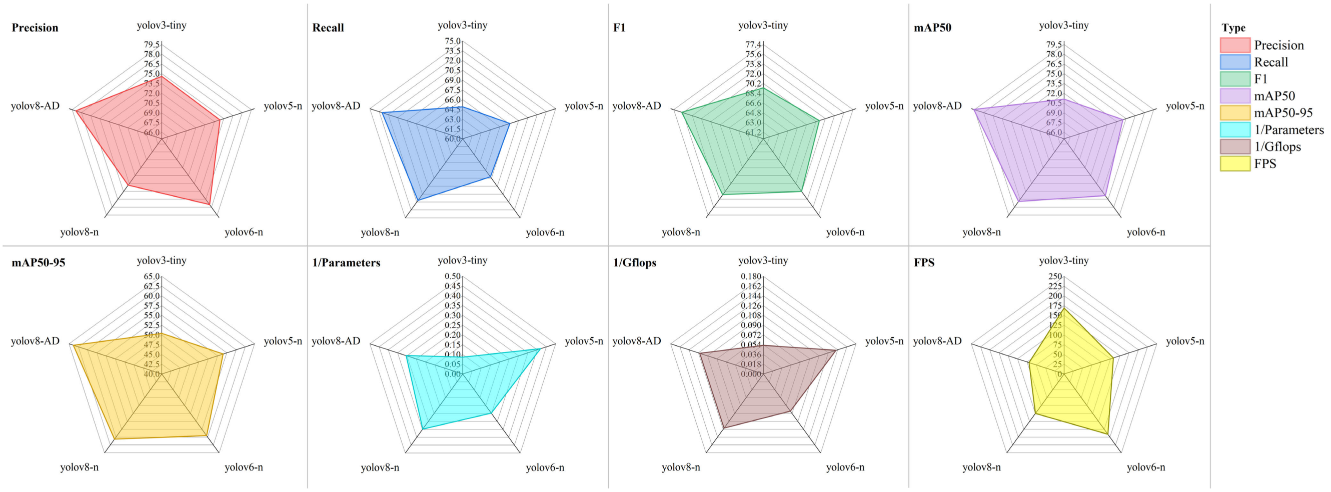

**FIGURE 12.** YOLO series model detection performance radar charts.

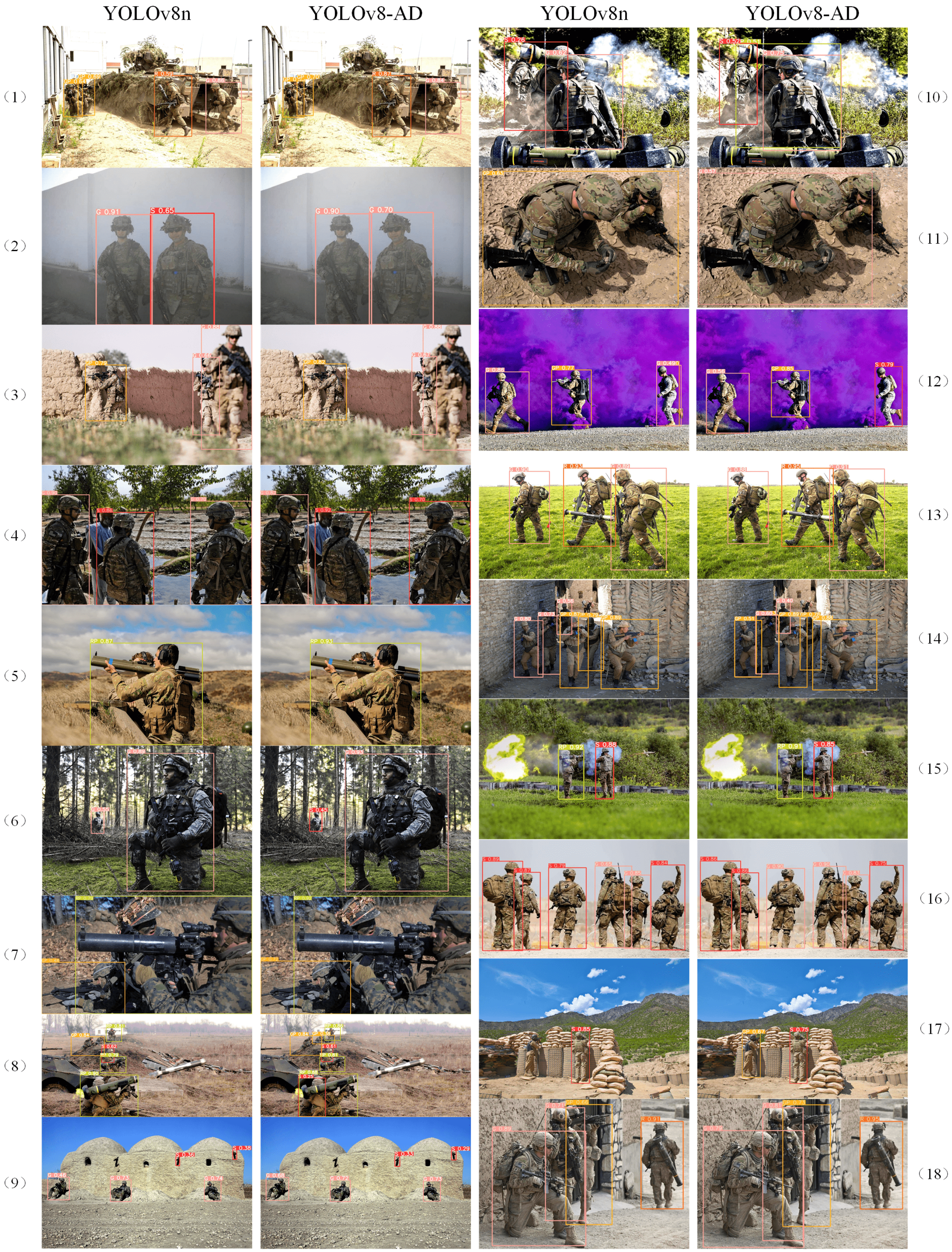

**FIGURE 13.** The prediction results of YOLOv8n and proposed YOLOv8-AD.

**TABLE 8.** The Comparison on public dataset Pascal VOC 2012.

| Model | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| YOLOv3-Tiny | 67.4% | 56.6% | 58.5% | 35.5% |
| YOLOv5n | 68.8% | 58.3% | 63.9% | 45.9% |
| YOLOv6n | 72.3% | 60.5% | 66.4% | 49.0% |
| YOLOv8n | 70.1% | 60.9% | 66.5% | 48.8% |
| YOLOv8-AD(ours) | 73.5% | 61.4% | 67.7% | 51.0% |

**TABLE 9.** Test results for five categories of soldiers.

| Type | Precision | Recall | mAP50 | mAP50-95 |
|---|---|---|---|---|
| S | 68.3% | 71.9% | 72.3% | 50.0% |
| G | 69.3% | 76.8% | 80.1% | 63.7% |
| R | 69.7% | 56.5% | 65.3% | 55.4% |
| GP | 79.9% | 73.2% | 82.2% | 65.2% |
| RP | 91.1% | 91.4% | 95.4% | 80.3% |
| Total | 75.7% | 74.0% | 79.1% | 62.9% |

### B. SOLDIER TARGET FINE-GRAINED DETECTION RESULTS

Table 9 presents the detection results of testing 1,333 randomly selected soldier target images that were not previously trained using the YOLOv8-AD model. The precision and recall rates were 75.7% and 74%, respectively, and the average detection time per image was 11.2ms. Prediction results of eighteen images containing five types of targets using YOLOv8n and YOLOv8-AD are illustrated in Figure 13. According to the test results, both YOLOv8n and YOLOv8 AD have missed and false detections, as shown in image (9). Both YOLOv8n and YOLOv8 AD incorrectly identify the irregularly shaped hole in the upper right corner as a soldier. However, it is evident that the false detection rate and miss detection rate of YOLOv8-AD are lower than YOLOv8n. In addition, missed and false detections mostly occur in situations involving small targets, occlusion, blurring, uneven lighting, and similar features. In image (6), the small target of the soldier on the left is obscured by many small branches and YOLOv8n mistakenly identifies it as a soldier with a gun. This may be due to some similarities in appearance and shape between tree branches and firearms, and YOLOv8-AD's ability to accurately classify targets in this case can be attributed to the contribution of the C3DSA module. In images (2), (11), (12), and (14), YOLOv8n experienced classification errors due to poor image quality and unclear firearm features. In particular, when soldiers are shooting with weapons, YOLOv8n is more prone to classification errors, and its ability to capture interactions between soldiers and firearms is significantly weaker than that of YOLOv8-AD.

Moreover, in most cases, the confidence of detection results from YOLOv8-AD is higher than YOLOv8n. In short, YOLOv8-AD can more accurately locate the soldier target and effectively distinguish whether the type of weapon held by the soldier is a gun or a rocket launcher and whether they are taking attack action.

## IV. CONCLUSION

In this paper, we propose a soldier target fine-grained detection method based on the YOLOv8-AD network model to address the low detection accuracy and inaccurate classification in the detection process of soldier targets taking attack action with different firearms. This method accurately locates and classifies soldiers that hold different weapons, such as guns and rocket launchers, taking attack action through a dynamic deformable attention mechanism, multi-branch modules with dynamic snake convolution and atrous convolution, lightweight dynamic detection head with multi-dimensional attention, and a network based on the Inner-MPDIoU loss function. Comparative experimental results show that YOLOv8-AD achieves an mAP50 of 79.6% on the soldier target fine-grained dataset. The precision and recall rates of soldier target detection using this method on randomly selected 1,333 untrained images were 75.7% and 74% respectively, with an average detection time of 11.2ms per image. The proposed method for fine-grained detection of soldier targets can quickly and accurately identify and locate the type of weapon held by soldiers and whether they are taking attack action, providing a new solution for fine-grained detection of soldier targets in the fields of security monitoring and automatic weapons.

However, attack action is a complex and continuous behavior. This study relies solely on single-frame images to detect the attack action of the soldier target, lacking contextual temporal information, which brings great difficulties to both annotation and detection. Moreover, there is a problem of an imbalanced sample distribution and low data quality in the created dataset. Although we used data augmentation during the training process, beneficial effects were limited. In future research, we will reduce the difficulty of training by adding temporal data and balancing the sample distribution, and improve the real-time performance and efficiency of detection.

## REFERENCES

[1] S. Kim, "Target attribute-based false alarm rejection in small infrared target detection," *Proc. SPIE*, vol. 8537, pp. 115–126, Nov. 2012.

[2] B. N. Nelson, "Automatic vehicle detection in infrared imagery using a fuzzy inference-based classification system," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1, pp. 53–61, Jan. 2001.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[5] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9626–9635.

[6] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO series in 2021," 2021, *arXiv:2107.08430*.

[7] P. Qin, Y. Cai, J. Liu, P. Fan, and M. Sun, "Multilayer feature extraction network for military ship detection from high-resolution optical remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 11058–11069, 2021.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[9] L. Sifre and S. Mallat, "Rigid-motion scattering for texture classification," 2014, *arXiv:1403.1687*.

[10] B. Azam, M. J. Khan, F. A. Bhatti, A. R. M. Maud, S. F. Hussain, A. J. Hashmi, and K. Khurshid, "Aircraft detection in satellite imagery using deep learning-based object detectors," *Microprocessors Microsystems*, vol. 94, Oct. 2022, Art. no. 104630.

[11] L. Kong, J. Wang, and P. Zhao, "YOLO-G: A lightweight network model for improving the performance of military targets detection," *IEEE Access*, vol. 10, pp. 55546–55564, 2022.

[12] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 1577–1586.

[13] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.

[14] S. Wang, Y. Du, S. Zhao, and L. Gan, "Multi-scale infrared military target detection based on 3X-FPN feature fusion network," *IEEE Access*, vol. 11, pp. 141585–141597, 2023.

[15] X. Du, L. Song, Y. Lv, and S. Qiu, "A lightweight military target detection algorithm based on improved YOLOv5," *Electronics*, vol. 11, no. 20, p. 3263, Oct. 2022.

[16] B. Koonce, "MobileNetV3," in *Convolutional Neural Networks With Swift for Tensorflow: Image Recognition and Dataset Categorization*. Berkeley, CA, USA: Apress, 2021, pp. 125–144, doi: 10.1007/978-1-4842-6168-2_11.

[17] H. Wang, D. Han, M. Cui, and C. Chen, "NAS-YOLOX: A SAR ship detection using neural architecture search and multi-scale attention," *Connection Sci.*, vol. 35, no. 1, pp. 1–32, Dec. 2023.

[18] P. Shan, R. Yang, H. Xiao, L. Zhang, Y. Liu, Q. Fu, and Y. Zhao, "UAVPNet: A balanced and enhanced UAV object detection and pose recognition network," *Measurement*, vol. 222, Nov. 2023, Art. no. 113654.

[19] Y. Li, Q. Fan, H. Huang, Z. Han, and Q. Gu, "A modified YOLOv8 detection network for UAV aerial image recognition," *Drones*, vol. 7, no. 5, p. 304, May 2023.

[20] Y. Qi, Y. He, X. Qi, Y. Zhang, and G. Yang, "Dynamic snake convolution based on topological geometric constraints for tubular structure segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6070–6079.

[21] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[22] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4784–4793.

[23] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7369–7378.

[24] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, Y. Li, B. Zhang, Y. Liang, L. Zhou, X. Xu, X. Chu, X. Wei, and X. Wei, "YOLOv6: A single-stage object detection framework for industrial applications," 2022, *arXiv:2209.02976*.

[25] C.-Y. Wang, A. Bochkovskiy, and H.-Y.-M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 7464–7475.

[26] G. Jocher, A. Chaurasia, and J. Qiu. (Jan. 2023). *YOLO by Ultralytics*. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] C. Feng, Y. Zhong, Y. Gao, M. R. Scott, and W. Huang, "TOOD: Task-aligned one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3490–3499.

[28] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, Nov. 2018.

[29] X. Li, W. Wang, L. Wu, S. Chen, X. Hu, J. Li, J. Tang, and J. Yang, "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2020, pp. 21002–21012.

[30] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.

[31] S. Ma and Y. Xu, "MPDIoU: A loss for efficient and accurate bounding box regression," 2023, *arXiv:2307.07662*.

[32] H. Zhang, C. Xu, and S. Zhang, "Inner-IoU: More effective intersection over union loss with auxiliary bounding box," 2023, *arXiv:2311.02877*.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaisr, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, 2017, pp. 5998–6008. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[34] L. Yang, R. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," in *Proc. 38th Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11863–11874.

[35] D. Wan, R. Lu, S. Shen, T. Xu, X. Lang, and Z. Ren, "Mixed local channel attention for object detection," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106442.

[36] M.-H. Guo, C.-Z. Lu, Q. Hou, Z. Liu, M.-M. Cheng, and S.-M. Hu, "SegNeXt: Rethinking convolutional attention design for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 1140–1156.

[37] H. Huang, Z. Chen, Y. Zou, M. Lu, and C. Chen, "Channel prior convolutional attention for medical image segmentation," 2023, *arXiv:2306.05196*.

**YU YOU** received the B.E. and M.E. degrees in mechatronical engineering from Beijing Institute of Technology, Beijing, China, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree. His research interests include mechanical structure designs, deep learning, target detection, and tracking.

**JIANZHONG WANG** received the B.E., M.E., and Ph.D. degrees from Nanjing University of Science and Technology, Nanjing, China. From 1990 to 2002, he was with Wuhan University of Technology, Wuhan, China, where he is currently a Professor of mechanical and electrical engineering. Since 2002, he has been with Beijing Institute of Technology, Beijing, China, where he is currently a Professor with the School of Mechatronical Engineering and the State Key Laboratory of Explosion Science and Technology. His current research interests include intelligent systems, unmanned ground vehicles, and multi-robot cooperative technology.

**ZIBO YU** was born in 1998. He received the B.E. degree in mechatronical engineering from Beijing Institute of Technology, in 2020, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include target detection and tracking.

**YONG SUN** received the B.E. degree from Beijing Institute of Technology, Beijing, China, in 2017, where he is currently pursuing the Ph.D. degree. His research interests include computer vision and deep learning.

**YIGUO PENG** received the B.E. degree from Beijing Institute of Technology, Beijing, China, in 2021, where he is currently pursuing the M.E. degree. His research interests include camouflage object detection, computer vision, and deep learning.

**SHENG ZHANG** received the B.E. degree in mechatronical engineering from Beijing Institute of Technology, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree. His current research interests include unmanned ground vehicles (UGV), deep reinforcement learning, and simulation technology.

**SHAOBO BIAN** received the B.E. degree in mechatronic engineering from Beijing Institute of Technology, China, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include human pose estimation and object recognition.

**ENDI WANG** received the B.E. degree in electronic information engineering from the Ocean University of China, Qingdao, Shandong, in 2021. He is currently pursuing the M.E. degree with Beijing Institute of Technology. His research interests include image segmentation and deep reinforcement learning.

**WEICHAO WU** received the B.E., M.E., and Ph.D. degrees from Northwestern Polytechnical University, Xi’an, China. From 2014 to 2016, he was with Northwestern Polytechnical University. Since 2016, he has been with Beijing Institute of Technology, Beijing, China. His current research interests include intelligent unmanned systems and smart munitions.