An Improved YOLOv5 Algorithm for Detecting Target

Chao Chen, Bin Wu, Yongguo Shi

Abstract—**In order to have shorter detection time and higher detection accuracy for target detection, this article mainly improves the YOLOv5 algorithm from optimizing active function and the loss function. The results of comparative experiments shown that our improved YOLOv5 algorithm's performance for detecting object was greatly improved.**

Index Terms—**target detection, YOLOv5, active function, losses function.**

I. Introduction

IDENTIFYING the targets is applied by using automated equipment[1], [2], [3], including Faster R-CNN based insect control[4], but Faster R-CNN had slow speed and too much storage space. For example, scale invariant feature approach was used to identify the insects, but it requires too much storage space[5]. However, such as Faster R-CNN relies on the pre-defined anchor box, and there is still a lot of calculation work, but they can not achieve the effect of real-time monitoring[6]. This two-stage algorithm cannot meet the requirements of real-time detection. Single-stage detection architectures'disadvantage is imprecise. The anchor-based detection architecture detects the boundary of the target as multiple anchor boxes, and predicts the offset and category of each target. YOLOv1 and YOLOv2 were used for faster target detection[7], [8], [9], the speed of detection has increased, but their accuracy has decreased. YOLOv3 (You Only Look Once v3) is typical faster detection methods with good detection performance[7], [10]. To abtain better detection effect, the author designed darknet-53 architecture. Compared with ResNet-152 and ResNet-101, darknet-53 not only has good accuracy, but also has faster detection speed and fewer layers[7]. Adarsh.et.al. put forward YOLOv3-Tiny based object detection to reduce the storage capacity and can be well applied to embeded devices[11]. Improved YOLOv3 with DenseNet perfect detect some objects with small pixel proportion in remote sensing images[12], [13], [14]. Some real-time object detection methods were proposed to apply on the mobile devices[15], [16], [17]. The object detector was proposed to solve the problem with some training techniques and optimized hyperparameters which did not contribute much to the deep network itself[18]. YOLOv4 combined a lot of previous research techniques and makes appropriate innovations. CSPNet(Cross Stage Partial Networks) was used for object detection. And,its number of parameters and FLOPS of the model were reduced[19]. YOLOv5 modified some structures inside the network, replaced the loss function, and achieved a good detection effect[20]. YOLOv6 and YOLOv7 had no fundamental improvements involving a stack of tricks[21], [22]. In 2023, the latest YOLOv8 was a state-of-the-art (SOTA) model for no anchor. A key feature of YOLOv8 is its scalability. It was designed as a framework that supports all previous versions of YOLO, to easily switch between versions and compare their performance. In addition to scalability, YOLOv8 includes many other innovations that make it widely used in object detection and image segmentation tasks. The key point-based detection architecture detects the target bounding box as multiple key points to replace the anchor frame[23], [24], [25]. Algorithms such as CornerNet, centerNet, and Fcos are all based on the very classic architecture in key point detection [26], [27], [28]. However, these detection accuracy are not high all the time, which is prone to false detection. YOLOv6 introduced SIoU loss function introducing the vector angle and the distance loss to accelerat network convergence[21], and further improved the accuracy of regression[22], [29]. The degree of freedom of regression was effectively reduced, network convergence was accelerated, and the accuracy of regression was further improved, for example: GIoU[29], Distance-IoU Loss[30], EIoU[31], αIoU[32], [33], WIoU[34]. Existing methods based on YOLOv5 that doesn't consider the mismatching direction between real and predicted frames obtained a slower and less efficient model.

This article mainly improved the YOLOv5 algorithm from the active function and the losses function. Comparative experiments results shown our method improved the detection's performance. Our main contributions are as follow:

1) New active function is helpful for effective feedback of gradient, so as to speed up convergence and can extract more distinguishing features.
2) New losses function can surpass existing IoU_based losses and the accuracy of regression is further improved. Especially, the detection effect is obvious for some crowded or occluded small targets.
3) Five experimental related evaluation indicators were proposed to compare the experimental results.
4) The comparative experiments on VOC data set and COCO data set were carried out.

II. Related works

A. The network of YOLOv5

YOLOv5 performs bounding box coordinate estimation and class prediction at the same time. The network convo-

Manuscript received January 18, 2024; revised August 2, 2024. This work was supported in part by Object Detection Technology Based on Deep Learning (No.202307033), NSF of Sichuan Province (2023NSFSC0065) and Bridge Nondestructive testing and Engineering calculation Key Laboratory project of Sichuan University (No.2023QYY04).

Chao Chen is a PhD candidate of Southwest University of Science and Technology, Mianyang, Qinglong Avenue 59, P. R. China. (e-mail: ch10503@ 126.com).

Bin Wu is a professor of Southwest University of Science and Technology, Mianyang, Qinglong Avenue 59, P. R. China. (phone: 139- 0901-8585; e-mail: wubin@swust.edu.cn).

Yongguo Shi is a professor of Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal University, Neijiang, Hongqiao Street 1, P. R. China. (e-mail: scumat@163.com).

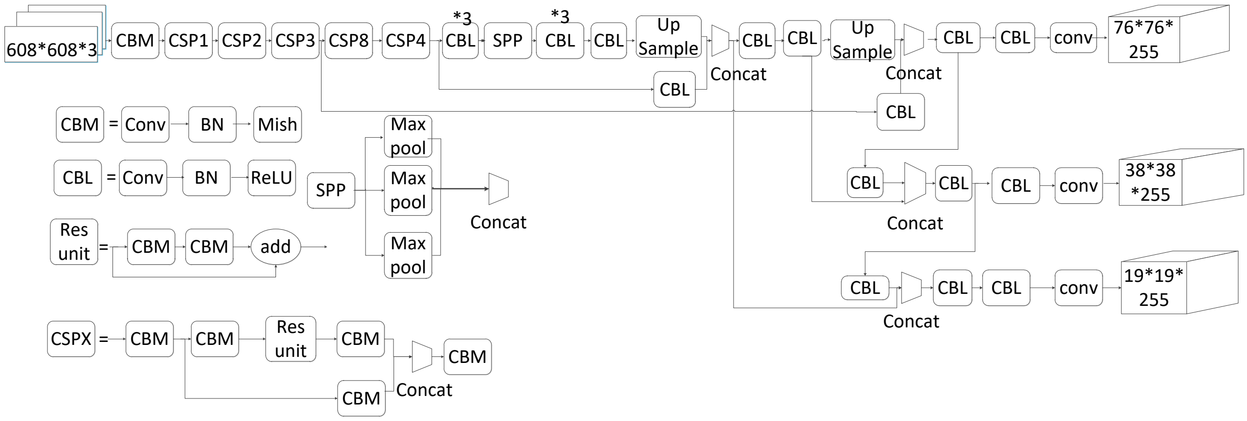

Fig. 1. The network of YOLOv5 (redrawing based on[17], [18], [19], [20], [21])

lutional kernels are 1×1 or 3×3. YOLOv3 extract the deeper features by Darknet53 network that added 5 residual modules involving one or multiple residual units with the Conv+BN+Leaky_ReLU. Backbone of YOLOv5 consists of focus structure, new CSP-Darknet53 and the improved neck(New CSP-PAN structure of FPN+PAN). We redrawn the structure of YOLOv5 based on YOLO's idea[17], [18], [19], [20], [21]. The structure of YOLOv5 was shown in Figure 1[17], [18], [19], [20], [21].

B. YOLOv5's location prediction

The final frame coordinate value is b_x, b_y, b_w, b_h. That is, the position and size of the bounding box which is relative to the feature map are the predicted output coordinates we need[14]. But the actual learning goal of our network is t_x, t_y, t_w, t_h and other four offsets, where t_x and t_y are predicted coordinate offset values, t_w and t_h are scale scaling. With these 4 offsets, it is natural to find the 4 coordinates (b_x, b_y, b_w, b_h) that are really needed according to the previous formula. YOLOv5 Directly predict the center point, and use the Sigmoid function to limit the offset between 0 and 1. The specific formula are shown as follows[7].

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = p_w e^{t_w} \tag{3}$$

$$b_h = p_h e^{t_h} \tag{4}$$

The network predicts 4 coordinates for each bounding box (t_x, t_y, t_w, t_h)[3], [21]. We redrawn four coordinates for each bounding box of YOLOv5 based on YOLO's idea[18]. It was shown in Figure 2.

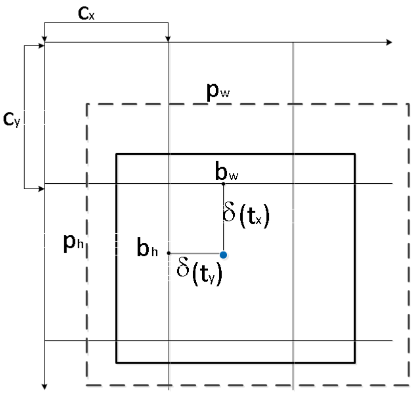

Fig. 2. four coordinates for each bounding box of YOLOv5(redrawing based on[18])

C. Loss fucntion of YOLOv5

YOLOv5 ends the iterative procedure with the smallest total error, which is divided into three types. The specific formula are shown as follows[4], [20], [17].

$$\begin{aligned}
&\text{Loss} = \\
&\lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\left(x_i - \hat{x}_i^j \right)^2 + \left(y_i - \hat{y}_i^j \right)^2 \right] + \\
&\lambda_{\text{coord}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\left(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j} \right)^2 + \left(\sqrt{h_i^j} - \sqrt{\hat{h}_i^j} \right)^2 \right] - \\
&\sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[\left(\hat{C}_i^j \log \left(C_i^j \right) + \left(1 - \hat{C}_i^j \right) \log \left(1 - \hat{C}_i^j \right) \right] - \\
&\lambda_{\text{noobj}} \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} \left[\left(\hat{C}_i^j \log \left(C_i^j \right) + \left(1 - \hat{C}_i^j \right) \log \left(1 - \hat{C}_i^j \right) \right] - \\
&\sum_{i=0}^{s^2} I_{ij}^{obj} \sum_{j \in class}^{B} \left[\left(\hat{P}_i^j \log \left(P_i^j \right) + \left(1 - \hat{P}_i^j \right) \log \left(1 - \hat{P}_i^j \right) \right]
\end{aligned} \tag{5}$$

Loss function is the degree of difference between the predicted value and the real value, which determines the performance of the model to a great extent. YOLOv5 has three loss functions: The cls_loss calculates whether the anchor and the corresponding calibration classification are correct.The box_loss calculates error between prediction anchor and calibration anchor (GIoU)[21].The obj_loss calculates the confidence of belonging to the class. YOLOv5's box_loss is different from YOLOv3's. Total loss function = classification loss + location loss + confidence loss. Classification losses and location losses were calculated using the binary cross entropy loss function BCE with Logits loss. The IoU function introduced above is used to calculate the confidence loss. IoU only solves the overlapping of two objects.

III. Improve related work

A. Improved active fuction

Common activation functions in deep networks are prone to problems such as gradient disappearance or slow convergence. The graphs of corresponding functions are shown in Figure 3.

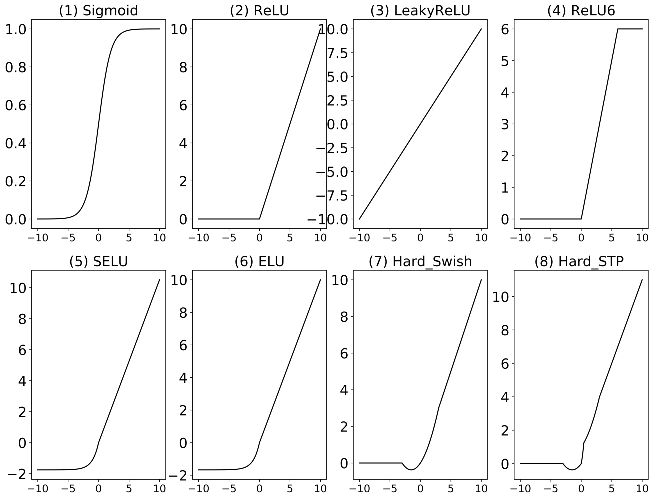

Fig. 3. Common activation functions

And derivations functions of these common activation functions are shown in Figure 4.

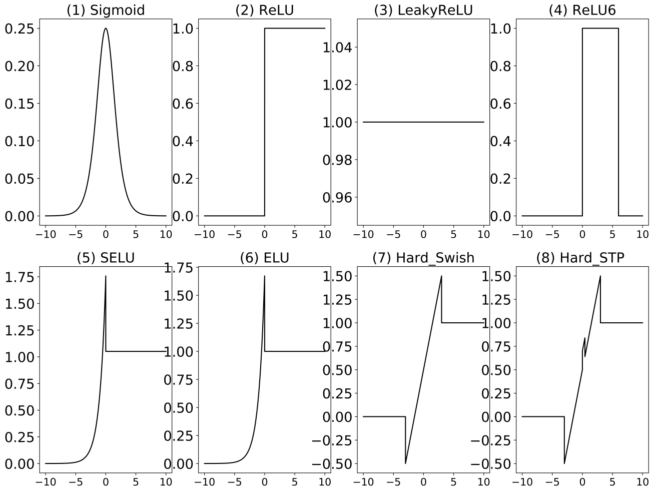

Fig. 4. Derivation of common activation functions

YOLOv5 uses Sigmoid activation function, and the average output value of Sigmoid function is 0.5, so that the input of neurons in the next layer is distributed on both sides with 0.5 as the center. The most popular ReLU series activation function keep the gradient and accelerate weight update of loss function. Hard_Swish activation function is shown in equation (6).

$$Hard\_Swish(x) = x\frac{ReLU6(x+3)}{6} \tag{6}$$

However, the Hard_Swish activation function is non-zero-mean, so this paper combines the triple polyomial activation function which does not disappear the gradient and the output is zero-mean. A new activation function named Hard_STP function was designed and shown in equation (7).

$$\begin{aligned} Hard\_STP(x) =& x\frac{ReLU6(x+3)}{6} \\ &+ \left(ax^3+bx^2+cx\right) \end{aligned} \tag{7}$$

where a, b and c are hyperparameters. Compared with these classical activation function, Hard_STP activation function solved the problem of gradient disappearance near 0 and is shown in Figure 4. The real values for hyperparameters a, b, and c are set $a, b, c \in [1, 2]$. By repeated contrast experiments, it was found that $a = 1.05, b = 1.5$ and $c = 1.6$ were the best results.

B. Improved losses function

In recent years, the commonly boundary box regression loss has been proposed, including IoU, GIoU, CIoU, DIoU, SIoU, WIoU and other loss functions [31], [32], [33], [34], [35].

IOU_Loss is overlap area between the detection frame and the target frame. GIOU_Loss solves the problem when the boundary frame does not coincide. DIOU_Loss is the distance between center points of boundary frame is considered. CIOU_Loss is scale information which considers the aspect ratio of boundary frame based on DIOU. CIOU_Loss regression mode was adopted in YOLOv4. These loss functions is the gap between the prediction frame and the target frame by considering the overlap degree, center point distance, aspect ratio and other factors, so as to guide the network to minimize the loss and improve the regression accuracy. It can make the prediction box quickly drift to the nearest axis, and then only need to return one coordinate (X or Y), which effectively reduces the total number of degrees of freedom[35].

In order to solve the overlapping of two objects, GIoU is introduced to maintain the invariance of the size of IoU, and strong correlation with IoU can be maintained when overlapping. The specific formula of GIoU is as follows.

$$\mathrm{G_{IoU}} = \mathrm{L_{IoU}} - \frac{A^c - u}{A^c} \tag{8}$$

where, $-1 \le \mathrm{G_{IoU}} \le 1$.

$$L_{\mathrm{IoU}} = 1 - IoU = 1 - \frac{I}{U} \tag{9}$$

I is intersection of two objects, U is union of two objects.

WIoU(Weighted Intersection over Union) weighted the IoU by considering the area between the prediction box and the real box, which solves the bias that may occur in traditional IoU evaluation results. WIoU can evaluate the detection results more accurately. The deviation problem of traditional IoU is avoided [34]. The specific formulas are shown as follows[34].

$$\mathrm{L_{WIOU}} = R_{\mathrm{WIoU}} L_{\mathrm{IoU}}, \tag{10}$$

$$R_{\mathrm{WIoU}} = \exp\left(\frac{\left(x-x_{gt}\right)^2 + \left(y-y_{gt}\right)^2}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{11}$$

Where:α is between c_w(level edges) and δ(oblique edges). β is between c_h(vertical edges) and δ(oblique edges).

SIoU further consideres the vector angle between the real box and the prediction box, and redefined the correlation loss function, which includes four parts. We redrawn the angle cost of WIoU[34]. Where, the angle cost is shown in Figure 5 and their formula involved are shown in equation (12) - (16).

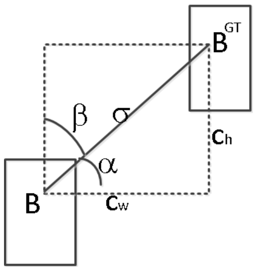

Fig. 5. Angle cost(redrawing based on[34])

$$r = \frac{\beta}{\delta\alpha^{\beta-\delta}}. \tag{12}$$

$$\begin{aligned}\Lambda =&1-2^{*}\sin^{2}\left(\arcsin\left(\frac{\mathrm{c_h}}{\sigma}\right)-\frac{\pi}{4}\right)\\ =&\cos\left(2*\left(\arcsin\left(\frac{\mathrm{c_h}}{\sigma}\right)-\frac{\pi}{4}\right)\right)\end{aligned} \tag{13}$$

$$\frac{\mathrm{c_h}}{\sigma} = \sin(\alpha) \tag{14}$$

$$\sigma = \sqrt{\left(b_{c_x}^{gt}-b_{c_x}\right)^2+\left(b_{c_y}^{gt}-b_{c_y}\right)^2} \tag{15}$$

$$c_h = \max\left(b_{c_y}^{gt}, b_{c_y}\right) - \min\left(b_{c_y}^{gt}, b_{c_y}\right) \tag{16}$$

Where: $\sin(\sigma)$ is opposite over hypotenuse in a right triangle. σ is the distance between the center point of the real box and the predicted box. c_h is the height difference between the center point of the real box and the predicted box, $b_{c_x}^{gt}$ and $b_{c_y}^{gt}$ are the center coordinate of the real box; b_{c_x} and b_{c_y} are the center coordinate of the prediction box.

The distance cost's formula involved are shown in equation (17) - (20).

$$\Delta = \sum_{t=x,y}\left(1-e^{-\gamma\rho_t}\right) = 2-e^{-\gamma\rho_x}-e^{-\gamma\rho_y} \tag{17}$$

$$\rho_x = \left(\frac{b_{c_x}^{gt}-b_{c_x}}{c_w}\right)^2 \tag{18}$$

$$\rho_y = \left(\frac{b_{c_y}^{gt}-b_{c_y}}{c_w}\right)^2 \tag{19}$$

$$\gamma = 2-\Lambda \tag{20}$$

where: c_w and c_h are the width and height of the minimum enclosing rectangle of the real box and the prediction box.

The shape cost's formula involved are shown in equation (21) - (22).

$$\Omega = \sum_{t=w.h}\left(1-e^{-w_t}\right)^{\theta} = \left(1-e^{-w_w}\right)^{\theta}+\left(1-e^{-w_h}\right)^{\theta} \tag{21}$$

$$w_W = \frac{w-w^{gt}}{\max\left(w,w^{gt}\right)}, w_h = \frac{h-h^{gt}}{\max\left(h,h^{gt}\right)}. \tag{22}$$

Where:w, h, w^{gt}, h^{gt} are the width and height of the prediction box and the real box respectively, so as to control the attention to shape loss. In order to avoid the excessive attention to shape loss and reduce the movement of the prediction box, the author uses the genetic algorithm to set the parameter θ as 4[2], [6].

The IoU cost remains. Then, the final SIoU loss function was defined as Formula (23):

$$\mathrm{Loss}_{SIoU} = 1-\mathrm{L}_{\mathrm{IoU}}+\frac{\Delta+\Omega}{2} \tag{23}$$

Inspired by SIoU[33] and WIoU[34], [35], an improved loss function SWIoU is proposed here to optimize the target detection model. SWIoU loss function redefines the distance loss by introducing the vector angle between required regressions, effectively reduces the degree of freedom of regression, speeds up the network convergence, and further improves the regression accuracy. SWIoU loss function is defined as formula (24) and (25) :

$$\mathrm{Loss}_{SWIOU} = \left(1-\mathrm{L}_{\mathrm{IoU}}+\frac{\Delta+\Omega}{2}\right)r \tag{24}$$

$$\mathrm{Loss}_{SWIOU} = \left(1-\mathrm{L}_{\mathrm{IoU}}+\frac{\Delta+\Omega}{2}\right)\frac{\beta}{\delta\alpha^{\beta-\delta}} \tag{25}$$

IV. Experiment

A. Experimental hardware information

GPU resource information: GPU v100 32GB, RAM 128 GB.

B. Environment configuration

Python = 3.8.8, pytorch=1.13.1, torchvision=0.14.1, torchaudio=0.13.1, pytorch-cuda=11.7

C. Hyperparameters

For fairness of comparison, the hyperparameters which trained and evaluated on the two publicly available datasets were exactly the same. momentum=0.937, lr0 = 0.01, lrf = 0.01, warmup_momentum = 0.8, warmup_epochs = 3.0, warmup_bias_lr = 0.1, box = 0.05, cls = 0.5, cls_pw = 1.0, obj = 1.0, obj_pw = 1.0, iou_t = 0.2, anchor_t = 4.0, fl_gamma = 0.0, hsv_h = 0.015, hsv_s = 0.7, hsv_v = 0.4, degrees = 0.0, translate = 0.1, scale = 0.5, shear = 0.0, perspective = 0.0, flipud = 0.0, fliplr = 0.5, mosaic = 1.0, mixup = 0.0, copy_paste = 0.0.

D. Related evaluation indicators

To verify the effectiveness of the improved algorithm, we used 10 algorithms to do two sets of comparative experiments.The mAP(The average detection accuracy), Loss(the final loss function value), FLOPs, evaluate time and the memory of the optimal neural network's weight are five basic experimental indicators[1], [5], [9].

1) The higher the mAP50, the better. Average precision is the evaluation index of the mainstream target detection model. Generally speaking, the better the classifier, the higher the AP value. The size of mAP must be in the range of [0,1]; The higher, the better. This indicator is the most important one in the target detection algorithm. AP50 means that the value of the IoU is 50%, and mAP50_95 means that the value of the IoU

is taken from 50% to 95%, and then the Average precision under these IoU is calculated.

2) Loss is the value of the loss function which is divided into three types(box_loss, obj_loss and cls_los); The lower, the better; YOLOv5 ends the iterative procedure with the smallest total error.
3) The FLOPs is the floating-point operations per second.
4) Evaluate time is the evaluation indicators of the detection's speed.
5) The Memory of the optimal neural network's weight can limit the application scenarios of the model.

E. Experimental results

1) The VOC experiments: For the dataset, we select VOC 2012 dataset, and select 5717 images as the training data and 5823 images as the validation data. we choose YOLOv5 model as baseline, and these models are trained for 100 epochs, where: batch size is 64. The number of network layers are 157 layers, the number of network's parameter is 7073569; The number of anchors per target is 4.45, Image size for training and evaluation sizes are 640 * 640. The specific results of this experiment are shown in Table I.

In order to verify whether the proposed method can effectively improve the detection accuracy, the ablation experiment was carried out on VOC experiments.

As can be seen from Table 1. After adding SWIOU loss function and Hard_STP(x) activation function, the mAP50 of the method on the VOC dataset was 72.876 %, which compared with other mainstream methods, the average mAP50 was improved by 0.6 percentage points,and mAP50 and mAP50_95 was improved by 1 percentage points than that of other mainstream methods. The data comparison of the mAP50 on VOC dataset are shown in Figure 6.

The related loss value also had a certain downward trend, and FLOPs, training and evaluate time and the memory of the optimal neural network's weight did not increase significantly. The method presented in this paper can more accurately detect the category and true boundary of the target. Therefore, it was proved that the proposed algorithm has certain superiority.

In addition, we analyzed the detection effect of some crowded and occluded small targets, the proposed method can detect the class and true boundary of these small targets more accurately. The mAP50 of some crowded and occluded small targets of the VOC experiments had been improved while the FLOPs had to be reduced to some extent. The comparative experiments on VOC data sets shown that the optimized detection algorithm had a good performance in detection accuracy, and did not increase the more extra calculation amount. The detection effect of some crowded and occluded small targets(For example: bird, bottle, cow, pottedplant, Sheep and boat.) were particularly obvious, and the detailed data of small target detection in some crowded or occluded states are shown in Table II.

The experimental results shown that some part crowded and occluded small targets(bird, bottle, cow, pottedplant, Sheep, boat) of the VOC were particularly obvious. The part detection effect of VOC data are shown in Table II.. Therefore, the detection effect based on the proposed algorithm on small targets was obvious. This method has good practicability and potential for further study.

2) The COCO experiments: In order to verify the effectiveness of the proposed algorithm, we conducted experiments on a public generic COCO dataset. For the dataset, we selected MS-COCO2017 dataset, and selected 117266 images as the training data and 4952 images as the validation data. Since this version was officially recognized, we choose YOLOv5 model, and these models were trained for 100 epochs, where: batch size is 64. The number of network layers are 214 layers, the number of network's parameter is 7235389 parameters. Image size for training and evaluation sizes are 640 * 640. The specific results of The COCO experiments are shown in Table III. As can be seen from Table 3, after adding SWIOU loss function and HardSTP(x) activation function, mAP50 and mAP5095 were 72.8% and 60.9% respectively on COCO dataset, which is 0.8-1.7 percentage points higher than mAP of other mainstream methods. the related loss value also has a certain downward trend. And FLOPs, trainingtime and the memory of the optimal neural network's weight did not increase significantly. The data comparison of the mAP50 on COCO dataset are shown in Figure 7.

The comparative experiments on COCO data sets shown that the optimized detection algorithm had a good performance in detection accuracy, and did not increase the extra calculation amount.

In addition, we analyzed the detection effect of some crowded and occluded small targets on COCO dataset. The detailed data of small target detection in some crowded or occluded states are shown in Table IV. The detection effect of some crowded and occluded small targets(person motorcycle trafc light bird cat pizza person Motor-cycle.) was particularly obvious and detailed data were shown in Table IV. On COCO data set, most of the small targets were detected by the algorithm in this paper. Because the proposed algorithm suppressed the background and removed most of the clutter background and banding edges, it obtained a good small target detection effect. On COCO experiments, while indicators such as FlOPs did not increase, mAP50 did improve. The experimental detailed results was shown that some part crowded and occluded small targets of the COCO data set were particularly obvious. the mAP50 was 1-3 percentage points higher than other mainstream methods. The part detection effect of COCO data is shown in Table IV.

From the analysis of the small target detection effect diagram, it can be seen that the proposed method can indeed detect the categories and true boundaries of these small targets more accurately. Some part crowded and occluded small targets(person, motorcycle, trafficlight, bird, cat, pizza, person, motorcycle) were also detected. The proposed method can detect the class and true boundary of these small targets more accurately. Therefore, it was proved that the proposed algorithm has certain superiority.

V. Conclusion

The results of comparative experiments show that our algorithm improved for detecting target in terms of accuracy and speed. The SWIoU losses function and the Hard_STP active function can ensure the accuracy of detection, some part crowded and occluded small targets are also detected. Compared with the mainstream deep learning object detection algorithms, the proposed algorithm has the best

TABLE I
COMPARATIVE DATA FOR THE VOC EXPERIMENTS

	mAP50	mAP50-95	val/box_loss	val/obj_loss	val/cls_loss	FLOPs(G)	Time (h)	Model Memory(M)
YOLOv5 with CIOU	0.72188	0.49839	0.045403	0.045403	0.013651	16.1	1.984	13.7
YOLOv5 with DIOU	0.72568	0.50144	0.044892	0.019641	0.012937	16.1	1.984	13.7
YOLOv5 with GIOU	0.72337	0.49858	0.045168	0.019666	0.013052	16.1	1.984	13.7
YOLOv5 with SIOU	0.72512	0.50085	0.036329	0.021692	0.013289	16.1	1.984	13.7
YOLOv5 with EIOU	0.72116	0.4959	0.04545	0.019667	0.013384	16.1	1.984	13.7
YOLOv5 with WIOU	0.72051	0.4935	0.038407	0.022158	0.013326	16.1	1.984	13.7
YOLOv5 with alphaIOU	0.71975	0.50376	0.070898	0.015148	0.013208	16.1	1.984	13.7
YOLOv5 with FocalIOU	0.72412	0.50123	0.027177	0.021784	0.013378	16.1	1.984	13.7
YOLOv5 with SWIOU	0.72273	0.50046	0.036235	0.021883	0.013396	16.1	2.021	14.5
YOLOv5 with SWIOU and HSP_activation function	0.72876	0.51044	0.035247	0.021824	0.013794	15.9	2.014	14.5

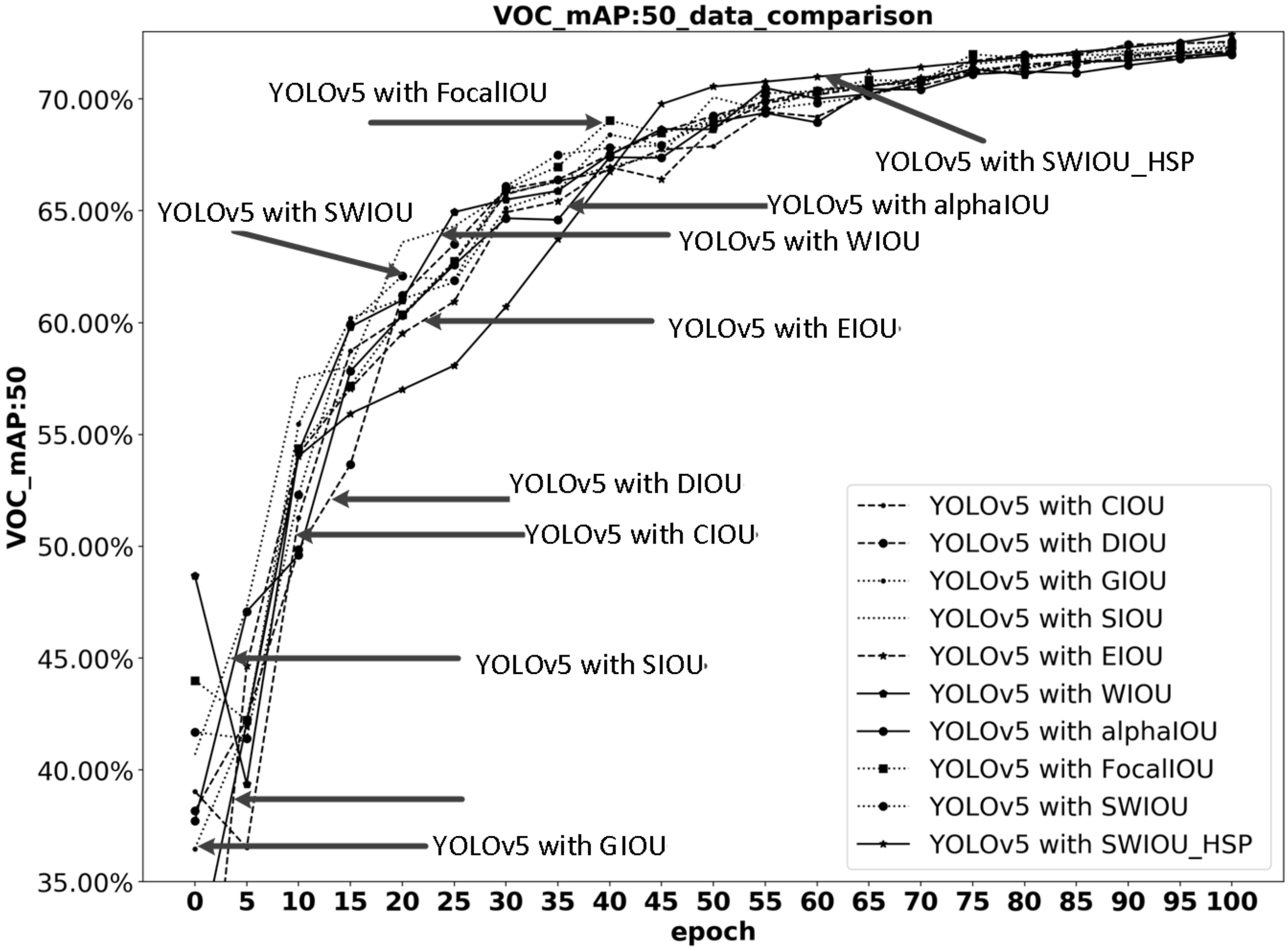

Fig. 6. VOC mAP 50 data comparison

TABLE II
mAP50 OF SOME CROWDED AND OCCLUDED SMALL TARGETS OF THE VOC EXPERIMENTS

	bird	bottle	cow	pottedplant	Sheep	boat
YOLOv5 with CIOU	0.724	0.594	0.695	0.481	0.772	0.553
YOLOv5 with DIOU	0.729	0.596	0.676	0.479	0.766	0.562
YOLOv5 with GIOU	0.721	0.592	0.681	0.475	0.714	0.706
YOLOv5 with SIOU	0.720	0.597	0.682	0.475	0.714	0.627
YOLOv5 with EIOU	0.723	0.592	0.681	0.475	0.773	0.624
YOLOv5 with WIOU	0.724	0.601	0.681	0.478	0.771	0.567
YOLOv5 with alphaIOU	0.724	0.592	0.682	0.475	0.770	0.567
YOLOv5 with FocalIOU	0.725	0.595	0.680	0.472	0.772	0.556
YOLOv5 with SWIOU	0.726	0.609	0.682	0.478	0.775	0.728
YOLOv5 with SWIOU and HSP_activation function	0.728	0.609	0.684	0.488	0.788	0.788

TABLE III
COMPARATIVE DATA FOR THE COCO EXPERIMENTS

	mAP50	mAP50-95	val/box_loss	val/obj_loss	val/cls_loss	FLOPs(G)	Time (h)	Model Memory (M)
YOLOv5 with CIOU	0.56403	0.37073	0.042906	0.048552	0.015517	16.4	25.254	14.8
YOLOv5 with DIOU	0.56507	0.37374	0.043405	0.048557	0.015647	16.4	25.172	14.8
YOLOv5 with GIOU	0.56528	0.36898	0.043352	0.048394	0.015597	16.4	24.771	14.8
YOLOv5 with SIOU	0.56461	0.36985	0.04391	0.048242	0.015512	16.4	25.205	14.8
YOLOv5 with EIOU	0.56397	0.36962	0.054289	0.045108	0.015674	16.4	25.407	14.8
YOLOv5 with WIOU	0.56647	0.36750	0.046753	0.049151	0.015471	16.4	25.418	14.8
YOLOv5 with alphaIOU	0.55698	0.37545	0.078068	0.035975	0.015614	16.4	25.204	14.8
YOLOv5 with FocalIOU	0.55698	0.37545	0.078068	0.035975	0.015614	16.4	25.507	14.8
YOLOv5 with SWIOU	0.56444	0.36819	0.043958	0.038252	0.015517	16.4	24.699	14.8
YOLOv5 with SWIOU and HSP_activation function	0.57145	0.37819	0.042712	0.046153	0.014517	16.4	25.202	14.8

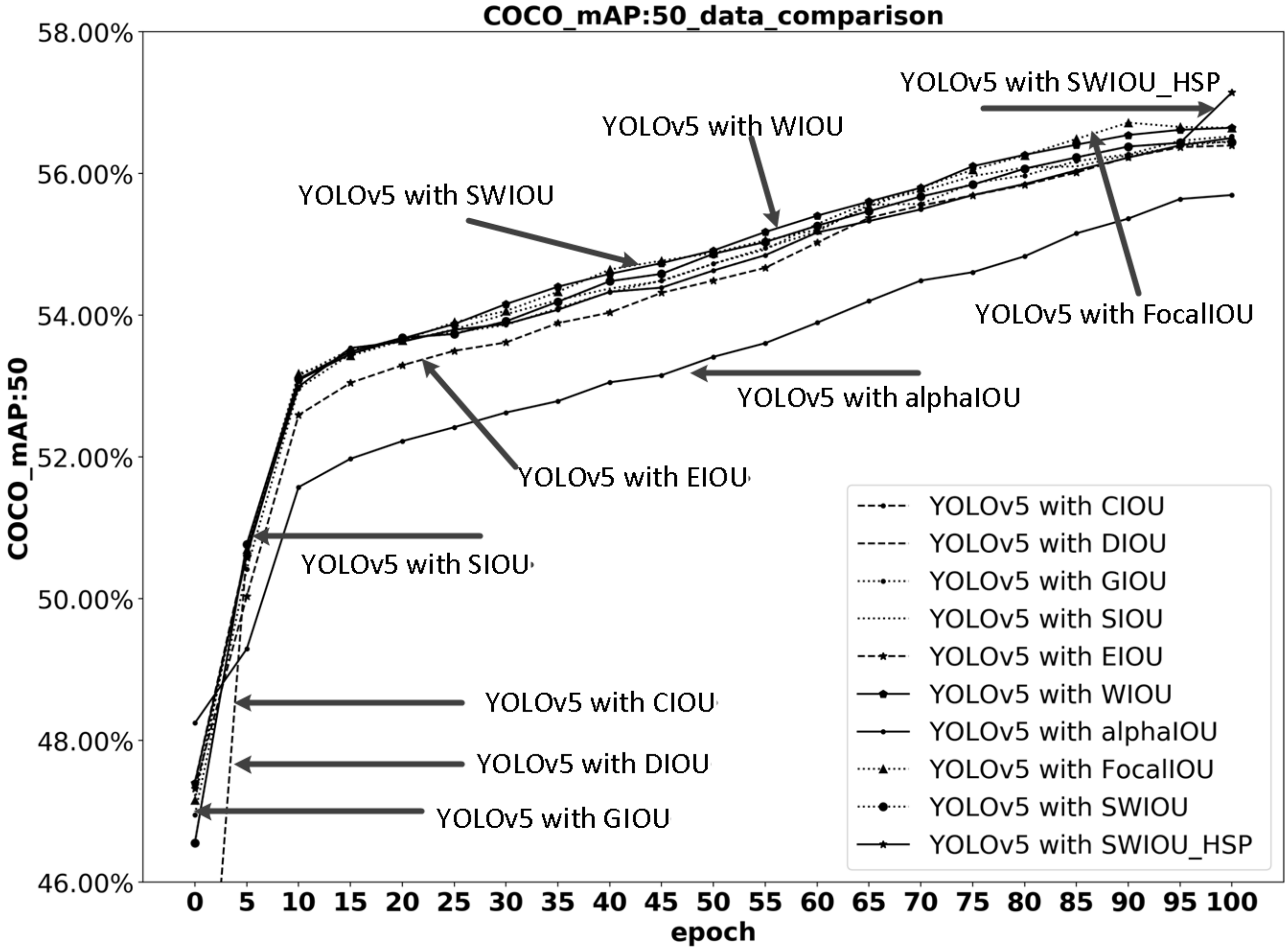

Fig. 7. VOC mAP 50 data comparison

TABLE IV
MAP50 OF SOME CROWDED AND OCCLUDED SMALL TARGETS OF THE COCO EXPERIMENTS

	person	motorcycle	traffic light	bird	cat	pizza	person	Motor-cycle
YOLOv5 with CIOU	0.762	0.672	0.526	0.486	0.83	0.715	0.762	0.672
YOLOv5 with DIOU	0.762	0.672	0.526	0.486	0.83	0.715	0.762	0.672
YOLOv5 with GIOU	0.760	0.677	0.53	0.467	0.825	0.701	0.760	0.677
YOLOv5 with SIOU	0.762	0.672	0.527	0.473	0.834	0.705	0.762	0.672
YOLOv5 with EIOU	0.763	0.672	0.527	0.473	0.834	0.701	0.763	0.672
YOLOv5 with WIOU	0.761	0.665	0.526	0.473	0.826	0.692	0.761	0.665
YOLOv5 with alphaIOU	0.764	0.669	0.494	0.459	0.852	0.687	0.764	0.669
YOLOv5 with FocalIOU	0.763	0.668	0.498	0.465	0.852	0.677	0.763	0.668
YOLOv5 with SWIOU	0.763	0.665	0.526	0.475	0.828	0.674	0.752	0.665
YOLOv5 with SWIOU and HSP_activation function	0.764	0.674	0.526	0.481	0.836	0.704	0.764	0.674

suppression effect on background and noise, obtains absolute suppression effect on background, and obtains more accurate and clean small targets. The future research direction is to streamline the backbone network and introduce new loss functions to improve the detection efficiency. This method has great research potential and great practical application prospect. For example: our algorithm has a positive effect in forests, farmland, urban greening and other environments.

References

[1] T. W. Teng, P. Veerajagadheswar, B. Ramalingam, J. Yin, R. Elara Mohan, and B. F. Gómez, "Vision based wall following framework: A case study with hsr robot for cleaning application," *Sensors*, vol. 20, no. 11, p. 3298, 2020.

[2] J. Yin, K. G. S. Apuroop, Y. K. Tamilselvam, R. E. Mohan, B. Ramalingam, and A. V. Le, "Table cleaning task by human support robot using deep learning technique," *Sensors*, vol. 20, no. 6, p. 1698, 2020.

[3] B. Ramalingam, J. Yin, M. Rajesh Elara, Y. K. Tamilselvam, M. Mohan Rayguru, M. V. J. Muthugala, and B. Félix Gómez, "A human support robot for the cleaning and maintenance of door handles using a deep-learning framework," *Sensors*, vol. 20, no. 12, p. 3543, 2020.

[4] H. Zhang, G. Wang, Y. Li, and H. Wang, "Faster r-cnn, fourth-order partial differential equation and global-local active contour model (fpde-glacm) for plaque segmentation in iv-oct image," *Signal, Image and Video Processing*, vol. 14, no. 3, pp. 509–517, 2020.

[5] L. O. Solis-Sánchez, R. Castañeda-Miranda, J. J. García-Escalante, I. Torres-Pacheco, R. G. Guevara-González, C. L. Castañeda-Miranda, and P. D. Alaniz-Lumbreras, "Scale invariant feature approach for insect monitoring," *Computers and Electronics in Agriculture*, vol. 75, no. 1, pp. 92–99, 2011.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[8] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[9] X. Zhang, Z. Qiu, P. Huang, J. Hu, and J. Luo, "Application research of yolo v2 combined with color identification," in *2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE, 2018, pp. 1381–1383.

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *ArXiv Preprint ArXiv:1804.02767*, 2018.

[11] P. Adarsh, P. Rathi, and M. Kumar, "Yolo v3-tiny: Object detection and recognition using one stage improved model," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*. IEEE, 2020, pp. 687–694.

[12] W. He, Z. Huang, Z. Wei, C. Li, and B. Guo, "Tf-yolo: An improved incremental network for real-time object detection," *Applied Sciences*, vol. 9, no. 16, p. 3225, 2019.

[13] D. Xu and Y. Wu, "Improved yolo-v3 with densenet for multi-scale remote sensing target detection," *Sensors*, vol. 20, no. 15, p. 4276, 2020.

[14] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "Dc-spp-yolo: Dense connection and spatial pyramid pooling based yolo for object detection," *ArXiv preprint ArXiv:1903.08589*, 2019.

[15] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2019.

[16] P. Ren, W. Fang, and S. Djahel, "A novel yolo-based real-time people counting approach," in *2017 International Smart Cities Conference (ISC2)*. IEEE, 2017, pp. 1001–1015.

[17] J. Wang, N. Wang, L. Li, and Z. Ren, "Real-time behavior detection and judgment of egg breeders based on yolo v3," *Neural Computing and Applications*, vol. 32, pp. 5471–5481, 2020.

[18] X. Long, K. Deng, G. Wang, Y. Zhang, Q. Dang, Y. Gao, H. Shen, J. Ren, S. Han, E. Ding *et al.*, "Pp-yolo: An effective and efficient implementation of object detector," *ArXiv preprint ArXiv:2007.12099*, 2020.

[19] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *ArXiv preprint ArXiv:2004.10934*, 2020.

[20] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "Tph-yolov5: Improved yolov5 based on transformer prediction head for object detection on drone-captured scenarios," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2778–2788.

[21] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *ArXiv preprint ArXiv:2209.02976*, 2022.

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[23] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *ArXiv preprint arXiv:1911.08287*, 2017.

[25] Y. Tsung, P. Lin, and R. Goyal, "Focal loss for dense object detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, pp. 2999–3007.

[26] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9627–9636.

[27] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6569–6578.

[28] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 24th ACM International Conference on Multimedia*, 2016, pp. 516–520.

[29] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 658–666.

[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." *ArXiv preprint ArXiv:1911.08287*, 2019.

[31] 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, 2022.

[32] J. He, S. Erfani, X. Ma, J. Bailey, Y. Chi, and X.-S. Hua, "α-iou: A family of power intersection over union losses for bounding box regression," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 230–20 242, 2021.

[33] H. Su and C. Jung, "Perceptual enhancement of low light images based on two-step noise suppression," *IEEE Access*, vol. 6, pp. 7005–7018, 2018.

[34] Y.-J. Cho, "Weighted intersection over union (wiou): A new evaluation metric for image segmentation," *ArXiv preprint arXiv:2107.09858*, 2021.

[35] B. W. Chao Chen, "An image recognition technology based on deformable and cbam convolution resnet50," *IAENG International Journal of Computer Science*, vol. 50, pp. 274–281, 2023.

Chao Chen is an associate professor in Key Laboratory of Numerical Simulation in Sichuan University, Neijiang Normal University, Neijiang, Hongqiao Street 1, P. R. China.
Bin Wu is an expert in computer vision.
Yongguo Shi is a professor of Data Recovery Key Laboratory of Sichuan Province, Neijiang Normal University. He is also a part-time Master's Supervisor of Sichuan Light Chemical Engineering University.