GCD-YOLOv5: An Armored Target Recognition Algorithm in Complex Environments Based on Array Lidar

Jian Dai, Xu Zhao, Lian Peng Li, and Xiao Fei Ma

***Abstract*—For the recognition of armored targets in complex battlefield environments, how to reduce missed and false alarms while achieving real-time is an urgent issue. To this end, the GCD-YOLOv5 algorithm is innovatively proposed. Firstly, array lidar is used to acquire the armor target data. Secondly, the armor target data is expanded with an improved GAN(Generative Adversarial Network) to increase the diversity of training data. Afterward, the expanded dataset is fed into the GCD-YOLv5(You Only Look Once) for training. And the GCD-YOLOv5 is reflected in the following aspects. Firstly, the CBAM(Convolutional Block Attention Module) and the multi-scale feature fusion are added to improve the feature extraction capability and detection efficiency, increasing the recognition capability of small and obscured targets. Secondly, combining with DETR(Detection Transformer) to lighten YOLOv5 to achieve the real-time requirement. Thirdly, the YOLOv5 loss function and prediction box filtering method are improved to increase the detection accuracy and the confidence of the detection boxes. The experimental results show that the GCD-YOLOv5 algorithm has higher accuracy and real-time, the mAP(mean Average Precision) can reach 99.7%, and fps is 68.56% higher compared to YOLOv5, which significantly improves the recognition capability of armored targets in complex battlefield environments.**

***Index Terms*—Armor target, target recognition, GAN, CBAM, DETR, YOLOv5.**

I. Introduction

With the complexity of the battlefield environment, especially the continuous development of new interference technologies such as stealth coatings, cloaking materials, and infrared interference. The detection and recognition capabilities of traditional millimeter-wave radar and infrared sensors are reduced. The ability of array lidar to acquire geometric information and profile characteristics of targets in the scanned area makes it the primary means of detecting armored targets in complex battlefield environments. At present, the array-based lidar target recognition algorithms for target feature extraction and recognition mostly use the traditional manual selection of target geometric features or local features detection methods. Among them, the statistical histogram of the distance image is often analyzed by the height statistical feature method for the distance image [1], which makes a large leakage and false alarm situation occur in complex environments and the real-time performance is poor. The deep learning-based approach can effectively solve the problems of missed alarms, false alarms, and poor real-time performance with the powerful feature extraction and learning capabilities of multi-layer convolutional neural networks.

In recent years, Transformer, as a deep neural network model based on the self-attentive mechanism, was initially mainly used in the field of NLP(Natural Language Processing). However, with the impressive achievement of the Transformer in the field of NLP, more and more researchers have carried out Transformer-related research and gradually applied it to the field of computer vision. Chen [2] *et al.* proposed a pixel regression prediction model based on Transformer and achieved good classification results in the field of image classification. Dosovitskiy [3] *et al.* proposed a Vit-based Transformer model that utilizes only pure Transformer and achieves the best results on multiple publicly available datasets for image recognition. Carion [4] *et al.* of the Facebook AI team took advantage of the Transformer's ability to simplify the process of object detection to construct a new method that treats object detection as a direct ensemble prediction problem. Although it achieves extremely good detection results, it needs to be supported by a large number of data samples. And for the specific application to the problem of identifying armored targets in complex battlefield environments, domestic and international scholars have conducted extensive research. Deng [5] *et al.* proposed a holistic nested convolutional network based on a multi-pyramid pooling model, which can effectively improve the detection and recognition accuracy of armored targets in complex environments by introducing the idea of dilated convolution and feature fusion. However, the high complexity of the algorithm model leads to a large amount of computation and the real-time performance cannot be effectively satisfied. Wang [6] *et al.*

Manuscript received 7 May 2022; revised 1 June 2022; accepted 19 June 2022. Date of publication 22 June 2022; date of current version 30 June 2022. This work was supported in part by the National Key Research and Development Project under Grant 2020YFC1511702, in part by the Beijing Information Science and Technology University Research Fund Project under Grant 2022XJJ10, in part by the National Natural Science Foundation of China under Grants 61971048, 61771059, and 61801032, and in part by the Beijing Scholars Program. *(Corresponding author: Xu Zhao.)*

Jian Dai, Xu Zhao, and Lian Peng Li are with the Beijing Information Science and Technology University, Beijing 100192, China, and also with the Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing Information Science and Technology University, Beijing 100192, China (e-mail: hbjsdj970321@163.com; zhaoxu00@126.com; llpstudy@163.com).

Xiao Fei Ma is with the Beijing Institute of Control and Electronic Technology, Beijing 100038, China (e-mail: jzys8000@163.com).

Digital Object Identifier 10.1109/JPHOT.2022.3185304

proposed an improved algorithm for the problems of the Faster R-CNN algorithm in the detection of small-scale tank armor targets. The improved algorithm achieved good detection results for tank armor targets of multiple scales, and the detection accuracy and speed were better than the original Faster R-CNN algorithm. But it has some difficulties in detecting obscured targets. Cheng [7] *et al.* proposed an end-to-end cross-scale feature fusion(CSFF) framework for remote sensing images that contain a large number of targets with highly variable target sizes and inter-class similarity. This framework can obtain powerful and differentiated multi-level feature representations, which can effectively improve the target detection accuracy. But it increases the computational effort of the model and brings the problem of real-time degradation.

In summary, the above methods applied to armor target recognition in complex battlefield environments have the problems of complex model structure, low training and detection efficiency, and insufficient real-time performance. In the face of small targets and obscured targets, there are a large number of missed and false alarms. In contrast, this paper uses array lidar as a detection means and deep learning methods to recognize armored targets in the scanned area. Firstly, the dataset is expanded using an improved GAN network. Secondly, the CBAM attention mechanism is added and multi-scale feature fusion is performed on the extracted features. After that, the loss function calculation method and the prediction box filtering method of the YOLOv5 algorithm are improved. Finally, YOLOv5 is lightened by combining DETR. The above measures can effectively solve the problems of low training and detection efficiency, and lack of real-time performance. At the same time, they solve the problems of missed and false alarms in the detection of small targets and obscured targets, thus realizing the recognition of armored targets in complex environments based on array lidar.

II. Array Lidar Scanning Imaging

With the complexity of the environment and the development of countermeasure technologies such as jamming and stealth, the recognition of armored targets requires more refined imaging. Array lidar can achieve more refined imaging of armored target areas. Array lidar scanning imaging is achieved with the help of steady-state rotational scanning or linear scanning motion. The scanning field of view is shown in Fig. 1. Where α is the array lidar scanning angle, and β is the array lidar field of view. The two-dimensional distance image data of $\mathrm{m}\times\mathrm{n}$ can be obtained by the steady-state rotational scanning or linear scanning motion.

In the process of armor target distance image data acquisition, the unstable jitter situation of the projectile during the flight will bring a lot of measurement noise. And the array lidar will bring the problems of graphic distortion and resolution reduction when the flight speed and scanning speed are unstable. These unavoidable factors will bring great difficulties to the subsequent image processing and target recognition. And due to the graphics distortion and resolution reduction, it will cause the problem of target characteristics of the sampled data is not obvious, resulting in missed alarms and false alarms. At the same time, in the detection process, the complex background and environmental noise interference can also lead to missed alarms and false alarms. Similarly, the complex background, a large amount of measurement noise, and environmental noise can also lead to a large amount of calculation of the target detection process, resulting in poor real-time target detection problems. Its scanning target field of view is shown in Fig. 2.

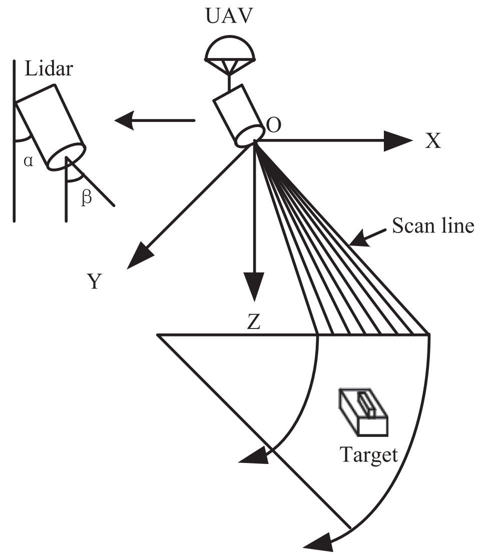

Fig. 1. Array lidar scanning field.

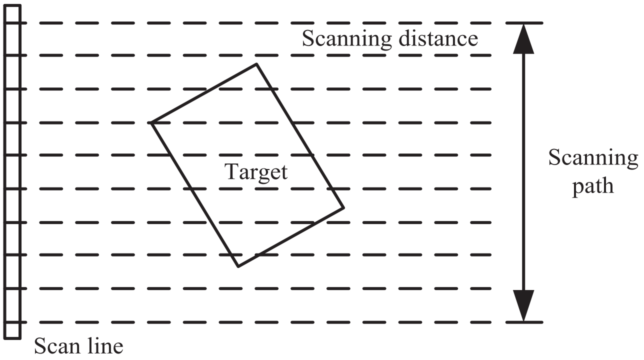

Fig. 2. Scanning armor target field.

III. GCD-yolov5 Model

In this section, we first analyze the original yolov5 model and the DETR model. Based on the above, we improve the yolov5 model and combine it with the DETR model to propose the GCD-YOLOv5 model. The GCD-YOLOv5 model consists of some elements as follows. Firstly, the expansion of the dataset samples is achieved by using the improved GAN network. Secondly, CBAM is added to YOLOv5 and multi-scale feature fusion is performed on the extracted features. Then, the DETR structure was innovatively integrated into YOLOv5. Finally, the loss function calculation method and the prediction box filtering method of YOLOv5 are improved.

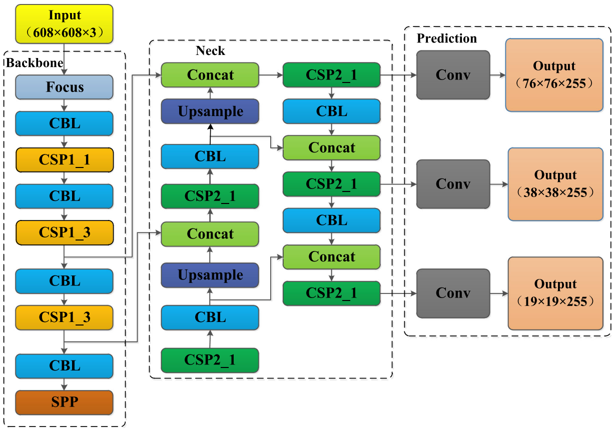

Fig. 3. YOLOv5 network structure diagram.

A. YOLOv5

The original YOLOv5 mainly consists of four parts: Input, Backbone, Neck, and Prediction. Its network model structure is shown in Fig. 3.

Input consists of the following three parts: Mosaic data enhancement, image adaptive scaling, and adaptive anchor box calculation [8]. Among them, Mosaic data enhancement is performed by randomly selecting four images for scaling, cropping, lining up, and other operations, and finally stitching the images. This can enrich the training set data and improve the generalization ability of the trained model and the detection of small targets. Image adaptive scaling adaptively adds the least black edges to both ends of the images, which can transform the training set images input to the neural network training to a fixed size. It can effectively solve the problem of redundant image information during training and improve the inference speed of neural networks. And the adaptive anchor box calculation is to set the initial anchor box adaptively before starting the training, compare it with the real box continuously, and iterate the reverse update according to the difference between them to adjust the network parameters and reduce the loss function.

The Backbone network includes four parts: Focus processing, CBL layer, CSP(Cross Stage Partial) [9] structure, and SPP(spatial pyramid pooling) [10]. Among them, Focus processing is a slicing operation, which can ensure that the feature map increases the number of features of the image without changing the information of each feature. The CBL layer is a convolutional block, which consists of three network layers, Conv, Batch Normalization, and Leaky relu [11], and is mainly used to extract the features of the target and input these features into the next layer network. Two CSP structures are used in YOLOv5, the Backbone network uses the CSP1_X structure and the Neck network uses the CSP2_X structure [12]. Where X indicates that there are several residual components. The use of two CSP structures makes the algorithm lightweight and can reduce the computation while improving the model learning ability. The SPP consists of three components: Conv, max-pooling, and concat, whose role is mainly to greatly increase the perceptual field of feature extraction with no impact on the inference speed, which can be used as an important feature for the network to separate the context [13].

The Neck network uses a structure of FPN(Feature Pyramid Networks) [14] combined with PAN(Path Aggregation Network) [15] as the fusion part of the network. A top-down FPN structure and a feature pyramid with two bottom-up PAN structures [16] are used. It is mainly used to mix and combine the extracted features and pass them to the prediction layer to enhance the network feature fusion [17].

On the prediction side of YOLOv5, GIOU [18] was used as the loss function to filter the target box by NMS(non-maximal suppression) [19].

B. DETR

The DETR structure consists of Encoder, Decoder, and Prediction, as shown in Fig. 4. In the Backbone part, a conventional CNN(Convolutional Neural Network) is used to learn the features of the input image and send them to Encoder for position encoding. In the Encoder part, firstly, the feature map output from Backbone is dimensionally compressed, and the C×H×W dimensional feature map is convolved by a 1×1 convolution kernel to obtain a d×H×W dimensional feature map by compressing the number of channels C to d. Next, the feature map is serially transformed to compress the spatial dimension H×W to HW to obtain a 2-dimensional feature map of d×HW. Finally, the 2-dimensional feature map is encoded with positional encoding for position encoding. The Encoder part contains 6 layers, each layer contains 8 self-attentive modules and FFN(Feed Forward Network). The decoder part also contains 6 layers, each layer contains 8 self-attentive modules, 8 co-attentive modules, and FFN. Decoder extracts feature from the feature map output by Encoder, and Decoder embeds a small number of a fixed number of positions into Object Queries as input and participates in the output. Finally, the output of the Decoder is passed to FFN for network detection of class and location or no object class(no object).

The introduction of the DETR attention module enables the model to selectively focus on the effective part of the input to improve the target feature learning of the model [20]. And at the same time, unlike the traditional Transformer, DETR processes all the Object Queries at once during the feature map processing, all the predictions are output at once, instead of left-to-right one by one. This greatly saves the efficiency of model training and facilitates the goal of model lightweight.

C. GCD-YOLOv5

In order to increase the sample diversity of the experimental dataset and improve the robustness of the trained model, as well as to solve the problem that a large amount of data is required to support DETR to achieve excellent results as mentioned in the literature [4]. This paper uses an improved GAN network to achieve the expansion of the dataset samples. In order to strengthen the sensitivity of certain important feature channels, effectively improve the feature extraction ability and detection efficiency of the algorithm, and also increase the recognition ability of small armored targets and obscured targets. This paper

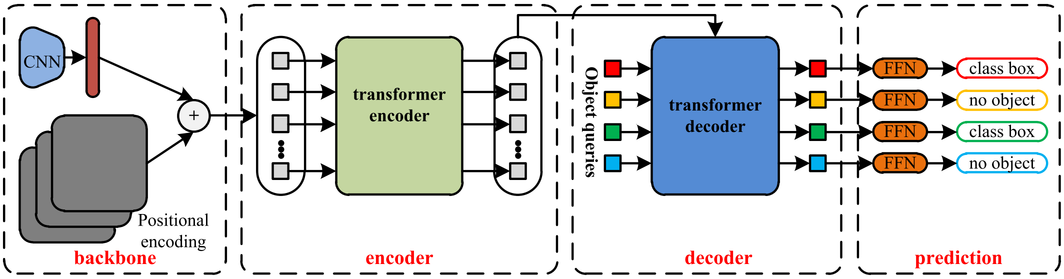

Fig. 4. DETR network structure diagram.

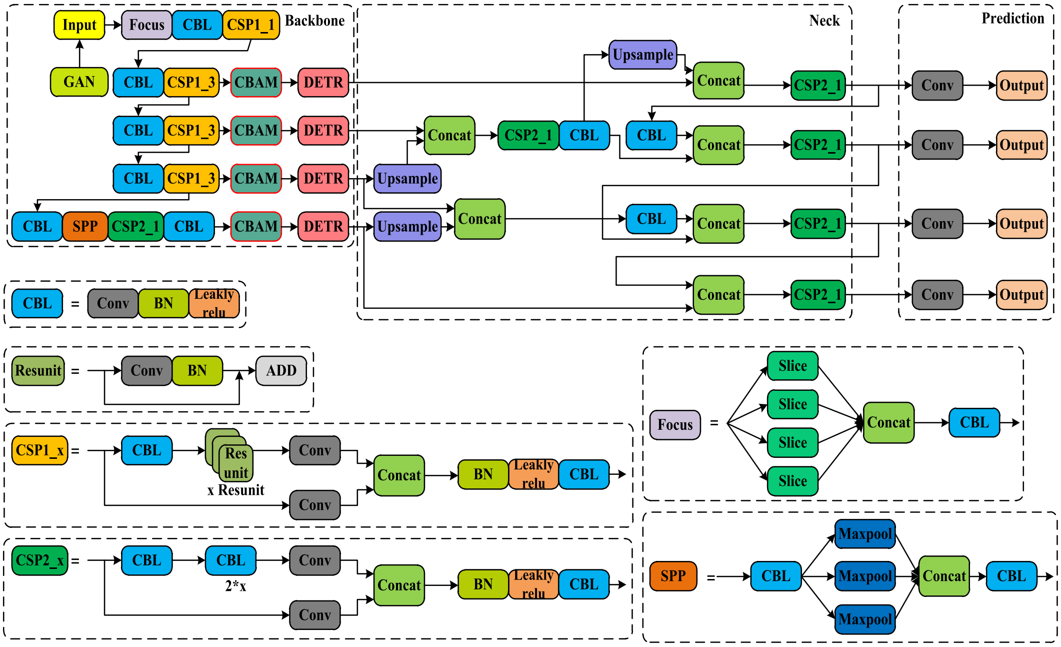

Fig. 5. GCD-YOLOv5 network structure diagram.

adds the CBAM attention mechanism and the multi-scale feature fusion implementation of the extracted features. In order to meet the needs of high real-time armor target recognition, this paper incorporates the DETR model. In order to improve the detection accuracy and confidence of the detection box, this paper is achieved by improving the loss function calculation method and the prediction box filtering method of the YOLOv5 algorithm. The GCD-YOLOv5 model structure consists of four parts: Input, Backbone, Neck, and Prediction, as shown in Fig. 5.

1) Add the multidimensional attention mechanism CBAM to the Backbone network of YOLOv5 [22].
2) Introduce the multi-scale feature fusion module to the Neck network of YOLOv5.
3) Introduce ***CIOU_Loss*** [23] loss function calculation and WBF(Weighted Boxes Fusion) [24] prediction box filtering method in Prediction of YOLOv5.
4) The DETR structure is combined to apply to the needs of armor target recognition requiring high real-time.

IV. Algorithm Flow

The flow chart of the algorithm in this paper is shown in Fig. 6. Firstly, the armor target scene information is obtained by scanning with an array lidar. Next, the armor target scene distance image data is obtained by data preprocessing. Then, the training dataset is expanded using an improved GAN [21]. After that, the GCD-YOLOv5 algorithm is constructed based on the YOLOv5 target detection network and DETR model, which mainly includes four parts, Input, Backbone, Neck, and Prediction, and the details are as follows.

A. CBAM Attention Mechanism

Adding the attention mechanism CBAM to the Backbone network of YOLOv5 can effectively improve the detection speed of the network trained out models while improving the feature extraction ability and detection accuracy of the network. And added as a multidimensional attention mechanism module combining both channel attention module and spatial attention module [25]. The former focuses on what features of the input image are meaningful, while the latter focuses on where features are meaningful. The specific model structure is shown in Fig. 7.

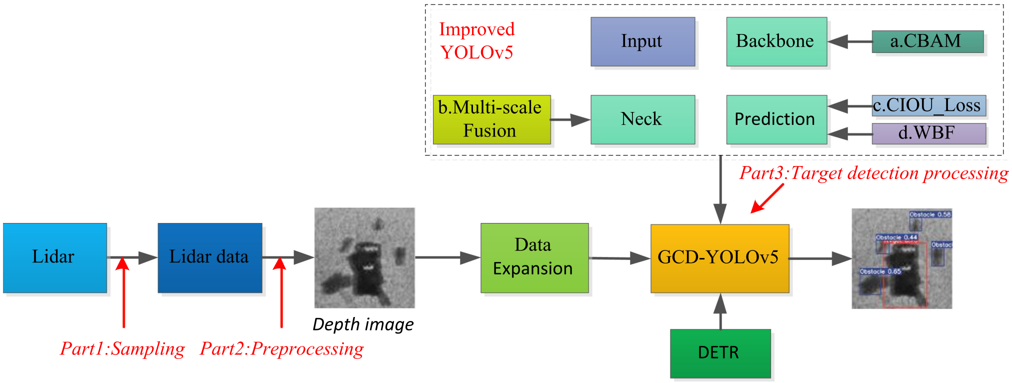

Fig. 6. Algorithm flow chart.

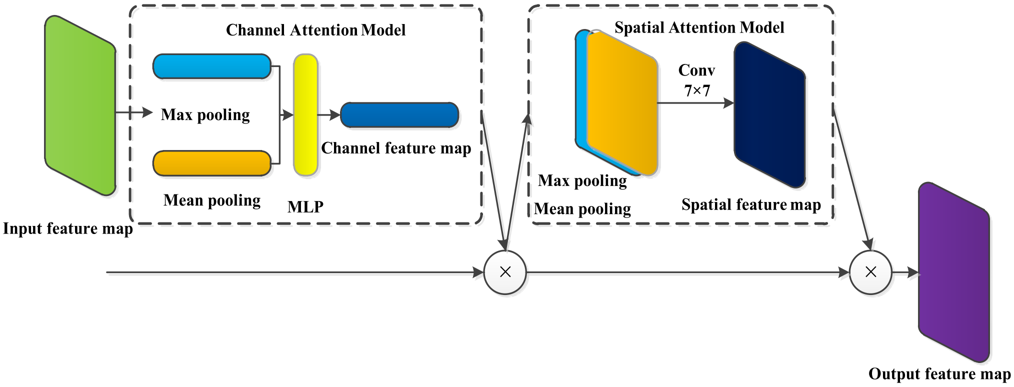

Fig. 7. CBAM network structure diagram.

The Channel Attention Module(CAM) extracts the spatial information of the feature map by summing up the input feature map $\boldsymbol{G}$ through average pooling and maximum pooling to obtain two spatial context description feature maps $\boldsymbol{G^c_{avg}}$ and $\boldsymbol{G^c_{max}}$. The former is the average pooling feature and the latter is the maximum pooling feature. Then these two features are passed into the shared network hidden layer $\boldsymbol{MLP}$ [26] for processing, and finally, the attention channel feature map $\boldsymbol{H_c}$ [27] is obtained by the activation function $\boldsymbol{\sigma}$. The two layers of parameters in the multilayer perception model are represented by $\boldsymbol{\alpha_1}$, $\boldsymbol{\alpha_2}$ to obtain the channel attention calculation formula.

$$\begin{aligned} H_c(G) &= \sigma(MLP(AvgPool(G)) + MLP(MaxPool(G))) \\ &= \sigma\left(\alpha_2\left(\alpha_1\left(G^c_{avg}\right)\right) + \alpha_2\left(\alpha_1\left(G^c_{\max}\right)\right)\right) \end{aligned} \tag{1}$$

The input of the Spatial Attention Module(SAM) is the output feature map of the previous step of the channel attention module. Firstly, the module performs average pooling and maximum pooling on the input feature maps to obtain the aggregated channel information [28] for $\boldsymbol{G^s_{avg}}$ and $\boldsymbol{G^s_{max}}$. Secondly, the two feature maps are stitched into one feature map. Finally, a 7 × 7 convolution is used to generate a two-dimensional spatial feature map. The formula for calculating spatial attention is as follows.

$$\begin{aligned} H_s(G) &= \sigma\left(f^{7\times7}\left([AvgPool(G); MaxPool(G)]\right)\right) \\ &= \sigma\left(f^{7\times7}\left(\left[G^s_{avg}; G^s_{\max}\right]\right)\right) \end{aligned} \tag{2}$$

B. Multi-Scale Feature Fusion

In YOLOv5, the images are adaptively scaled to 608∗608 before being fed into the network for training. After continuous deepening of the network, five convolution kernels of 3∗3 with a step size of 2 can output feature maps of 304∗304, 152∗152, 76∗76, 38∗38, and 19∗19 after downsampling. Conventional YOLOv5 uses 76∗76, 38∗38, and 19∗19 feature maps, which correspond to 8∗8(608/76 = 8), 16∗16, and 32∗32 receptive fields for target detection. The receptive field is the size of the region where the pixel points on the feature map are mapped back to the input image, which indicates that small target detection requires a small receptive field. Therefore, this paper proposes a multi-scale feature fusion method based on YOLOv5 to retain a 152∗152 feature map, which corresponds to a 4∗4 receptive field for target detection.

Since the deeper network has a larger receptive field, it can learn stronger semantic information features, but the larger downsampling factor will bring about a loss of location information. The shallow network has a smaller receptive field, and its semantic information characterization ability is weak, but its location information characterization ability is strong. Meanwhile, the lack of information fusion in YOLOv5 leads to low feature information utilization, which is not conducive to model training. In this paper, a multi-scale feature fusion structure is proposed to address the above characteristics. As shown in Fig. 8, firstly, upsampling enables the rich location

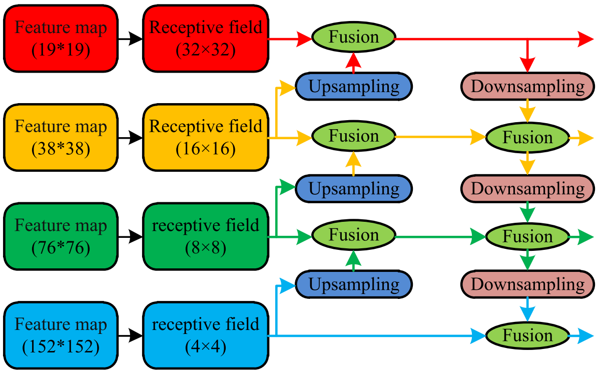

Fig. 8. Multi-scale feature fusion schematic.

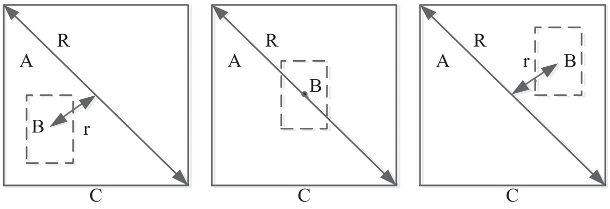

Fig. 9. Schematic diagram of CIOU_Loss.

information features from the shallow layer of the network to be passed upward to enhance the multi-scale localization capability. Subsequently, the rich semantic information features can be passed downward by downsampling to achieve feature cross-fusion to improve the multi-scale semantic expression capability. Finally, the model's two feature learning capabilities and target detection capability of the model are comprehensively improved.

C. CIOU_Loss Loss Function

Currently, YOLOv5's ***GIOU_Loss*** loss function calculation method solves the problem that the distance between two boxes cannot be reflected when the two boxes do not intersect, but it cannot discern the position of the predicted box when the target's predicted box is inside the target's real box. Therefore, the ***CIOU_Loss*** loss function calculation method with better results can be used instead. This method considers the distance information of the center point of the bounding box while considering the scale information of the width-to-height ratio of the bounding box, which can effectively solve the problems of the ***GIOU_Loss*** method. The specific calculation method is as follows.

As shown in Fig. 9, let the diagonal of the smallest outer rectangle C be ***R***, and the distance between the centers of the target real box A and the prediction box B be ***r***. Then ***CIOU_Loss*** is calculated as follows.

$$CIOU = IOU - \frac{R^2}{r^2} - \frac{v^2}{(1 - IOU) + v} \tag{3}$$

$$CIOU\_Loss = 1 - CIOU = 1 - IOU + \left(\frac{R^2}{r^2} + \frac{v^2}{(1 - IOU) + v}\right) \tag{4}$$

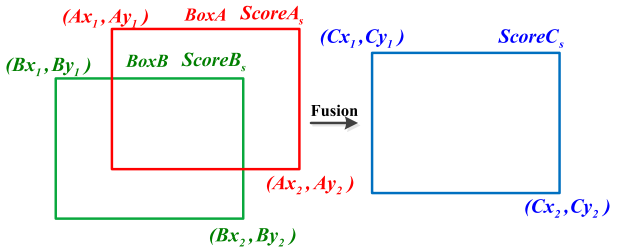

Fig. 10. WBF prediction box fusion diagram.

where ***v*** is a parameter characterizing the consistency of the aspect ratio of the target prediction box and is calculated as follows.

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w^p}{h^p}\right)^2 \tag{5}$$

$\boldsymbol{w^{gt}}$, $\boldsymbol{h^{gt}}$ denote the width and height of the target real box. $\boldsymbol{w^p}$ and $\boldsymbol{h^p}$ denote the width and height of the prediction box [29].

D. WBF Prediction Box Filtering Method

In the post-processing process of target detection, for filtering many prediction boxes, it is usually necessary to eliminate duplicate redundant prediction boxes and retain the information of the highest confidence prediction box. The common NMS approach is used in the YOLOv5 algorithm, which uses the intersection ratio ***IOU*** to suppress redundant detection boxes, where the overlapping region is the only factor that often produces false suppression for the occlusion case. In contrast, this paper uses the WBF prediction box filtering approach. It takes into account the role of each prediction box in the generation of detection boxes, assigns a weight to each prediction box based on the confidence score, and generates the coordinates of the weighted fusion box. The confidence of the fusion box is the average confidence of all prediction boxes. The specific formula is as follows.

$$Cx_1 = \frac{Ax_1 \times A_s + Bx_1 \times B_s}{A_s + B_s} \tag{6}$$

$$Cx_2 = \frac{Ax_2 \times A_s + Bx_2 \times B_s}{A_s + B_s} \tag{7}$$

$$Cy_1 = \frac{Ay_1 \times A_s + By_1 \times B_s}{A_s + B_s} \tag{8}$$

$$Cy_2 = \frac{Ay_2 \times A_s + By_2 \times B_s}{A_s + B_s} \tag{9}$$

$$C_s = \frac{A_s + B_s}{2} \tag{10}$$

As can be seen in Fig. 10, the coordinates of the two boxes are fused to obtain a new box, using the box's score as a weight. Also the higher the score the higher the weight of the box, and the more it contributes to the process of generating new boxes. Where, (Ax_1, Ay_1), (Bx_1, By_1) are the coordinates of the upper left corner of the two fused boxes. (Ax_2, By_2), (Bx_2, By_2) are the lower right coordinates of the two fused boxes. (Cx_1, Cy_1), (Cx_2, Cy_2) are the top-left and bottom-right coordinates of the generated fusion box. Compared with the NMS strategy, which

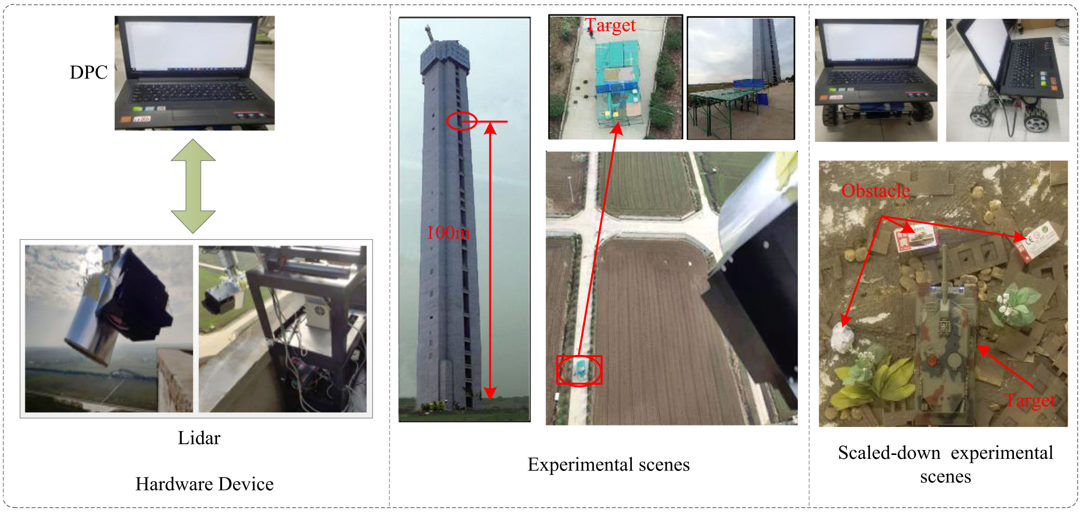

Fig. 11. Experimental data collection device and environment.

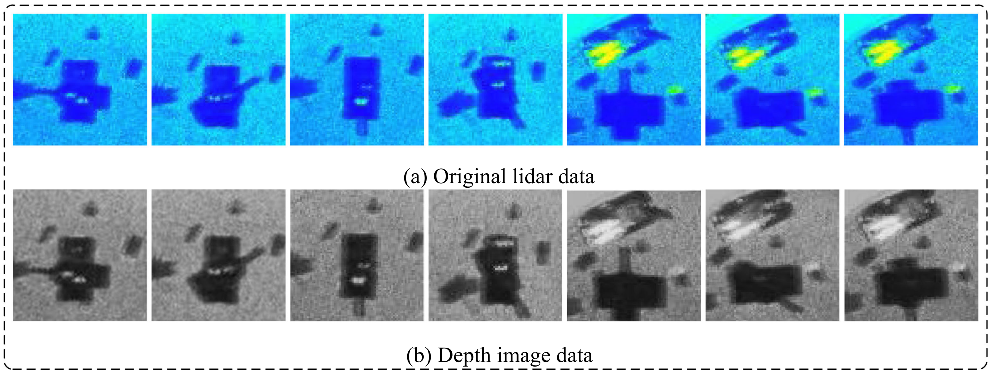

Fig. 12. Armored target data.

falsely detects two overlapping targets as a single target, the WBF strategy detects both targets correctly, which effectively reduces the probability of missing similar targets to a certain extent, and has higher localization accuracy and confidence level.

V. Experiments and Analysis

A. Training Data Collection

The samples of the experimental dataset are divided into 3 categories: armored targets, obstacles, and ground background. The armor target sample was simulated using scaled-down experiments to simulate the distance image of the armor target in different terrains and different attitudes at 100–60 m altitude. The obstacle samples simulate the distance images of hills and trees. Meanwhile, to increase the number of samples, the improved GAN data augmentation is used to expand the dataset. Since the armored targets and obstacles can be placed at arbitrary angles and positions, the training and testing samples are randomly rotated, cropped, and scaled, with a total of 3000 samples. And 70% of the obtained 3000 samples are randomly selected as training samples, 20% as validation samples, and 10% as test samples. Fig. 11 shows the data acquisition device, the experimental scenes, and the scaled-down experimental scenes. Fig. 12 shows the collected raw array lidar data and the processed distance image data.

B. Training Data Expansion

The full name of DCGAN is Deep Convolution Generative Adversarial Networks, which has powerful feature extraction capability, thus improving the effectiveness of unsupervised learning of generators. The network structure of DCGAN is shown in Fig. 13, where FC is the fully connected layer, BN is the batch normalization layer, Deconv is the deconvolution layer, and both ReLU and Tanh are nonlinear activation functions. Compared with the original GAN, the whole network removes the fully-connected layer and uses the convolutional layer directly instead. Meanwhile, the discriminative model is almost symmetric to the generative model. The discriminator uses convolutional steps instead of spatial pooling, and the generator uses the deconvolution operation to achieve upsampling to expand the dimensionality of the data and obtain better training stability. As for the generative network, its purpose is to generate pictures, the input is normally distributed in random numbers, and the output is fake pictures. For the discriminative network,

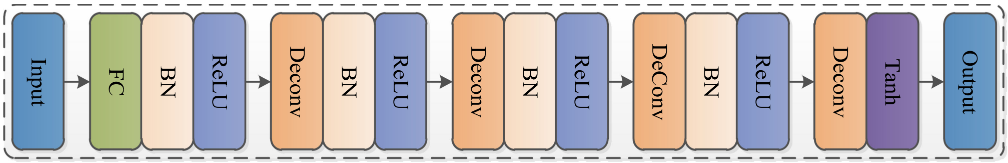

Fig. 13. DCGAN network structure diagram.

its purpose is to judge the authenticity of the input pictures, the input is fake pictures, and the output is the judgment result.

Its network training process can be represented as, assuming the existence of n samples, its network training process is transformed into the process of finding the network parameters whose loss function $L(z, \widehat{z})$ is reduced to the lowest.

$$z = (z_1, z_2, z_3, \ldots z_n) \tag{11}$$

$$\widehat{z} = (z_1, z_2, z_3, \ldots z_n) \tag{12}$$

In the above equation, n is the total number of samples, z is the true output of the network, z is the ideal output of the network, z_n is the true output of the nth sample, z_n is the ideal output of the nth sample.

The loss function of each training batch is the actual value of the jth batch and z_j is the predicted probability of the model. The loss function of the model is obtained by summing the loss functions of each batch and then averaging them.

$$L(z_j, z_j) = -\frac{1}{N}\sum_{j=1}^{N}[z_j \log z_j + (1 - z_j)\log_2(1 - z_j)] \tag{13}$$

In order to find the minimum weight w and bias b that can reduce the loss function, gradient descent is generally used to update the weights w and bias b in each network layer.

$$w' = w{-}\eta\frac{\partial L}{\partial w} \tag{14}$$

$$b' = b{-}\eta\frac{\partial L}{\partial b} \tag{15}$$

In the above equation, η is the learning rate, L is the loss function calculated in equation (13), w' is the updated weight. b' is the updated bias.

C. Experimental Analysis

In this paper, the experiments are based on the deep learning framework Pytorch, running on ubuntu 16.04, and the specific parameters configured as shown in Table I.

In order to ensure the correctness of the experimental algorithm model comparison, the experiment uses the same hyperparameters for the training, validation, and testing of different algorithm models. Among them, the initial learning rate is 0.01, the weight decay is 0.0005, the Batch_size is 8, and the IOU is 0.2. The details are shown in Table II.

TABLE I
TRAINING ENVIRONMENT CONFIGURATION

Parameter	Configuration
Operating System	Ubuntu 16.04
Video memory	8G
RAM	8G
GPU	NVIDIA GeForce GTX 1050 Ti
GPU acceleration environments	CUDA10.1
Training framework	Pytorch

TABLE II
EXPERIMENTAL HYPERPARAMETERS CONFIGURATION

Parameter	Value
Learning rate	0.01
Weight_decay	0.0005
Batch_size	8
Iou	0.2

The evaluation metrics of this experiment are mainly measured by Precision, recall, and mean average precision, which are ***precision***, ***Recall***, and ***mAP***. The calculation formula is shown below.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

In the above equation, ***TP*** indicates that the target is an armored target and the network model detects the result as an armored target. ***FP*** indicates that the target is not an armored target and the network model detects the result as not an armored target. ***FN*** indicates that the target is an armored target and the network model detects the result as not an armored target. ***Precision*** indicates how many of the samples detected as armored targets are true armored targets, reflecting the question of whether the detection results are accurate. ***Recall*** indicates how many armored targets are correctly detected in the total sample of armored target images, reflecting the question of whether the detection of armored targets is complete. The AP is the value of the area of the curve enclosed by the ***Precision*** and ***Recall***. The ***mAP*** is the average of the learned precision means for all categories. denotes the value of AP calculated for all image datasets in each category when the intersection ratio ***IOU*** is set to 0.5, and then all categories are averaged. The ***precision*** and ***recall*** of the trained target detection model are shown in Figs. 14 and 15.

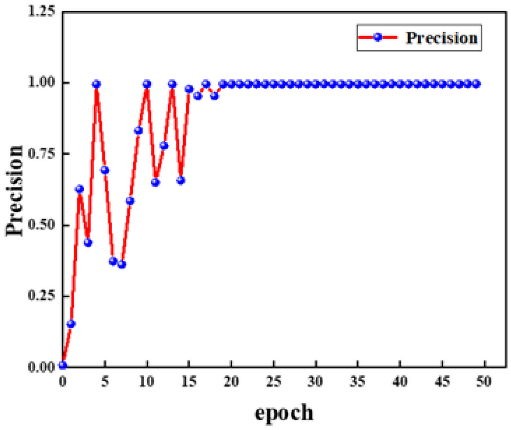

Fig. 14. Target detection precision.

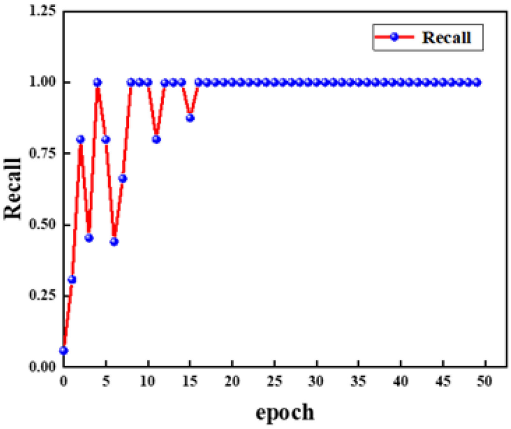

Fig. 15. Target detection recall.

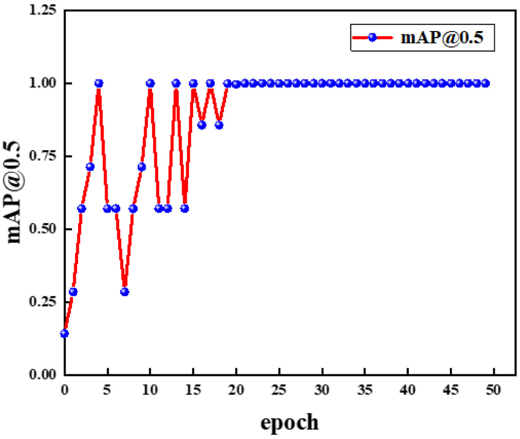

Fig. 16. mean Average Precision.

From Figs. 14 and 15, it can be seen that the ***Precision*** and ***recall*** of the GCD_YOLOv5 target detection network model stabilize and reach the fit state after about 25 epochs of training. And at this time, it can be seen from Fig. 16 that the determined by the target detection ***precision*** and ***recall*** also tends to be stable, which verifies the accuracy of the GCD-YOLOv5 target detection network model.

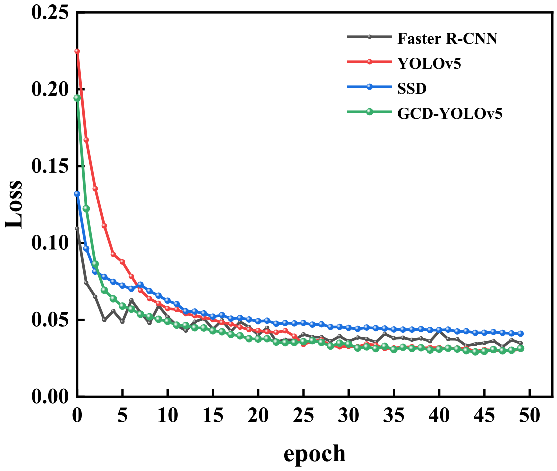

Fig. 17. Loss function.

And then, from Figs. 14, and 16, it can be seen that in the first 20 epochs of training, there are large fluctuations in ***Precision***, ***Recall***, and . This is due to the expansion of the training dataset using the improved GAN network, which generates a dataset with irregular distribution and a large resolution span. As the model learns fewer features in the early stage, there is a possibility of missed recognition and false recognition. But as the training continues, the model learns more and more target features, ***Precision***, ***Recall***, and will gradually stabilize. And irregularly distributed data sets with a large resolution span can improve the robustness of the trained model.

As can be seen from Fig. 17, after about 20 epochs, the training of the GCD-YOLOv5 target detection network model tends to converge and the loss function drops to below 0.04. Since the GCD-YOLOv5 model uses the CBAM attention mechanism and the multi-scale feature fusion strategy to obtain more comprehensive and detailed feature information, the confidence loss value is the lowest. And due to the global sensing capability of DETR and the advantage of parallel information processing, while considering the improved loss function calculation method and the prediction box filtering method, the position loss value decreases rapidly. In contrast, SSD [30](Single Shot MultiBox Detector), Faster R-CNN [31], and YOLOv5 algorithms converge relatively slowly and tend to converge after about 30 epochs, which verifies the rapidity of convergence of the GCD-YOLOv5 target detection network model.

Meanwhile, in order to test the performance of the model trained by the GCD-YOLOv5 target detection algorithm, the experiments compared SSD, Faster R-CNN, and YOLOv5 target detection algorithm, mainly using ***mAP*** as evaluation index, and compared the detection speed of the trained model, as shown in Table III.

As can be seen from Table III, the ***mAP*** of the GCD-YOLOv5 target detection algorithm reaches 99.58%, which is 19.94%, 14.26%, and 6.71% higher than the SSD, Faster R-CNN, and YOLOv5 algorithms. SSD uses multiple layers of feature maps as the resultant output leading to deeper network layers and weaker extracted armor target features, which is not conducive to

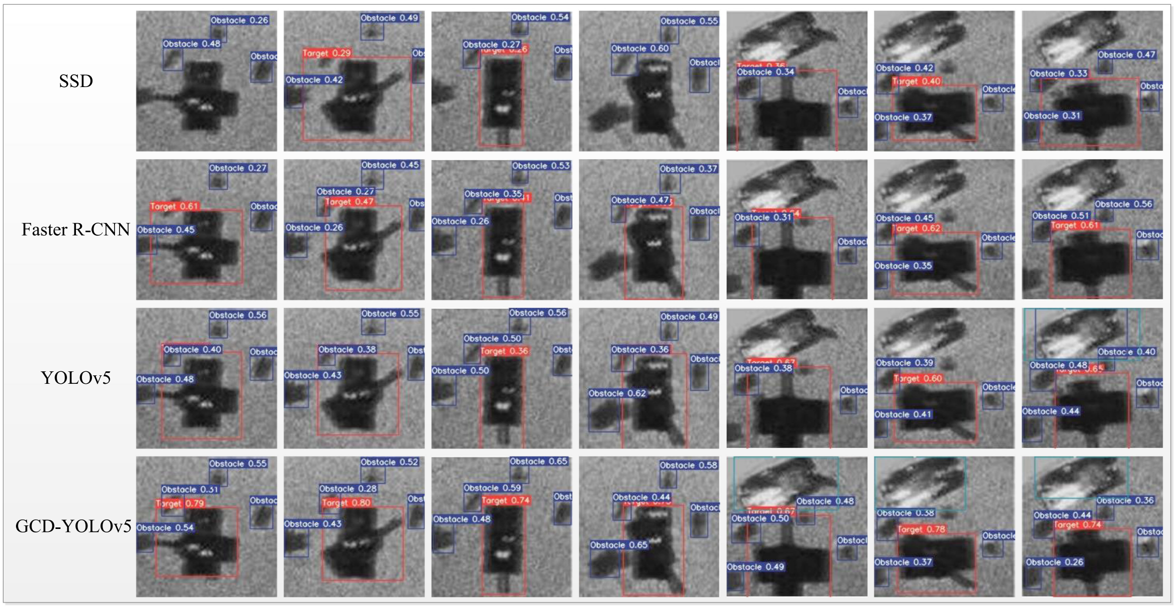

Fig. 18. Target detection results.

TABLE III
COMPARISON OF TARGET DETECTION MODEL RESULTS

Parameter	mAP	FPS	Time/epoch
SSD	79.64%	0.78	0.13h
Faster R-CNN	85.32%	2.24	0.23h
YOLOv5	92.87%	2.64	0.17h
GCD-YOLOv5	99.58%	4.45	0.09h

the detection of armored targets. Faster R-CNN due to multiple downsampling operations, resulting in the inability to effectively extract features for armored targets. YOLOv5 compared to the first two methods, ***mAP*** has significantly improved. And this paper's GCD-YOLOv5 target detection algorithm ***mAP*** is still improved based on YOLOv5, and the target detection model detects each frame in a shorter time, relative to the experimental comparison target detection algorithm has significantly improved the detection accuracy and detection time.

Meanwhile, it can be seen from Table III that the Faster R-CNN takes the longest time to train. Since Faster R-CNN is a two-stage network, the first stage uses the region suggestion network to get the candidate box regions of interest, and the second stage maps the candidate box regions of interest to the feature map through pooling for classification and location regression, which makes its training process slow. YOLOv5 is a one-stage network, which directly outputs classification and localization results, thus improving its training speed compared to Faster R-CNN. SSD is also a one-stage network, but its structure is simpler than that of YOLOv5, which can improve the training speed with the loss of certain detection accuracy. In this paper, the GCD-YOLOv5 has the advantage of global awareness and parallel information processing due to the introduction of DETR, which can greatly reduce the training time of the network and improve the learning efficiency of the network.

To further test the specific performance of the GCD-YOLOv5 target detection algorithm for detecting armored targets, the results of the model trained by the GCD-YOLOv5 target detection algorithm were experimentally tested under the experimental data set. The details are shown in Fig. 18.

From the detection results in Fig. 18, it can be seen that the trained model of the GCD-YOLOv5 target detection algorithm can be effectively identified for armored targets in complex battlefield environments with high accuracy and low false alarm rate. The confidence level of the trained target detection model is around 0.8, which verifies the correctness and effectiveness of the GCD-YOLOv5 target detection algorithm and the trained model.

The GCD-YOLOv5 target detection algorithm achieves these results by first adding the multidimensional attention mechanism module CBAM to the Backbone network of the YOLOv5 target detection algorithm. This makes the extracted armor target features given different weights and can obtain feature maps with different weighting channels. And, more comprehensive and detailed features are also obtained, thus enabling targeted training of the target detection network. Secondly, the multi-scale feature fusion strategy effectively solves the problem of difficult small target detection. From the iterative process of loss function during training in Fig. 17, we can see that the GCD-YOLOv5 target detection network model tends to converge after about 20 epochs of training, which greatly saves the network training time and improves the network learning efficiency due to the powerful parallel processing capability of DETR. Thirdly, the addition of the attention mechanism module CBAM can improve the relevance of the trained model and lighten the model, which can reduce the occurrence of missed and false alarms. Also, the improved loss function calculation method and prediction box filtering method can effectively avoid the occurrence of missed alarms when two targets overlap and improve the confidence of the target detection boxes.

Finally, from Table III, we can see that the GCD- YOLOv5 target recognition algorithm trains the model to detect approximately 4.45 frames per second, which can detect 3.67, 2.21, and 1.81 frames per second more than the experimental comparisons of SSD, Faster R-CNN, and YOLOv5 algorithms, significantly improving the real-time speed of the trained model to detect armored targets.

VI. Conclusion

In summary, the recognition of armored targets in complex environments overly relies on the accurate separation and information extraction of armored targets from the environmental background, and there are problems of missed alarms, false alarms, and poor real-time performance. In this paper, we propose a GCD-YOLOv5 algorithm for complex background armor target recognition, which improves the performance of the algorithm by fusing DETR and YOLOv5 structures, introducing an attention mechanism CBAM, incorporating a multi-scale feature fusion module, and using ***CIOU_Loss*** loss function calculation and WBF prediction box filtering method. The experimental results show that the GCD-YOLOv5 algorithm can achieve the requirement of real-time detection time while ensuring a high accuracy rate. The trained model for different complex backgrounds, armored targets can be effectively detected, there is no leakage and false alarm phenomenon, the average accuracy means value reaches more than 99.7%, and the fps is improved by 68.56% compared to the traditional YOLOv5 algorithm, which further verifies the correctness of the algorithm and model. The next step is to continue to lighten the model without affecting the performance of the algorithm and model and improve the recognition in real-time is a research direction afterward.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable suggestions.

References

[1] Y. Xu and W. Wang, “A method for single frame detection of infrared dim small target in complex background,” *J. Phys.: Conf. Ser.*, vol. 1634, no. 1, 2020, Art. no. 012063.

[2] C. F. R. Chen, Q. Fan, and R. Panda, “CrossViT: Cross-attention multi-scale vision transformer for image classification,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 357–366.

[3] A. Dosovitskiy, L. Beyer, and A. Kolesnikov, “An image is worth 16x16 words: Transformers for image recognition at scale,” in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–22.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 213–229.

[5] L. Deng and H. F. Li, “An improve multi-scale holistically-nested pooling semantics for object detection,” *Laser Infrared*, vol. 52, no. 2, pp. 295–304, 2022.

[6] Q. D. Wang, T. Q. Chang, and L. Zhang, “Automatic detection and tracking system of tank armored targets based on deep learning algorithm,” *J. Comput.-Aided Des. Comput. Graph.*, vol. 30, no. 12, pp. 2278–2291, 2018.

[7] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, “Cross-scale feature fusion for object detection in optical remote sensing images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 3, pp. 431–435, Mar. 2021.

[8] S. L. Tan, X. B. Bie, and G. L. Lu, “Real-time detection for mask-wearing of personnel based on YOLOv5 network model,” *Laser J.*, vol. 42, no. 2, pp. 147–150, 2021.

[9] C. Y. Wang *et al.*, “CSPNet: A new backbone that can enhance learning capability of CNN,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 390–391.

[10] K. M. He, X. Zhang, and S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[11] D. G. Lowe, “Distinctive image features from scale invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[12] X. L. Yang, W. X. Jiang, and H. Yuan, “Traffic sign recognition detection based on yolov5,” *Inf. Technol. Informatization*, vol. 46, no. 4, pp. 28–30, 2021.

[13] Q. P. Lin, Q. L. Zhang, and L. Xiao, “A remote sensing image target recognition method using improved YOLOv5 network,” *J. Air Force Early Warning Acad.*, vol. 35, no. 2, pp. 117–120, 2021.

[14] T. Y. Lin *et al.*, “Feature pyramid networks for object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[15] W. Wang *et al.*, “Efficient and accurate arbitrary-shaped text detection with pixel aggregation network,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 8440–8449.

[16] F. Lei, W. B. Gu, and W. Li, “Bidirectional parallel multi-branch convolution feature pyramid network for target detection in aerial images of swarm UAVs,” *Defence Technol.*, vol. 17, no. 4, pp. 1531–1541, 2021.

[17] X. D. Hu, X. Q. Wang, and X. Yang, “An infrared target intrusion detection method based on feature fusion and enhancement,” *Defence Technol.*, vol. 16, no. 3, pp. 737–746, 2020.

[18] H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 658–666.

[19] F. Y. Zhou, L. P. Jin, and J. Dong, “Review of convolutional neural network,” *Chin. J. Comput.*, vol. 40, no. 6, pp. 1229–1251, 2017.

[20] Y. L. Tang, H. P. Li, and W. D. Zhang, “Lightweight DETR-YOLO method for detecting shipwreck target in side-scan sonar,” *Syst. Eng. Electron.*, vol. 44, no. 6, pp. 1–13, 2022.

[21] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional neural network for remote sensing images,” in *Proc. Int. Conf. Image Graph.*, 2017, pp. 97–108.

[22] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[23] Z. Zheng *et al.*, “Enhancing geometric factors in model learning and inference for object detection and instance segmentation,” *IEEE Trans. Cybern.*, to be published.

[24] R. Solovyev, W. Wang, and T. Gabruseva, “Weighted boxes fusion: Ensembling boxes from different object detection models,” *Image Vis. Comput.*, vol. 107, pp. 1–6, 2021.

[25] Y. Wang and L. B. Liu, “Bilinear residual attention networks for fine-grained image classification,” *Laser Optoelectron. Prog.*, vol. 57, no. 12, pp. 171–180, 2020.

[26] C. Zhang, X. Pan, and H. Li, “A hybrid MLP-CNN classifier for very fine resolution remotely sensed image classification,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 140, pp. 133–144, 2018.

[27] X. Zhou and L. F. Chen, “Object detection of remote sensing image based on dual attention mechanism,” *Comput. Modernization*, no. 8, pp. 1–7, 2020.

[28] Y. Q. Yao, G. Cheng, and X. X. Xie, “Optical remote sensing image object detection based on multi-resolution feature fusion,” *Nat. Remote Sens. Bull.*, vol. 25, no. 5, pp. 1124–1137, 2021.

[29] H. Q. Zhang, Y. G. Ban, and L. L. Guo, “Detection method of remote sensing image ship based on YOLOv5,” *Electron. Meas. Technol.*, vol. 44, no. 8, pp. 87–92, 2021.

[30] W. Liu, D. Anguelov, and D. Erhan, “SSD: Single shot multibox detector,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[31] S. Ren, K. M. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2016.