IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 July 2023, accepted 26 July 2023, date of publication 1 August 2023, date of current version 7 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3300372

APPLIED RESEARCH

YOLOv5s_2E: Improved YOLOv5s for Aerial Small Target Detection

TAO SHI[1], YAO DING[1], AND WENXU ZHU[2]

[1]Tianjin Key Laboratory for Control Theory and Applications in Complicated Systems, School of Electrical Engineering and Automation, Tianjin University of Technology, Tianjin 300384, China

[2]College of Electrical Engineering, North China University of Science and Technology, Tangshan, Hebei 063210, China

Corresponding author: Tao Shi (st99@email.tjut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62103298, and in part by the Natural Science Foundation of Hebei Province under Grant F2018209289.

ABSTRACT To address the issues of low accuracy in existing small object detection algorithms, an improved network model algorithm called YOLOv5s_2E is proposed. This method first uses the k-means++ clustering algorithm to calculate the prior boxes of the Visdrone dataset. Then, it introduces Soft_NMS and combines it with EIoU to propose the EIoU_Soft_NMS algorithm to replace the non-maximum suppression (NMS) of the original network, improving the detection of objects that are occluded. The bounding box regression loss function uses Focal-EIoU, which speeds up model convergence and reduces loss. Additionally, a detection layer is added to the original detection head to unify the channel numbers, and with the dynamic head framework DyHead, the attention mechanism is integrated with the detector's head to further improve small object detection accuracy. Finally, the system robustness is improved by adjusting the ratio of data augmentation methods Mixup and Mosaic.Experimental results show that the proposed algorithm improves the mAP@0.5, mAP@0.5:0.95 and detection accuracy by 12.6%, 12.2%, and 20.5%, respectively, compared to the previous method on the VisDrone dataset. The parameter size only increases by 4%, and the weight file size increases by only 0.57MB, meeting the accuracy requirements for small object detection.

INDEX TERMS Data augmentation, DyHead, small object detection, soft_NMS, YOLOv5s.

I. INTRODUCTION

Object detection [1], [2], [3], [4] has always been a hot topic in the field of deep learning and has been widely used in various fields such as unmanned driving [5], [6], medical image lesion detection [7], and security systems [8], becoming one of the research directions of many scholars. Existing object detection algorithms can mainly be divided into two types: two-stage algorithms and single-stage detection algorithms. Two-stage algorithms, represented by Faster-RCNN [9], first extract object regions and then perform convolutional classification recognition on these regions. In contrast, single-stage detection algorithms, represented by YOLO [10] and SSD [11], directly perform regression using convolutional networks. Two-stage object detection can selectively choose samples to make the positive and negative samples more balanced, but it requires a lot of computing resources for location and classification tasks, leading to low efficiency. The single-stage detection algorithm is classified and detected at the beginning, which does not require more computing resources and greatly improves the processing speed, but also leads to a decrease in accuracy. These methods have become the mainstream of object detection, but there are still difficulties in detecting small objects. For example,YOLOv5 uses CSPDarknet-53 [12] as a backbone for feature extraction, but the final feature map size is small, and the pixel receptive field is large, which makes it easy to make location errors during detection. To address this, Jiaqi Wang, Kai Chen, and others [13] proposed the CARAFE upsampling operator to predict the upsampling kernel using the upsampling core prediction module, and then used the feature recombination module tocomplete upsampling, achieving a larger receptive field without introducing too much computational complexity. Sun et al. [14] proposed a RSOD algorithm based on

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

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

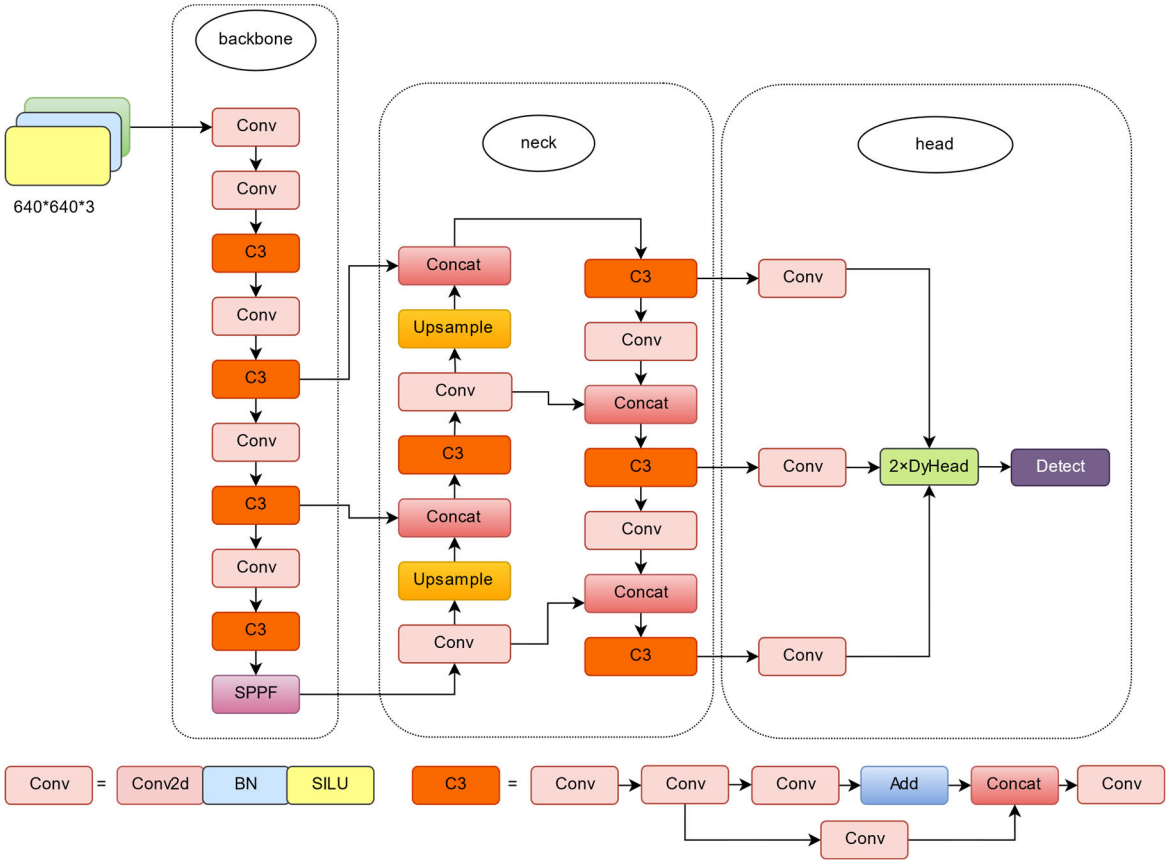

FIGURE 1. YOLOv5s_2E network structure diagram. The lower left and right corners are the structures of "Conv" and "C3" in the model.

YOLOv3 [15], which uses fine-grained information-rich shallow features to predict the positions of high-density small objects. Zhao et al. [16] proposed a lightweight real-time object detection network, Mixed YOLOv3-LITE, based on YOLO-LITE [17] that can be used with non-graphics processing units (GPUs) and mobile devices. Zhan et al. [18] introduced an attention mechanism and new anchor box sizes into the YOLOv5 network structure to preserve more information about small objects.

However, existing small object detection algorithms can still be optimized in the case of severe occlusion, and improving accuracy often comes at the cost of increased computational complexity. Consequently, this research proposes the YOLOv5s_2E algorithm, which principally encompasses the following tasks:

1) Using the K-means++ clustering algorithm to optimize the initial anchor boxes of the VisDrone dataset to achieve better location accuracy.

2) Introducing EIoU_Soft_NMS, which can improve detection results in situations where objects are occluded and prevent missed detections and false positives compared to the original NMS algorithm.

3) Introducing Focal-EIoU into the bounding box regression loss function to speed up network convergence and reduce loss, improving system inference accuracy.

4) By unifying the channel numbers in the detection head, integrating the attention mechanism with the detector's head using the dynamic head framework DyHead, our approach enhances detection accuracy without excessively increasing the number of parameters.

5) Introducing Mixup data augmentation and using it together with Mosaic, adjusting the ratio of the two to select the best method.

II. YOLOv5s_2E NETWORK DESIGN

A. OVERALL NETWORK STRUCTURE

YOLOv5s mainly consists of three parts: the backbone, neck, and head. The backbone refers to the network used for feature extraction, which extracts information from the image for later use. The neck is placed between the backbone and head to further utilize the features extracted by the backbone and improve the model's robustness. The head obtains the network output and makes predictions using the previously extracted features.

This paper puts forth the YOLOv5s_2E model algorithm, specifically designed for small object detection based on the proposed structure. The approach first employs the k-means++ clustering algorithm to compute the prior boxes of the dataset and then uses EIoU_soft_NMS to reduce the chances of missed detections and false positives. To expedite

network convergence and minimize loss, Focal-EIoU is introduced into the bounding box regression loss function, thereby enhancing system inference accuracy. Finally, three ordinary convolutional layers are added to standardize the channel numbers of the output from the neck, which is then fed into the dynamic head framework before being passed to the Detect layer. Figure 1 depicts the overall network model structure.

B. INITIAL ANCHOR BOX OPTIMIZATION

The initial anchor box size used in the original YOLOv5s algorithm is designed for detecting objects in the COCO dataset, and when it is not suitable for some datasets, the K-means clustering algorithm is used. The classic K-means clustering algorithm implementation steps are as follows:

1) Select K samples from the dataset as initial cluster centers $C = \{c_1, c_2, \ldots, c_k\}$.

2) Calculate the Euclidean distance between each sample and the cluster center, classify it, and add it to the class with the smallest Euclidean distance to the cluster center.

3) For each category c_i, recalculate the cluster center for each category, where n is the number of samples in each category.

4) Repeat steps 2 and 3 until the cluster center no longer changes.

Since the results of the K-means algorithm can vary depending on the initial point selection, the anchor box size calculated using this method may not achieve the desired effect, which can affect the results. Therefore, in this paper, we use the k-means++ clustering algorithm to recalculate the initial anchor boxes of the VisDrone dataset. The k-means++ clustering algorithm process is as follows:

1) Randomly select the first cluster center c_1.

2) Randomly select another point from the remaining points as the next cluster center and follow the mechanism that the farther away the distance, the greater the probability of selecting the new cluster center (roulette method), until k cluster centers are selected.

3) Calculate the distance between each samplepoint and the nearest cluster center $D(x)$.

4) Calculate the probability that each sample is selected as the next cluster center, and choose the sample with the maximum distance as the new cluster center with a certain probability. Repeat the above process until all K cluster centers are determined.

5) Use the K-means algorithm to calculate the final cluster centers for the K initial cluster centers. The clustering results are shown in Table 1. To verify the effectiveness of this method, a comparative experiment shown in Table 2 was added.

TABLE 1. Clustering results.

Detection target	Small target	In the target	Big target
Cluster anchor frame size	(4,8)	(12,12)	(42,35)
	(6,19)	(23, 53)	(51,75)
	(12,32)	(23,21)	(96,111)

TABLE 2. Comparison of results of different clustering algorithms.

clustering algorithm	mAP@0.5	mAP@0.5:0.95
k-means	30.2%	15.7%
k-means++	33.0%	18.0%

C. USING EIoU_Soft_NMS

Non-maximum suppression (NMS) is an effective method for obtaining local maximum values. It relies on the classifier to obtain multiple candidate boxes, sorts them according to the classifier's probability of classifying the obtained category, and the algorithm is shown in Equation (1).

$$s_i = \begin{cases} s_i, & IoU\ (M, b_i) < N_t \\ 0, & IoU\ (M, b_i) \geq N_t \end{cases} \tag{1}$$

In the process, the following steps are taken:

1) Sort all the box scores and select the highest score and its corresponding box;

2) Traverse the remaining boxes, and if the overlap area with the current highest score box is greater than a certain threshold, delete the box;

3) Select the next highest score from the remaining unprocessed boxes and repeat the above process.

For adjacent boxes with IoU≥NMS threshold, the traditional NMS method is to set their scores to 0, which is equivalent to discarding them. This may cause missed detections of the bounding boxes, especially in occluded scenes, as shown in Figure 2.

FIGURE 2. Occluded scene.

Both animals depicted in Figure 2 are objects to be detected, and there are two detected boxes with scores of 0.85 and 0.95, respectively. When utilizing the NMS

algorithm, the box with the highest score is the red box. However, if the overlap between the green box and the red box, determined by calculating the IoU, exceeds the set threshold, the green box will be deleted, leading to a situation where one animal is missed. Setting the threshold too low can lead to false positives.

Soft non-maximum suppression (soft_NMS) [19] improves on traditional NMS algorithm, and the algorithm is shown in Equation (2).

$$s_i = \begin{cases} s_i & IoU(M, b_i) < N_t \\ s_i e^{-\frac{IoU(M,b_i)^2}{\sigma}}, & IoU(M, b_i) \geq N_t \end{cases} \tag{2}$$

M is the current highest-scoring box, s_i is the current box to be detected, IoU is the intersection over union, N_t is the set threshold, b_i is the box to be processed, and the larger the IoU between b_i and M, the more the score of s_i will be reduced. For a box with an IoU greater than the threshold with the highest-scoring box, it is not deleted but replaced with a lower score to achieve better results. However, it only considers the overlap between two boxes, so the EIoU_Soft_NMS algorithm is proposed by combini- ng with EIoU [20]. It not only considers the overlap area, but also the distance between the center points and the true differences in length and width. The algorithm is shown in Equations (3) and (4).

$$s_i = \begin{cases} s_i, & EIoU(M, b_i) < N_t \\ s_i e^{-\frac{EIoU(M,b_i)^2}{\sigma}}, & EIoU(M, b_i) \geq N_t \end{cases} \tag{3}$$

$$EIoU = IoU - \frac{\rho^2(b, b^{gt})}{c^2} - \frac{\rho^2(\omega, \omega^{gt})}{c_\omega^2} - \frac{\rho^2(h, h^{gt})}{c_h^2} \tag{4}$$

b and b^{gt} are the center points of the predicted box and the ground truth box, respectively. ρ is the Euclidean distance between the two center points. c is the diagonal length of the minimum bounding rectangle that contains both the predicted box and the ground truth box. c_ω and c_h are the width and height of the minimum bounding rectangle that covers both boxes.

D. IMPROVED LOSS FUNCTION

IoU [21] is the intersection over union between the predicted box and the ground truth box, which reflects the detection performance and is insensitive to scale while possessing scale invariance. However, when the predicted box and the ground truth box do not overlap, according to Equation (5), the IoU is calculated as $IoU = 0$ and $Loss = 0$, which leads to the IoU being 0 and cannot accurately reflect the degree of overlap between the two boxes when they do not overlap. As a result, using the IoU as the loss function may not be the best choice for object detection tasks, particularly when there are numerous non-overlapping predicted boxes.

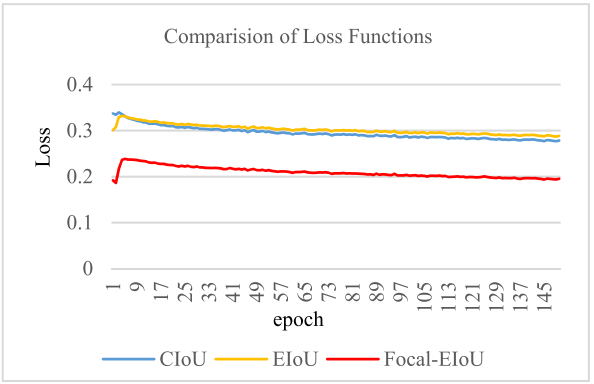

FIGURE 3. Comparison of loss functions.

$$IoU = \frac{A \cap B}{A \cup B}$$
$$L_{IoU} = 1 - IoU \tag{5}$$

YOLOv5 uses CIoU [22] as the bounding box regression loss function based on the original algorithm. The calculation formula is shown in Equation (6), which considers the overlap area, distance between center points, and aspect ratio. α is a weight parameter, υ is used to measure the similarity of aspect ratios and reflects the difference in aspect ratios, rather than the difference between width and height and their confidence.

$$CIoU = IoU - \left(\frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha\upsilon \right)$$
$$\upsilon = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$$
$$\alpha = \frac{\upsilon}{(1 - IoU) + \upsilon} \tag{6}$$

Although the existing method has shown some effectiveness, it suffers from ambiguity and does not adequately balance the difficulty of detecting samples. Therefore, this study proposes Focal-EIoU as the bounding box regression loss function to overcome the ambiguity of CIoU's width-height difference and improve the balance of detecting samples. From a gradient perspective, Focal-EIoU segregates high-quality anchor boxes from low-quality ones and emphasizes the former. The formula is presented in Equation (7), where γ is a parameter controlling the degree of outlier suppression. EIoU separates the loss term of the aspect ratio of CIoU into the difference between the predicted width and height and the minimum bounding box width and height, as shown in Equation (4) in the previous section.

$$L_{EIoU} = 1 - EIoU$$
$$L_{Focal-EIoU} = IoU^\gamma L_{EIoU} \tag{7}$$

YOLOv5 has three types of losses, including *box_loss* (localization loss), *obj_loss* (confidence loss), and *cls_loss*

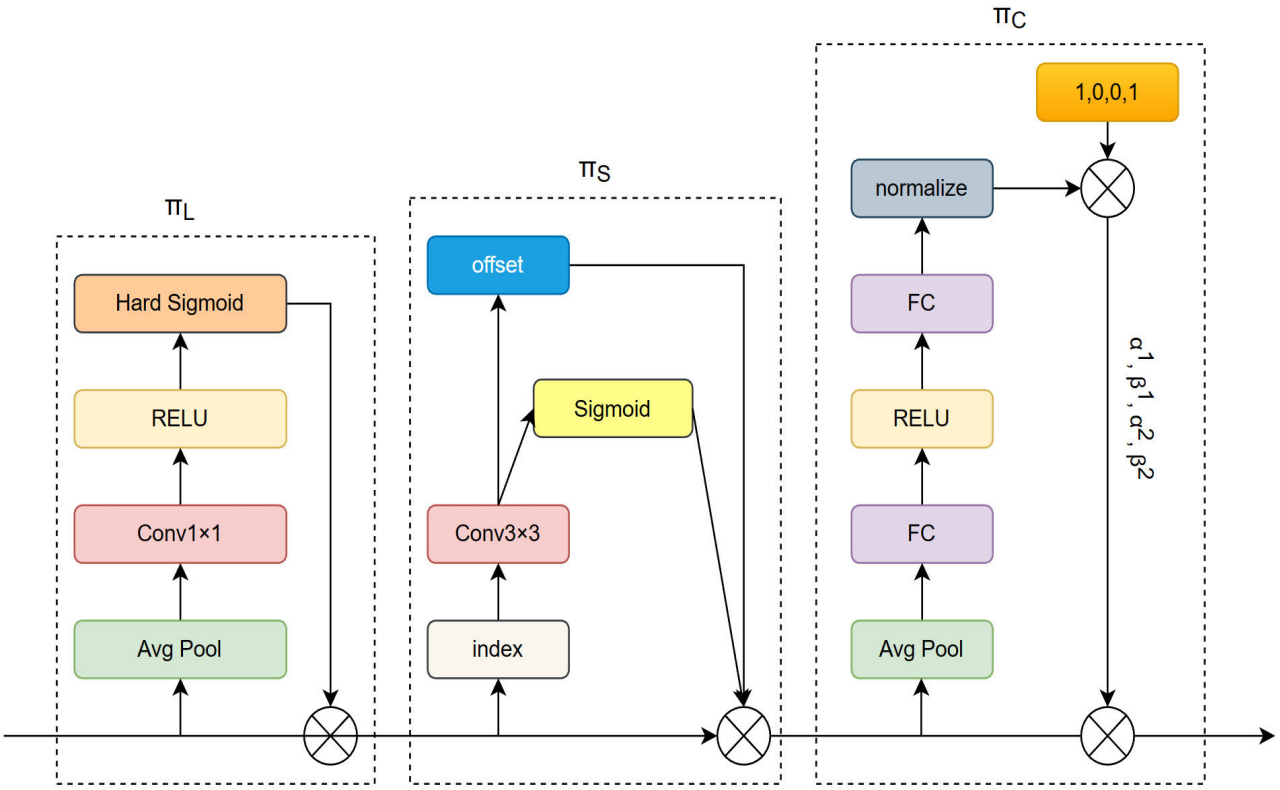

FIGURE 4. DyHead network structure.

(classification loss). To verify the performance of Focal-EIoU on this dataset, the results of CIoU and EIoU were compared, as shown in Figure 3. The total loss is calculated according to the following formula:

$$Loss = box\_loss + obj\_loss + cls\_loss$$

From Figure 3, it can be clearly seen that the loss of Focal-EIoU is significantly lower than that of CIoU and EIoU. Therefore, Focal-EIoU was selected as the loss function for this experiment.

E. DYNAMIC HEAD FRAMEWORK DyHead

Although traditional algorithms have attempted to improve head detection, they lack a unified perspective on the detection problem. In contrast, a recent approach known as DyHead [23] combines three self-attention mechanisms in the detection head, redefines of the four-dimensional tensor L × H × W × C as a three-dimensional tensor L × S × C. The approach employs scale-aware attention, spatial-aware attention, and task-aware attention in the L, S, and C dimensions, respectively.

1) Scale-aware attention module fuses features of different scales based on their semantic importance, and its expression is shown in Equation (8).

$$\pi_L(F) \cdot F = \sigma\left(f\left(\frac{1}{SC}\sum_{SC} F\right)\right) \cdot F \tag{8}$$

$f(\cdot)$ is a linear function that is approximated using a 1×1 convolution, $\sigma(x) = \max\left(0, \min\left(1, (x+1)/2\right)\right)$

2) Spatial-aware attention module first uses deformable convolution [24] to learn sparsity, and then aggregates cross-level features at the same spatial position. Its expression is shown in Equation (9).

K is the number of sparse sampling positions, $p_k + \Delta p_k$ is the shifted position where self-learned spatial offsets Δp_k focus on a distinctive region, and Δm_k is the self-learned important scalar at position p_k, all learned from the input features of the intermediate level F.

3) Task-aware attention module dynamically turns on or off feature channels to select different tasks, and its expression is shown in Equation (10).

$$\pi_s(F) \cdot F = \frac{1}{L} \cdot \sum_{l}^{L}\sum_{K=1}^{K} \omega_{l,k} \cdot F(l; p_k + \Delta p_k; c) \cdot \Delta m_k \tag{9}$$

$$\pi_c(F) \cdot F = \max\begin{pmatrix}\alpha^1(F) \cdot F_c + \beta^1(F), \\ \alpha^2(F) \cdot F_c + \beta^2(F)\end{pmatrix} \tag{10}$$

$[\alpha^1, \beta^1, \alpha^2, \beta^2]^T = \theta(\cdot)$ is a hyperfunction used to learn to control the activation threshold. $\theta(\cdot)$ first performs global pooling to reduce the dimensionality in the L × S dimension, then uses two fully connected layers, a normalization layer, and finally uses a shifted sigmoid function to normalize the output to [-1, 1]. The DyHead structure is shown in

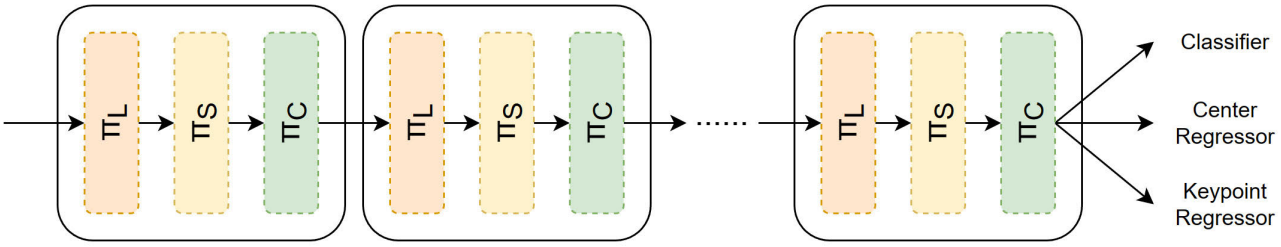

FIGURE 5. Connection scheme of DyHead blocks.

TABLE 3. Comparison of different dyhead blocks.

DyHead Block	layers	GFLOPs	mAP@0.5/%	mAP@0.5:0.95/%
0	214	16	40.3	24.8
1	248	16.4	41.4	25.8
2	272	16.7	42.5	26.0
3	296	17	41.3	25.9
4	320	17.2	41.3	26.0
5	344	17.5	41.9	26.2
6	368	17.8	41.7	26.4

Figure 4, where π_L, π_S, and π_C correspond to the scale-aware attention, spatial-aware attention, and task-aware attention modules, respectively.

To extract feature pyramids, a backbone network of any type can be employed, and these pyramids can be resized to a 3D tensor L × S × C with the same size. This tensor is then fed to the dynamic detection head, which comprises several DyHead blocks connected as illustrated in Figure 5. The output of DyHead can be utilized for various tasks, including classification and bounding box regression. Multiple DyHead blocks are arranged in the order of L, S, and C. Based on the number of DyHead blocks, this study compares the network depth, floating-point operations per second (GFLOPs), final mAP@0.5, and mAP@0.5:0.95, as presented in Table 3. After conducting an analysis, it is concluded that the optimal number of DyHead blocks is 2.

F. DATA AUGMENTATION METHODS

Mosaic data augmentation is a method of combining four images by random scaling, cropping, and arranging. It has the following advantages:

1) Enriching the dataset: Randomly using four images, randomly scaling, and then randomly arranging them greatly enriches the detection dataset, especially increasing many small targets, making the network more robust.

2) GPU Memory Consumption Reduction: The approach directly computes the data of four images, minimizing GPU memory consumption.

Due to the large amount of occlusion in the VisDrone dataset, Mixup data augmentation [24] is combined with Mosaic. The Mixup principle is to mix two random images proportionally to generate a new image, and the training process uses the new image for training. The formula for generating the new image is as follows:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)\, x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)\, y_j \end{aligned} \tag{11}$$

(x_i, y_i) and (x_j, y_j) are two randomly selected samples and their labels, λ is a randomly sampled number from the beta distribution, $\lambda \in \hat{\mathrm{E}}\,[0, 1]$, according to the literature [25], λ performs best when it is set to 0.5. Using Mixup can improve the robustness of the model.

III. EXPERIMENTAL ANALYSIS

For this experiment, we employed the VisDrone2019 dataset, which encompasses ten categories: pedestrian, people, bicycle, car, van, truck, tricycle, awning tricycle, bus, and motor. The dataset comprises 6,471 training images and 548 validation images. Figure 6 illustrates the dataset distribution, highlighting that small targets constitute the majority of the dataset. Moreover, Figure 7 showcases some of the training set images. All experiments were carried out under uniform environmental conditions and hyperparameters. The environment is shown in Table 4. The training epochs were set to 150, the batch size was set to 16, the initial learning rate was 0.01, the learning rate momentum was 0.937, and the weight decay coefficient was 0.0005. The initial three epochs are recognized as a warm-up phase during training. The SGD optimizer was used, and the input image size was 640 × 640. All training was conducted without pre-trained weights.

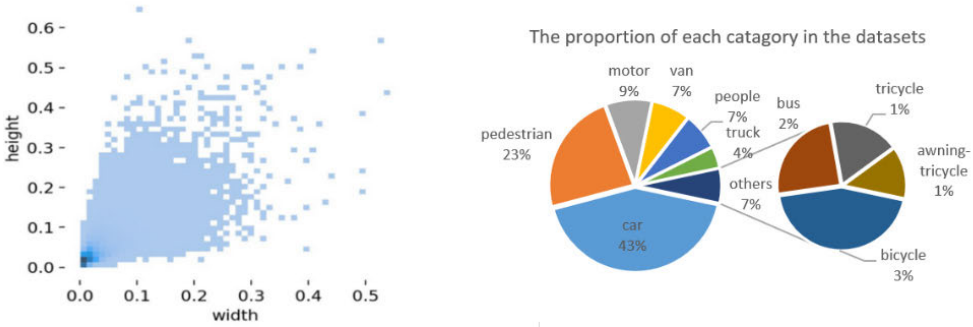

FIGURE 6. Distribution of the dataset. The left figure shows the overall data distribution map, and the right figure shows the proportion of each category.

FIGURE 7. Some images from the dataset.

TABLE 4. Experimental environment.

Use the system	Python	Pytorch	CUDA	GPU
Ubuntu20.04	3.8.10	1.10	11.3	RTX2080Ti

TABLE 5. Confusion matrix.

Reference	Prediction	
	positive example	counter-example
positive example	TP	FN
counter-example	FP	TN

A. EXPERIMENTAL RESULTS AND ANALYSIS

The experimental results are mainly evaluated based on mean average precision (mAP), precision, number of parameters, and model file size. Precision and recall are calculated using the confusion matrix shown in Table 5.

Precision mainly checks whether the prediction results are correct, and the formula is as follows:

$$P_{recision} = \frac{TP}{TP + FP} \tag{12}$$

Recall mainly checks whether the prediction results are comprehensive, and the formula is as follows:

$$R_{ecall} = \frac{TP}{TP + FN} \tag{13}$$

AP is the area enclosed by the precision-recall (PR) curve for a specific category in the training results, and its calculation formula is as follows:

$$AP = \int_0^1 p\,(r)dr \tag{14}$$

mAP is the average of all APs for different categories, and is an important indicator for evaluating the performance of object detection algorithms. The calculation formula is as follows: N is the number of categories in the dataset.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{15}$$

B. ABLATION EXPERIMENTS

According to Table 6, the comparison between groups A and B shows that, without increasing the number of parameters or model size, mAP@0.5 is improved by 2.90%. After using the K-means++ clustering method, the initial anchor box obtains better localization results.

Comparing group B and C in terms of mAP@0.5 and precision, it can be seen that mAP@0.5 is improved from 33.1% to 33.8%, an increase of 0.7 percentage points, and the model size remains the same. The use of Focal-EIoU reduces

TABLE 6. Ablation experiments on the visdrone dataset.

	method	mAP @0.5	mAP@0.5:0.95	Precision	Parameters	model file size	FPS
A	YOLOv5s	30.20%	15.70%	42.60%	7.0466M	13.74MB	45
B	YOLOv5s+K-means++	33.10%	18.00%	44.10%	7.0466M	13.74MB	45
C	YOLOv5s+Focal-EIoU+K-means++	33.80%	18.50%	44.90%	7.0466M	13.74MB	46
D	YOLOv5s+Focal-EIoU+K-means++ +EIoU_Soft_NMS	40.30%	24.80%	53.50%	7.0466M	13.74MB	21
E	YOLOv5s+Focal-EIoU+K-means++ +EIoU_Soft_NMS+DyHead	42.50%	26.00%	62.50%	7.3367M	14.31MB	20

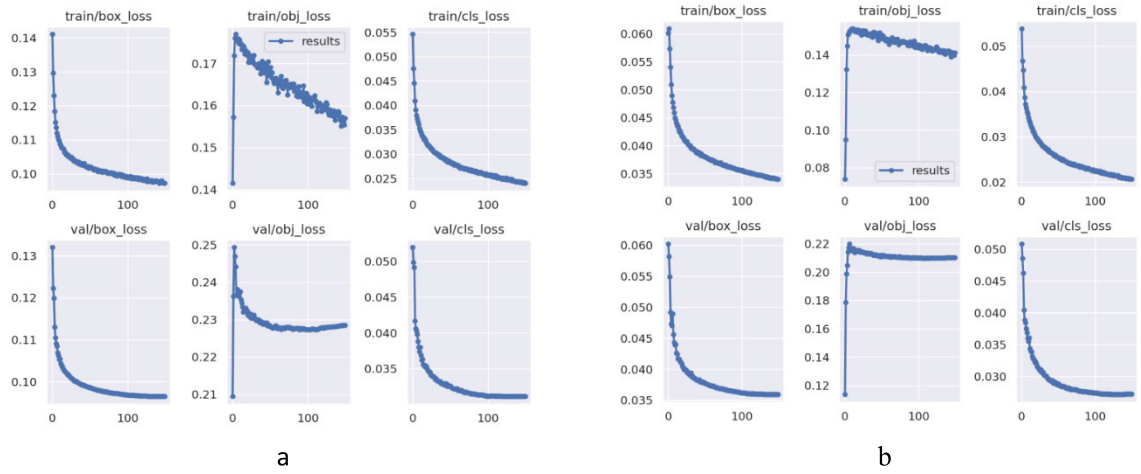

FIGURE 8. Loss comparison results. Where a is the CIoU result graph, and b is the Focal-EIoU result graph.

the overall loss, speeds up model convergence, and focuses more on high-quality anchor boxes, The loss contrast results are shown in Figure 8, where a is the original YOLOv5 result graph and b is the result graph using Focal-EIoU.

Comparing group C and D, there is a significant improvement after using EIoU_Soft_NMS, and mAP@0.5 and mAP@0.5:0.95 are improved to 40.3% and 24.8%, respectively, which are increased by 6.5% and 6.3%, respectively. The VisDrone dataset has a large amount of occlusion, and traditional NMS may miss detections after reaching the set threshold. EIoU_Soft_NMS replaces the original high score with a low score and then recalculates the score of the current detection box, which maximizes the retention of heavily occluded targets. The improved method has significant improvements for data with severe occlusion.

Comparing group D and E in terms of mAP and other aspects, using two DyHead blocks increases the number of parameters by 4% and the model size by 0.57MB. mAP@0.5 is improved by 2.2%, Adding a very small number of parameters brings about an improvement in accuracy.

The overall decrease in FPS is mainly due to the need for more calculations using Soft_NMS, including calculating the similarity between different bounding boxes, applying attenuation functions, and performing loop operations on each bounding box to update the score or confidence of the bounding boxes. These additional calculations will increase the computational load and inference time of the model, resulting in a decrease in FPS.

Regarding the mixed data augmentation method, the proportions of Mosaic and Mixup were adjusted, and the total sum was kept at 1. The results of experiments with different proportions are shown in Table 7, evaluated based on mAP and precision. The final choice of the proportion of Mosaic and Mixup is 0.5:0.5, which improves mAP@0.5 and mAP@0.5:0.95 by 0.3% and 1.9%, respectively, compared to not using Mixup.

In order to assess the effectiveness of the proposed improved method, we conducted a comparative analysis of the detection outcomes of YOLOv5s_2E and the original YOLOv5s model under suboptimal lighting conditions and

TABLE 7. Different proportions of data augmentation.

Mosaic : Mixup	mAP@0.5	mAP@0.5:0.95	Precision
1.0 : 0.0	42.5%	26.0%	62.5%
0.9 : 0.1	40.3%	25.2%	58.6%
0.7 : 0.3	40.6%	25.8%	61.0%
0.5 : 0.5	42.8%	27.9%	63.1%
0.3 : 0.7	41.7%	27.2%	63.3%
0.1 : 0.9	40.6%	25.2%	56.6%

FIGURE 9. Comparison diagram of the test results.

in the presence of occluded objects. Specifically, A1, B1, C1 shows the original image, while the detection results of the original YOLOv5s model and the improved detection results are depicted in images A2, B2, C2, and A3, B3, C3, respectively. The comparison results are detailed in Figure 9.

Group A depicts the detection outcomes under suboptimal lighting conditions. The original model exhibited a misclassification error, mistaking a pedestrian for a bicycle. Conversely, the proposed improved approach achieved an accurate detection of the pedestrian while simultaneously enhancing the confidence of other detections. The detection results under obstructed conditions are illustrated in Groups B and C. In Group B, the improved method successfully detected an obscured individual on the left, whereas the original model misidentified a backpack as a person. In Group C, the original model demonstrated an inability to recognize a pedestrian, as well as a heavily obstructed vehicle in the lower left corner. Furthermore, two vehicles were falsely detected. In contrast, the proposed enhanced

TABLE 8. Ablation experiments on the RSOD dataset.

method	aircraft	oiltank	overpass	playground	all
YOLOv5s	0.984	0.948	0.886	0.936	0.938
YOLO5s+k-means++	0.981	0.954	0.874	0.975	0.946
YOLO5s+Focal-EIoU+k-means++	0.983	0.955	0.905	0.963	0.952
YOLOv5s+Focal-EIoU+K-means++ +EIoU_Soft_NMS	0.978	**0.96**	0.904	**0.969**	0.953
YOLOv5s+Focal-EIoU+K-means++ +EIoU_Soft_NMS+DyHead	**0.985**	0.952	**0.936**	0.968	**0.960**

TABLE 9. Results of ablation experiments on the aquarium dataset.

method	fish	jellyfish	penguin	puffin	shark	starfish	stingray	all
YOLOv5s	0.668	0.865	0.562	**0.551**	0.594	0.563	0.619	0.632
YOLO5s+k-means++	0.676	**0.878**	0.551	0.481	0.656	0.596	0.612	0.636
YOLO5s+k-means++ +Focal_EIoU	0.688	0.864	0.511	0.414	0.574	0.601	0.555	0.601
YOLOv5s+CIoU+K-means++ +EIoU_Soft_NMS	0.703	0.835	0.529	0.531	**0.704**	0.731	0.668	0.672
YOLOv5s+CIoU+K-means++ +EIoU_Soft_NMS+DyHead	**0.707**	0.863	**0.688**	0.512	0.701	**0.742**	**0.741**	**0.705**

TABLE 10. Results of CIoU versus focal-EIoU in the aquarium dataset.

method	fish	jellyfish	penguin	puffin	shark	starfish	stingray	all
CIoU	0.668	0.865	0.562	0.551	0.594	0.563	0.619	0.632
Focal_EIoU	0.696	0.859	0.488	0.479	0.617	0.593	0.621	0.622

TABLE 11. Results of ablation experiments on the SSDD dataset.

method	Precision	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv5s	0.944	**0.957**	0.978	0.682
YOLOv5s+K-means++	0.970	0.932	0.977	0.687
YOLOv5s+EIoU_Soft_NMS+K-means++	**0.973**	0.929	0.977	0.702
YOLOv5s+EIoU_Soft_NMS+DyHead+K-means++	0.968	0.934	**0.978**	**0.717**

approach demonstrated a substantial improvement in detection performance.

C. PERFORMANCE ON OTHER DATASETS

The proposed improved method was tested on other datasets. The RSOD dataset is a remote sensing image dataset for object detection, which contains airplanes, oil tanks, playgrounds, and overpasses. The dataset consists of 753 training images and 183 validation images, with approximately 7000 annotations. The Aquarium dataset includes 7 categories of fish, jellyfish, penguins, puffins, sharks, starfish, and stingrays. The training set consists of 448 images and the validation set consists of 127 images. The SSDD dataset contains only the Ship class, with 928 training sets and 232 validation sets. The training epoch is 150, and the batch size is 16. The same environment as the VisDrone dataset was used for training.

Table 8 shows the ablation experiments on the RSOD dataset, mainly evaluating the mAP@0.5 of each category on the validation set. With one DyHead block, the improved

TABLE 12. Comparative experiments of different algorithms.

model	mAP@0.5	mAP@0.5:0.95
RSOD	33.00%	——
Mixed YOLOv3-LITE	28.50%	——
YOLOv5s-M5	37.66%	——
SSD	23.70%	——
Faster-RCNN	34.10%	18.60%
YOLOv5s+CARAFE	32.40%	17.10%
YOLOv7-tiny	32.80%	16.70%
YOLOv5m	34.80%	19.20%
YOLOv5l	37.30%	20.80%
GBS-YOLOv5	35.30%	20.00%
DMS-YOLOv5	39.70%	——
DC-YOLOv8[28]	41.50%	24.70%
KPE-YOLOv5s[29]	39.20%	——
Dy-YOLOv5[30]	40.60%	24.70%
UN-YOLOv5[31]	40.50%	22.50%
YOLOv5s+P2+MBF+FE+MSimA[32]	38.40%	22.20%
RetinaNet[33]	21.2%	——
Ours	43.00%	26.80%

method achieved a mAP of 96.0% on the RSOD dataset, and YOLOv5s_2E improved the mAP@0.5 by 2.2% compared to the original YOLOv5s model.

The results of ablation experiments on the Aquarium dataset are shown in Table 9. The mAP@0.5 for each category was compared using the validation set, and the conditions were consistent with the RSOD dataset. From the ablation experiments in Table 9, it can be seen that CIoU outperforms Focal-EIoU on the Aquarium dataset, the results of the two loss functions are shown in Table 10. Therefore, the CIoU loss function and a DyHead block of 2 were selected for this dataset. The final results show that our method improves the mAP@0.5 on the Aquarium dataset by 7.3%, further demonstrating the effectiveness of our approach.

Table 11 illustrates the results of the ablation experiments conducted on the SSDD dataset. Focal-EIoU was not utilized due to the lower overlap in the center point of the two boxes in the SSDD dataset, a DyHead block of 3 was selected. The model algorithm presented in this paper maintained the mAP @ 0.5 score while improving the mAP @ 0.5:0.95 score by 3.5%, further supporting the effectiveness of the proposed algorithm.

D. COMPARATIVE EXPERIMENT ANALYSIS

To demonstrate the advantages of our proposed algorithm, we compared it with other object detection algorithms under the same training parameters. The main evaluation metrics were the mAP@0.5 and mAP@0.5:0.95. As shown in Table 12, the improved YOLOv5s outperformed the same series of m and l models, with a lower number of parameters than the YOLOv5m and YOLOv5l models, and an increase of 8.2% and 5.7% in mAP@0.5, and 7.6% and 6.0% in mAP@0.5:0.95, respectively. Compared with YOLOv7-tiny, mAP@0.5 and mAP@0.5:0.95 increased by 10.2% and 10.1%, respectively. Compared with the latest algorithms GBS-YOLOv5 [26], DMS-YOLOv5 [27], etc., mAP @ 0.5 is also improved to different degrees, which proves the performance of this method on small target detection.

IV. CONCLUSION

Detection of small targets, particularly when they are heavily occluded, poses significant challenges, often leading to missed detections and false alarms. In this paper, we presented an enhanced small target detection algorithm founded on YOLOv5s. Empirical findings revealed that the proposed model achieved remarkable improvements in all evaluation metrics with only a minor increase in the number of parameters, satisfying the accuracy requirements for small target detection. Nonetheless, there is still ample room for improvement in our model, In future studies, we aspire to tackle the challenges of streamlining the inference process in the Detect module and minimizing the model's parameters to enhance the efficacy of the proposed approach.

REFERENCES

[1] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[3] 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.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[5] H. Abbas, M. E. O'Kelly, A. Rodionova, and R. Mangharam, "A driver's license test for driverless vehicles," *Mech. Eng.*, vol. 139, no. 12, pp. S13–S16, Dec. 2017.

[6] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, Jul. 2021, Art. no. 100057.

[7] L. Xu, G. Tetteh, J. Lipkova, Y. Zhao, H. Li, P. Christ, M. Piraud, A. Buck, K. Shi, and B. H. Menze, "Automated whole-body bone lesion detection for multiple myeloma on 68Ga-pentixafor PET/CT imaging using deep learning methods," *Contrast Media Mol. Imag.*, vol. 2018, pp. 1–11, 2018.

[8] M. Krišto, M. Ivasic-Kos, and M. Pobar, "Thermal object detection in difficult weather conditions using YOLO," *IEEE Access*, vol. 8, pp. 125459–125476, 2020.

[9] S. Ren, K. He, and R. Girshick, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 1–9.

[10] 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.

[11] W. Liu, D. Anguelov, and D. Erhan, "SSD: Single shot multibox detector," in *Computer Vision—ECCV*. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.

[12] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[13] J. Wang, K. Chen, R. Xu, Z. Liu, C. C. Loy, and D. Lin, "CARAFE: Content-aware reassembly of features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3007–3016.
[14] W. Sun, L. Dai, X. Zhang, P. Chang, and X. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Appl. Intell.*, vol. 52, no. 8, pp. 8448–8463, 2021.
[15] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[16] H. Zhao, Y. Zhou, L. Zhang, Y. Peng, X. Hu, H. Peng, and X. Cai, "Mixed YOLOv3-LITE: A lightweight real-time object detection method," *Sensors*, vol. 20, no. 7, p. 1861, Mar. 2020.
[17] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA,USA, Dec. 2018, pp. 2503–2510.
[18] W. Zhan, C. Sun, M. Wang, J. She, Y. Zhang, Z. Zhang, and Y. Sun, "An improved YOLOv5 real-time detection method for small objects captured by UAV," *Soft Comput.*, vol. 26, no. 1, pp. 361–373, Jan. 2022.
[19] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5562–5570.
[20] Y.-F. Zhang, W. Ren, Z. Zhang, Z. Jia, L. Wang, and T. Tan, "Focal and efficient IOU loss for accurate bounding box regression," *Neurocomputing*, vol. 506, pp. 146–157, Sep. 2022.
[21] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. 24th ACM Int. Conf. Multimedia*, Oct. 2016, pp. 516–520.
[22] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 7, pp. 12993–13000.
[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] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 764–773.
[25] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.
[26] H. Liu, X. Duan, H. Lou, J. Gu, H. Chen, and L. Bi, "Improved GBS-YOLOv5 algorithm based on YOLOv5 applied to UAV intelligent traffic," *Sci. Rep.*, vol. 13, no. 1, p. 9577, Jun. 2023.
[27] T. Gao, M. Wushouer, and G. Tuerhong, "DMS-YOLOv5: A decoupled multi-scale YOLOv5 method for small object detection," *Appl. Sci.*, vol. 13, no. 10, p. 6124, May 2023.
[28] H. Lou, X. Duan, J. Guo, H. Liu, J. Gu, L. Bi, and H. Chen, "DC-YOLOv8: Small-size object detection algorithm based on camera sensor," *Electronics*, vol. 12, no. 10, p. 2323, May 2023.
[29] R. Yang, W. Li, X. Shang, D. Zhu, and X. Man, "KPE-YOLOv5: An improved small target detection algorithm based on YOLOv5," *Electronics*, vol. 12, no. 4, p. 817, Feb. 2023.
[30] Q. Wei, X. Hu, and Q. Hou, "Dynamic-YOLOv5: An improved aerial small object detector based on YOLOv5," in *Proc. 3rd Int. Conf. Neural Netw., Inf. Commun. Eng. (NNICE)*, Feb. 2023, pp. 679–683.
[31] J. Guo, X. Liu, L. Bi, H. Liu, and H. Lou, "UN-YOLOv5s: A UAV-based aerial photography detection algorithm," *Sensors*, vol. 23, no. 13, p. 5907, Jun. 2023.
[32] J. Shang, J. Wang, S. Liu, C. Wang, and B. Zheng, "Small target detection algorithm for UAV aerial photography based on improved YOLOv5s," *Electronics*, vol. 12, no. 11, p. 2434, May 2023.
[33] W. Yu, T. Yang, and C. Chen, "Towards resolving the challenge of long-tail distribution in UAV images for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3257–3266.

TAO SHI received the Ph.D. degree in control science and engineering from the Beijing University of Science and Technology, Beijing, China, in 2015. He is currently an Associate Professor with the School of Electrical Engineering and Automation, Tianjin University of Technology. His research interests include brain-like intelligent robots, robot vision, and biologically inspired intelligent computing.

YAO DING received the B.S. degree in engineering from the Tianjin University of Technology, Tianjin, China, in 2022, where he is currently pursuing the master's degree in electrical engineering and automation. His current research interest includes image processing.

WENXU ZHU received the B.S. degree from the Taizhou University of Science and Technology, Taizhou, Jiangsu, China, in 2021. He is currently pursuing the master's degree in engineering with the North China University of Science and Technology. His research interests include computer vision, target detection, and deep learning.