Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(5): 3394–3409.
DOI: 10.3934/era.2025150
Received: 26 February 2025
Revised: 01 April 2025
Accepted: 21 April 2025
Published: 29 May 2025

Research article

ECF-YOLO: An enhanced YOLOv8 algorithm for ship detection in SAR images

Peng Lu[1], **Xinpeng Hao**[1], **Wenhui Li**[2], **Congqin Yi**[1,*], **Ru Kong**[3] **and Teng Wang**[3]

[1] College of Information Technology, Shanghai Ocean University, Shanghai 201306, China
[2] Modern Educational Technology Center, Shanghai Maritime University, Shanghai 201306, China
[3] Shandong Provincial Institute of Land Space Data and Remote Sensing Technology (Shandong Marine Dynamic Monitoring Center), Shandong 250014, China

* **Correspondence:** Email: cqyi@shou.edu.cn.

Abstract: Synthetic aperture radar (SAR) is an advanced microwave sensor widely used in ocean monitoring because of its resilience to light and weather conditions. However, SAR ship detection tends to have relatively low accuracy due to the prevalence of complex backgrounds and small targets in the detection process. To address these issues, we proposed ECF-YOLO, an improved ship detection algorithm based on YOLOv8. The algorithm enhanced the feature extraction ability of the model and reduced the number of parameters and computational cost by developing a novel C2f-EMSCP module, which replaced the original C2f module in the backbone network. Additionally, we proposed the CGFM module in the neck network, which was designed to improve the detection accuracy of small ship targets by selecting features after combining shallow and deep feature maps. Furthermore, the Inner-SIoU loss function was introduced to replace the CIoU, providing a more precise overlap calculation between the target and anchor boxes, thus further improving detection accuracy. The experimental results for the SAR ship detection dataset showed that compared to YOLOv8n, ECF-YOLO improved AP_{75} by 2.8% and $AP_{50:95}$ by 0.9%. Compared to other mainstream algorithms like YOLOv9t, YOLOv10n , and YOLO11n, ECF-YOLO achieved improvements of 3.4%, 4.6% , and 4.9% for AP_{75}, and 3.4%,1.9%, 3.0% for $AP_{50:95}$, respectively, demonstrating its effectiveness for detecting small targets.

Keywords: YOLOv8; ship detection; C2f-EMSCP; CGFM; Inner-SIoU loss function

1. Introduction

Ship detection is crucial in maritime traffic management, border patrol, and safety monitoring. Synthetic aperture radar (SAR) technology, capable of all-weather observation, provides high-quality images unaffected by adverse conditions like clouds, rain, or fog [1]. Traditional SAR ship detection, primarily utilizing constant false alarm rate (CFAR), generally involves land-sea segmentation, CFAR detection, and target discrimination [2–4]. Despite their strengths in leveraging strong scattering echoes for enhanced detection performance without prior information about unknown targets, these methods are sensitive to complex backgrounds and lack adaptability. This leads to decreased effectiveness as background complexity increases [5]. In contrast to CFAR-based ship detection methods, deep learning-based target detection approaches have garnered significant attention in the field of SAR image target detection due to their robust target feature extraction capabilities and superior detection performance. Common deep learning-based target detection algorithms include two-stage detectors and single-stage detectors. Two-stage detectors, such as fast region-based convolutional neural network (Faster R-CNN) [6], Mask R-CNN [7], and cascade R-CNN [8], excel in ship detection but often come with high computational costs. In comparison, single-stage detectors are widely recognized for their efficient detection speed, with classic examples including the YOLO series [9], single shot multiBox detector (SSD) [10], and RetinaNet [11].

To address insufficient detection accuracy in complex SAR scenarios, scholars have developed innovative deep learning solutions with distinct technical emphases. Bhattacharjee et al. [12] introduced S-Net, a lightweight architecture that enhances ship localization precision in SAR imagery while maintaining low computational overhead. Departing from conventional approaches, De Sousa et al. [13] designed a CNN framework operating directly on raw SAR echoes, circumventing traditional image formation processes to achieve near-real-time detection capabilities. For multi-resolution SAR analysis, Humayun et al. [14] developed YOLO-OSD through strategic anchor box customization and backbone network optimization, balancing detection accuracy with computational efficiency. Tang et al. [15] designed the DBW-YOLO model, an improved version of YOLOv7-tiny that integrates deformable convolutional networks (DCNet), BiFormer attention mechanisms, and Wise-IoU loss functions. For SAR-specific challenges, ELLK-Net [16] was proposed to address clutter interference, background variations, multi-scale target discrepancies, and noise contamination through novel architectural designs. Zhao et al. [17] achieved robust ship detection through feature alignment-based adversarial learning. In their subsequent work [18], they optimized discriminative accuracy for unknown classes in open-set domain adaptation (OSDA) tasks by employing dynamic threshold adjustment strategies.

2. Algorithm description

We present ECF-YOLO, an enhanced ship detection framework for SAR imagery developed through systematic modifications to the YOLOv8 [19] architecture. As shown in Figure 1, this architecture consists of three key components: 1) A feature extraction backbone, 2) A multi-scale feature fusion neck, and 3) A task-specific detection head.

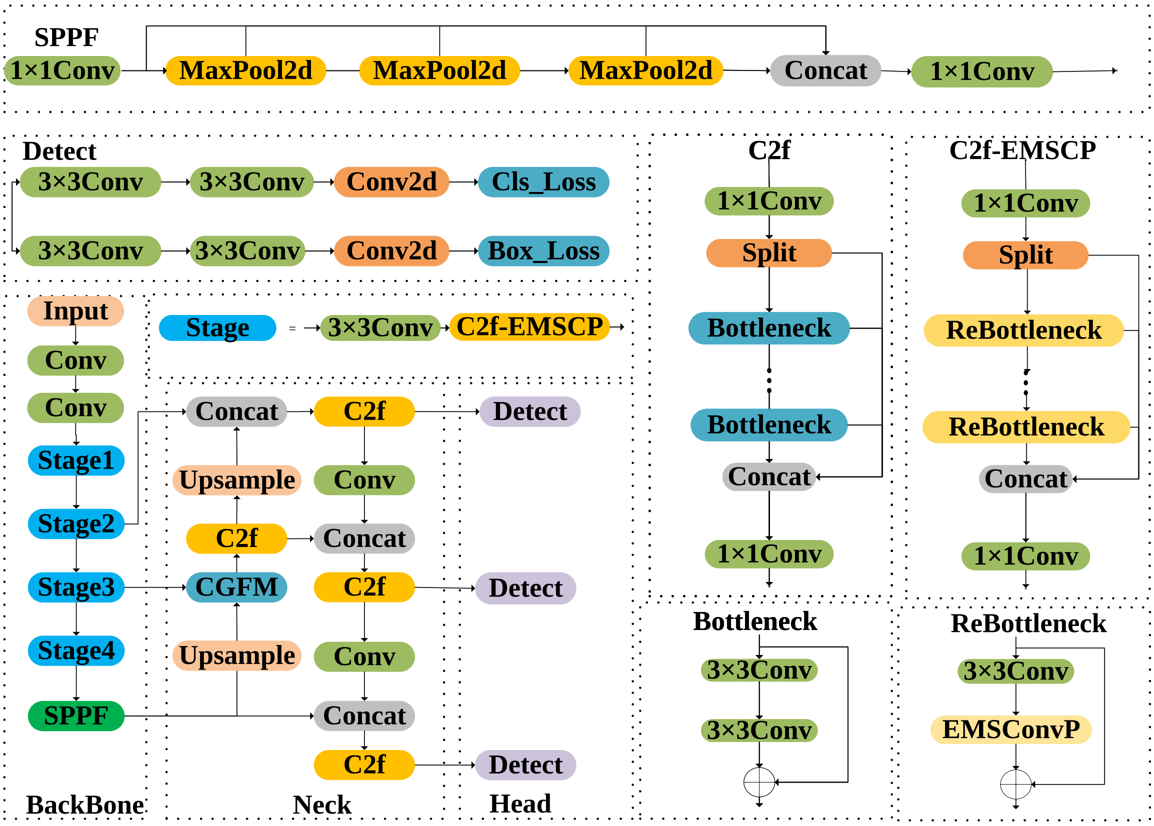

Figure 1. Architecture of the ECF-YOLO network for SAR ship detection.

2.1. C2f-EMSCP Module

We introduce the C2f-EMSCP module, developed by replacing the original Bottleneck in C2f with ReBottleneck,the key difference between Bottleneck and ReBottleneck lay in substituting the second 3 × 3 convolution with EMSConvP. EMSConvP combined multi-scale depthwise separable convolutions and a window multi-head self-attention mechanism (EW-MHSA), enabling the module to capture positional dependencies and enhance global perception. By leveraging multiple sizes of depthwise separable convolutions, it reduced redundancy and effectively extracted multi-scale features, thereby improving both feature extraction and detection accuracy. The specific structure of EMSConvP is illustrated in Figure 2, and the processing steps were outlined as follows: Preprocessing of the input feature map: The EMSConvP module began by normalizing the input feature map X. It then applied the EW-MHSA, where queries (Q) and keys (K) were generated using a single 1 × 1 convolution with shared inputs, thus optimizing computational efficiency. Values (V) were processed through grouped convolutions with 1 × 1 kernels, followed by a ReLU activation to enhance the nonlinear features, producing the feature map X_1. Feature map grouping and Convolution Processing: The resulting feature map X_1 was divided into four channel groups. Each group underwent depthwise separable convolutions (DW-Conv) with different kernel sizes: 1 × 1, 3 × 3, 5 × 5, and 7 × 7. These outputs were concatenated, and feature fusion was performed using a 1 × 1 convolution to generate the feature map X_2. Skip connection: Finally, the original input feature map X

was combined with the processed map X_2 through element-wise addition, resulting in the final output feature map X_3. The implementation of EMSConvP was summarized by the following formula:

$$F = \text{EW-MHSA}(\text{Norm}(X), \text{Act}) \tag{2.1}$$

$$X_{\text{out}} = \text{Conv}\Big(\text{Concat}(\text{DWConv}_{1,3,5,7}(F))\Big) + \text{Skip}(X) \tag{2.2}$$

By replacing the second 3 × 3 convolution in the Bottleneck module of the C2f block with the newly designed EMSConvP, we developed the C2f-EMSCP module. In the YOLOv8 architecture, the C2f module aims to reduce model size while maintaining rich gradient flow. As shown in Figure 1, the C2f module consisted of two 1 × 1 convolutions and a Bottleneck module with residual connections, which included two 3 × 3 convolutions. This structure effectively enhanced feature extraction but may introduce redundancy in the feature maps, thereby limiting the network's expressiveness [20]. To address these issues, we introduced the EMSConvP module, which integrated multi-scale depthwise separable convolutions with a window-based multi-head self-attention mechanism. This combination captured multi-scale feature representations while reducing redundancy, thereby improving model efficiency. Incorporating EMSConvP into the C2f structure resulted in a more compact model with enhanced overall performance. The network architecture of the C2f-EMSCP module is illustrated in Figure 2. Although researchers such as MobileViT [21] and EfficientFormer [22] also employed methods combining depthwise separable convolutions with self-attention mechanisms, the EMSConvP module innovatively integrated parallel multi-scale depthwise separable convolutions (ranging from 1 × 1 to 7 × 7 kernels) with efficient window-based multi-head self-attention. Its multi-branch design enabled simultaneous capture of ship details (e.g., edges and textures) and holistic contours, while filtering key features through attention mechanisms. This significantly enhanced adaptability to complex backgrounds, multi-scale targets, and noise interference. In contrast, MobileViT and EfficientFormer primarily relied on global self-attention or single-scale convolutions. Although these methods excelled at semantic correlation, they tended to overlook local details while incurring high computational costs.

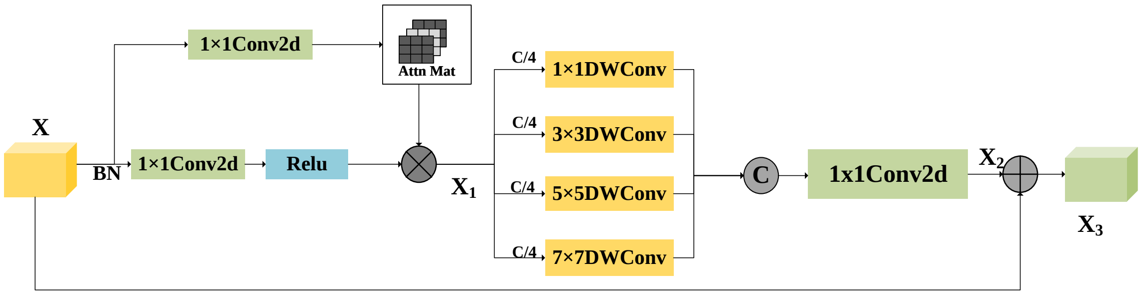

Figure 2. EMSConvP network structure with multi-scale depthwise convolutions and efficient window-based multi-head self-attention mechanism.

2.2. *Content-guided fusion module*

To enhance the detection capability of small target ships and improve overall ship detection accuracy, we proposed the content-guided fusion module (CGFM) in the neck network. The CGFM effectively highlighted key features by performing weighted integration and reorganization of the input features, thereby improving the model's precision in detecting small ships. The network structure of CGFM is illustrated in Figure 3, and its processing was detailed in the following steps: 1) Preprocessing of input and output: The module received two input feature maps, X_1 and X_2, denoted as input1 and input2, respectively. First, input1 was processed as 1×1 convolutions to adjust its channel dimensions to match those of input2, resulting in the output feature map X_3. The feature maps X_2 and X_3 were concatenated concatenated along the channel dimension, producing the output feature map X_4. 2) Feature selection and weighting: The feature map X_4 was processed through global average pooling and global max pooling layers, and the resulting values were summed to produce feature map X_5. Next, X_5 was passed through a Sigmoid activation function to generate weight values for each channel. These weights were applied to the feature maps X_3 and X_2 via element-wise multiplication, enabling adaptive weighting based on feature importance, resulting in the output feature maps X_6 and X_7. 3) Feature reorganization and output: Element-wise addition was performed between the feature maps X_6 and X_2, as well as between X_7 and X_3. This facilitates complementary enhancement of features. Finally, the two resulting feature maps were concatenated to produce the final output feature map X_8.

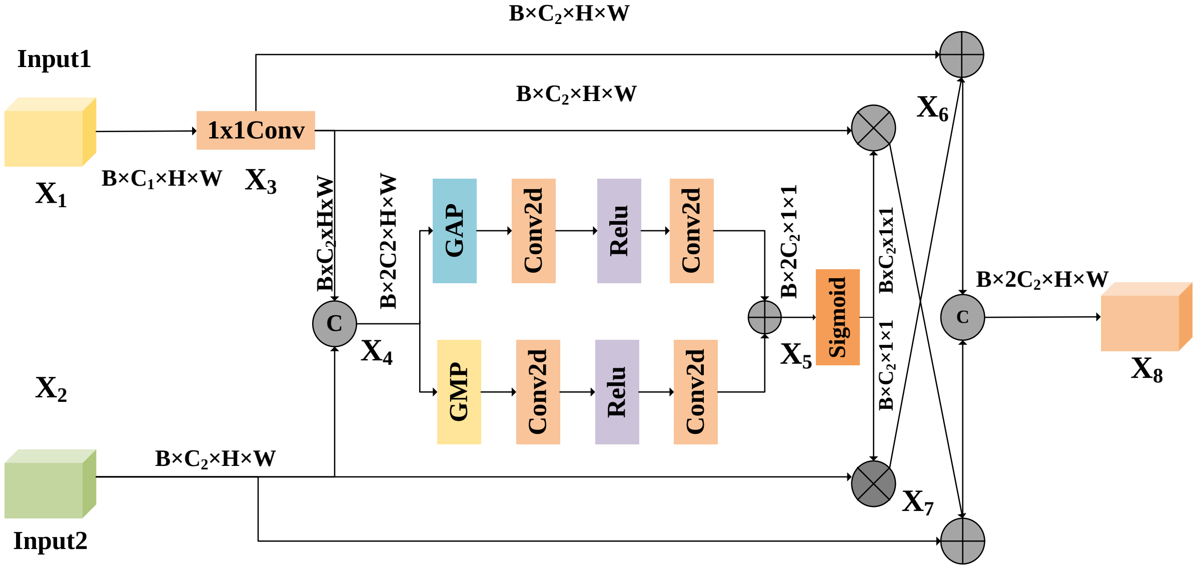
$\mathrm{B\times C_2\times H\times W}$
Input1
1x1Conv
$\mathrm{B\times C_2\times H\times W}$
$\mathrm{X_6}$
$\mathrm{X_1}$
$\mathrm{B\times C_1\times H\times W}$
$\mathrm{X_3}$
$\mathrm{BxC_2xHxW}$
B×2C2×H×W
GAP
Conv2d
Relu
Conv2d
$\mathrm{B\times 2C_2\times 1\times 1}$
$\mathrm{BxC_2x1x1}$
Sigmoid
$\mathrm{B\times 2C_2\times H\times W}$
C
C
$\mathrm{X_4}$
$\mathrm{X_5}$
$\mathrm{X_2}$
GMP
Conv2d
Relu
Conv2d
$\mathrm{B\times C_2\times 1\times 1}$
$\mathrm{X_8}$
$\mathrm{B\times C_2\times H\times W}$
$\mathrm{X_7}$
Input2

Figure 3. CGFM architecture with dual-branch fusion and channel attention.

2.3. *Loss function*

In YOLOv8, the loss function comprised classification and regression losses. The classification loss is computed using binary cross-entropy, while the regression loss included distribution focal loss (DFL) and bounding box regression loss. The total regression loss was defined as follows:

$$f_{\text{loss}} = \lambda_1 f_{\text{DFL}} + \lambda_2 f_{\text{BBRL}} \tag{2.3}$$

DFL refined the standard focal loss by incorporating discrete classification results into continuous outcomes:

$$f_{\text{DFL}}(S_i, S_{i+1}) = -\left((y_{i+1} - y)\log(s_i) + (y - y_i)\log(S_{i+1})\right) \tag{2.4}$$

where $\hat{y}_i$ and $\hat{y}_{i+1}$ are continuous labels surrounding y, and $y = \sum_{i=0}^{n} P(y_i)y_i$, with P applied via softmax. The bounding box regression loss is critical for object detection. In this work, the CIoU loss was replaced by the Inner-SIoU loss [23], which emphasized internal region overlap between the target and anchor boxes, thereby improving detection performance for ships with varying shapes and scales. The Inner-SIoU loss was defined as follows:

$$L_{\text{Inner-SIoU}} = L_{\text{SIoU}} + \text{IoU} - \text{IoU}^{\text{inner}} \tag{2.5}$$

Inner-SIoU enhanced the standard IoU loss by prioritizing overlap in internal regions of target and anchor boxes. Unlike standard IoU (measured via outer box intersection, blue area in Figure 4), Inner-SIoU computed $\text{IoU}^{\text{inner}}$ using intersections between scaled-down inner boxes (orange area), enabling finer boundary alignment. This improved detection accuracy for complex-shaped and multi-scale objects.

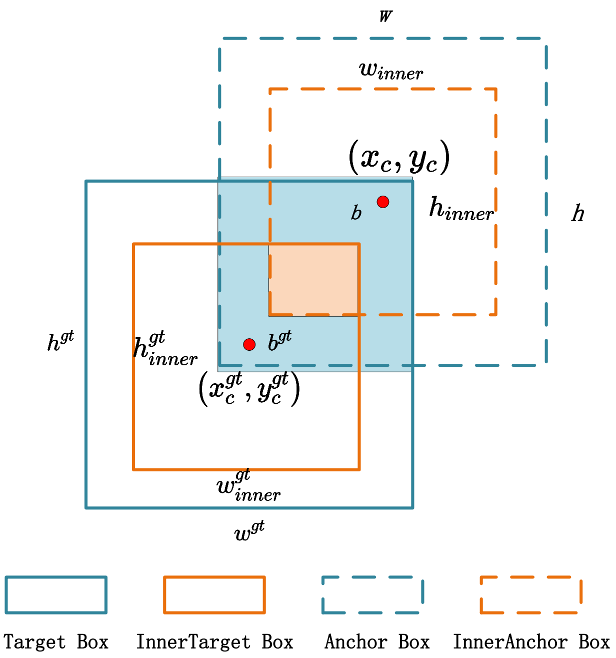

Figure 4. Inner-IoU: Bounding box scaling and overlap metric visualization.

The intersection and union areas inter and union are computed as:

$$b^{gt}_{\{l,r\}} = x^{gt}_c \pm \tfrac{1}{2}w^{gt}\text{ratio}, \quad b^{gt}_{\{t,b\}} = y^{gt}_c \pm \tfrac{1}{2}h^{gt}\text{ratio} \tag{2.6a}$$

$$b_{\{l,r\}} = x_c \pm \tfrac{1}{2}w\text{ratio}, \quad b_{\{t,b\}} = y_c \pm \tfrac{1}{2}h\text{ratio} \tag{2.6b}$$

$$\text{inter} = \left[\min(b_r^{gt}, b_r) - \max(b_l^{gt}, b_l)\right] \cdot \left[\min(b_b^{gt}, b_b) - \max(b_t^{gt}, b_t)\right] \tag{2.7}$$

$$\text{union} = (w^{gt}h^{gt} + wh) \cdot \text{ratio}^2 - \text{inter} \tag{2.8}$$

$$\text{IoU}^{\text{inner}} = \frac{\text{inter}}{\text{union}} \tag{2.9}$$

The ratio typically ranged from 0.5 to 1.5. The SIoU loss is expressed as:

$$L_{\text{SIoU}} = 1 - \text{IoU} + \frac{(\Delta + \Omega)}{2} \tag{2.10}$$

where Δ represents the distance loss, and Ω is the shape loss. The function also incorporated an angle loss Λ, which we describe in detail below.

Angle Loss Λ: The angle loss measured the alignment between the center points of the target and anchor boxes. It is defined as:

$$\Lambda = \sin\left(2\sin^{-1}\left(\frac{\min(|x_c^{gt} - x_c|, |y_c^{gt} - y_c|)}{\sqrt{(x_c^{gt} - x_c)^2 + (y_c^{gt} - y_c)^2} + \epsilon}\right)\right) \tag{2.11}$$

Here, ϵ is a small constant to prevent division by zero. The angle loss Λ encouraged the anchor box to align closer to the nearest coordinate axis. When $\Lambda = 1$, the angle was 45°, while $\Lambda = 0$ indicated alignment along the X-axis or Y-axis.

Distance Loss Δ: After incorporating the angle loss, the distance loss is redefined as:

$$\Delta = \frac{1}{2}\left[\left(1 - e^{-(2-\Lambda)\left(\frac{b_x - b_x^{gt}}{w^c}\right)^2}\right) + \left(1 - e^{-(2-\Lambda)\left(\frac{b_y - b_y^{gt}}{h^c}\right)^2}\right)\right] \tag{2.12}$$

Shape Loss Ω: The shape loss describes the size discrepancy between the target and anchor boxes:

$$\Omega = \frac{1}{2}\left[\left(1 - \exp\left(\frac{|w - w^{gt}|}{\max(w, w^{gt})}\right)\right)^4 + \left(1 - \exp\left(\frac{|h - h^{gt}|}{\max(h, h^{gt})}\right)\right)^4\right] \tag{2.13}$$

Parameter θ determined the weight of the shape loss, typically ranging from 2 to 6. In this study, $\theta = 4$ was used.

3. Experimental results and analysis

3.1. Evaluation metrics and experimental environment

To evaluate the performance of the ECF-YOLO algorithm for ship detection in SAR imagery, we employed standard detection metrics from the COCO dataset. These metrics, computed based on true positives (TP), false positives (FP), and false negatives (FN), included precision (P), recall (R), and average precision (AP), calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{3.1}$$

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

$$AP = \int_0^1 PR\,dR \tag{3.3}$$

In the experiments,YOLOv8 was employed as the baseline model. The training was conducted with a batch size of 16 and for 300 epochs. The software and hardware environments used in this study are listed in Table 1.

Table 1. Experimental software and hardware environment.

Item	Parameter
Operating System	Ubuntu 22.04
Programming Language	Python 3.8.18
CPU	11th Gen Intel Core i7-11700 @ 2.50 GHz
GPU	NVIDIA GeForce RTX 3060
Algorithm Framework	PyTorch 1.13.1

The dataset used in this study is the publicly available SAR ship detection dataset (SSDD) [24], which was used to train and evaluate the model. The SSDD, specifically designed for SAR ship detection, contains images captured from various scenarios such as nearshore, offshore, inland, and port areas. The dataset was divided into training, validation, and test sets with a ratio of 7:1:2. Detailed parameters of the SSDD are presented in Table 2.

Table 2. Detailed parameters of SSDD.

Item	Parameter
Sensor	RadarSat-2, TerraSAR-X, Sentinel-1
Resolution	1 m–15 m
Polarization	HH, VV, VH, HV
Location	Yantai, China; Visakhapatnam, India
Number of Images	1160
Number of Ships	2456

The SSDD contains 1,160 SAR images and 2,456 ship targets. According to [25], these data yield an average of approximately 2.12 ships per image, with detailed statistics summarized in Table 3.

Table 3. Correspondence between number of ships (NoS) and number of images (NoI).

NoS	1	2	3	4	5	6	7	8	9	10	11	12	13
NoI	725	183	89	47	45	16	15	8	4	11	5	3	3

Notes: NoS represents the number of ships per image, and NoI represents the number of images.

3.2. Ablation experiments

To systematically evaluate the effectiveness of the proposed architectural improvements, comprehensive ablation studies were conducted for the SSDD. Using YOLOv8n as the baseline model, we incrementally integrated three key modules to analyze their individual contributions to detection performance. For the evaluation framework, we employed four critical metrics: Mean average precision (mAP) for detection accuracy, parameter count for model complexity, floating-point operations (FLOPs) for computational efficiency, and frames per second (FPS) for inference speed.Quantitative results comparing different module configurations are summarized in Table 4.

Table 4. Ablation study configurations.

Experiment	C2f-EMSCP	CGFM	Inner-SIoU	Combination Type
1	–	–	–	Baseline
2	✓	–	–	Single-module
3	–	✓	–	Single-module
4	–	–	✓	Single-module
5	✓	✓	–	Dual-module
6	–	✓	✓	Dual-module
7	✓	–	✓	Dual-module
8	✓	✓	✓	Full-model

Experiment 1 referred to the use of the original YOLOv8n model (baseline configuration with all proposed modules disabled); Experiment 2 referred to the YOLOv8 model with the addition of the C2f-EMSCP module (single-module enhancement); Experiment 3 referred to the YOLOv8 model with the addition of the CGFM module (single-module enhancement); Experiment 4 referred to the YOLOv8 model with the replacement of the original loss function by the Inner-SIoU loss (single-module enhancement); Experiment 5 referred to the YOLOv8 model with dual-module integration combining C2f-EMSCP and CGFM; Experiment 6 implemented a dual-module configuration combining CGFM with Inner-SIoU loss; Experiment 7 demonstrated another dual-module combination integrating C2f-EMSCP with Inner-SIoU loss; Experiment 8 represented the full-model implementation incorporating all three proposed components (C2f-EMSCP module, CGFM module, and Inner-SIoU loss function).

As shown in Tables 4 and 5, the proposed modules collaboratively enhanced the YOLOv8 baseline model (Experiment 1). The C2f-EMSCP module designed in Experiment 2 reduced model parameters from 3.01M to 2.88M (4.3% reduction) and compressed FLOPs from 8.1G to 7.8G (3.7% reduction), while simultaneously improving AP_{75} by 2.0% and large-target ship AP by 5.9%. These results validated its efficient lightweight feature extraction and enhanced multi-scale ship detection capability. When combined with the CGFM module in Experiment 5, this configuration achieved 1.1% AP improvement for small-target ships and 7.5% AP gain for large-target ships, demonstrating synergistic enhancement of multi-scale detection performance.The CGFM module in Experiment 3 strengthened small-target detection with 0.5% AP and 0.6% AP_{50} improvements. Its integration with the Inner-SIoU loss in Experiment 6 further elevated small-target AP by 1% and $AP_{50:95}$ by 5%. Experiment 4 revealed that substituting CIoU with Inner-SIoU loss significantly enhanced large-target AP by 7.5% while increasing FPS by 2.5%. The combination with C2f-EMSCP in Experiment 7 pushed large-target ship AP to 75.4%. ultimately, ECF-YOLO in Experiment 8 achieved optimal balance: AP_{50} of 98.2%, AP_{75} of 90.3%, and $\text{AP}_{50:95}$ of 73.9%, representing respective improvements of 0.7%, 2.8%, and 0.9% over the baseline. Small-target and large-target APs increased by 1.1% and 6.3%, respectively, with inference speed reaching 128.6 FPS. This demonstrated the synergistic integration of three specialized components: C2f-EMSCP enabled lightweight feature representation and enhanced multi-scale extraction, CGFM facilitated context-aware small-target discrimination, and Inner-SIoU accomplished geometry-adaptive large-target regression, systematically addressing critical challenges in SAR ship detection.

Table 5. Ablation study results.

Experiment	AP_{50}	AP_{75}	$AP_{50:95}$	AP_S	AP_M	AP_L	**Parameters (M)**	**FLOPs (G)**	**FPS**
1	97.5%	87.5%	73.0%	69.1%	81.1%	64.6%	3.01	8.1	208.4
2	97.4%	89.5%	73.0%	69.0%	80.4%	70.5%	2.88	7.8	135.7
3	98.1%	88.3%	73.3%	69.6%	80.0%	63.7%	3.06	8.1	195.2
4	97.1%	88.8%	73.1%	69.8%	79.1%	72.1%	3.01	8.1	210.9
5	97.4%	89.6%	73.7%	70.2%	80.0%	72.1%	2.93	7.8	128.5
6	97.3%	88.3%	73.5%	70.1%	80.4%	63.3%	3.06	8.1	192.2
7	98.0%	90.3%	73.3%	69.4%	79.8%	75.4%	2.88	7.8	135.9
8	98.2%	90.3%	73.9%	70.2%	80.4%	70.9%	2.93	7.8	128.6

To validate the detection performance of the improved ECF-YOLO model, we presented comparative curves illustrating the variations in AP_{50} and $AP_{50:95}$ metrics between YOLOv8 and ECF-YOLO during the training process, as shown in Figure 5. In the AP_{50} curve analysis, both models exhibited comparable performance as training epochs progressed, with minimal discrepancies observed between ECF-YOLO and the baseline YOLOv8 model under the lower IoU threshold (IoU = 0.5). However, under more stringent evaluation criteria across the extended IoU threshold range (0.5 to 0.95), as demonstrated in the $AP_{50:95}$ curve, ECF-YOLO achieved marginally superior values in later training phases compared to YOLOv8. This observation suggested ECF-YOLO's enhanced capability in multi-scale object detection tasks. These findings collectively indicated that ECF-YOLO maintained parity with YOLOv8 in detection accuracy and stability while exhibiting superior performance characteristics during critical training stages.

As shown in Figure 6, the comparison of test images demonstrated the detection performance. Group a illustrated ship detection results in a complex port environment. From the comparison, ECF-YOLO accurately detected ships, whereas YOLOv8 exhibited false positives. Group b presented ship detection under complex backgrounds. ECF-YOLO showed greater robustness, accurately detecting ship positions even in the presence of significant background noise, while YOLOv8 produced false positives in certain areas. Group c displayed the detection results for small target ships. Both ECF-YOLO and YOLOv8 could identify ship targets, but ECF-YOLO achieved a higher overall confidence threshold.

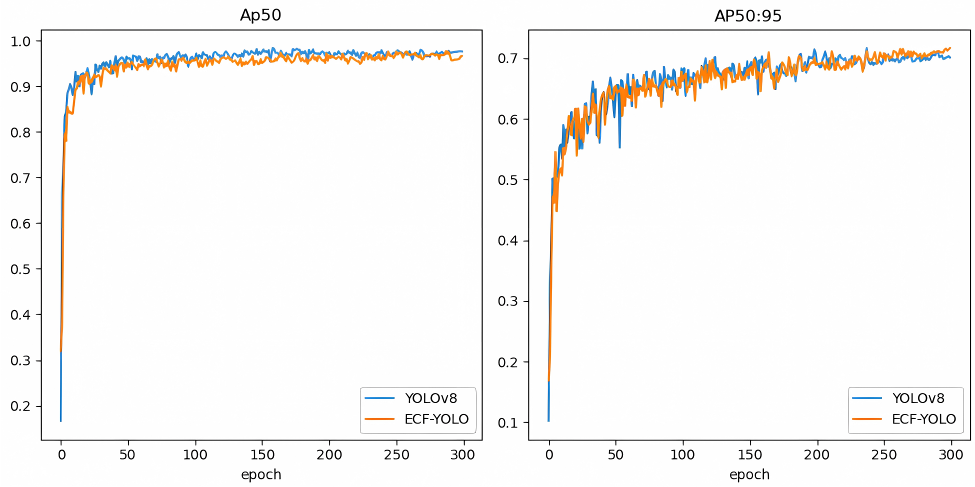

Figure 5. Training performance comparison of YOLOv8 and ECF-YOLO on AP50 and AP50:95.

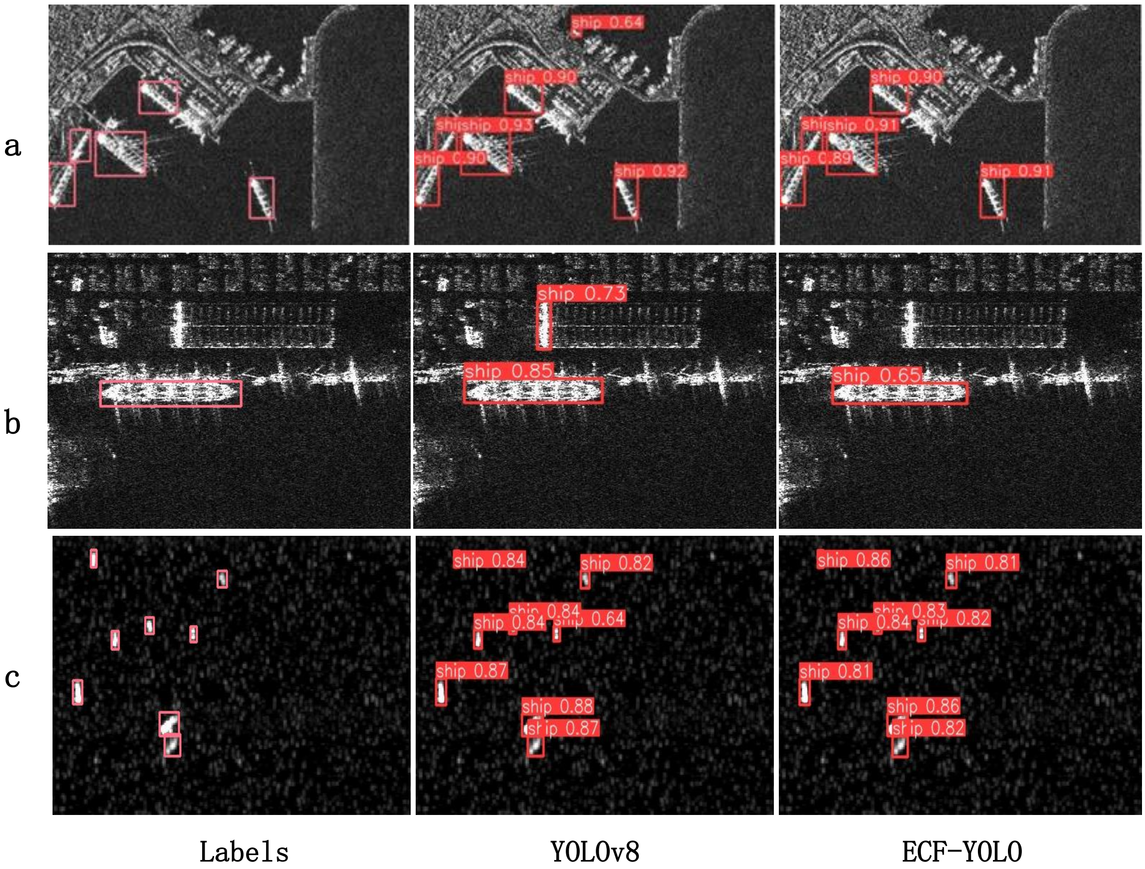

Figure 6. Test image effect comparison diagram.

3.3. Comparative analysis of IoU-Based loss functions

To evaluate the impact of the Inner-SIoU loss function on ship detection, we conducted comparative experiments on the YOLOv8 network structure using the loss functions PIoU [26], DIoU [27], CIoU [28], GIoU [29], SIoU [30], MPDIoU [31], ShapeIoU [32], and Inner-SIoU [23]. The experimental results are shown in Table 6.Table 6 demonstrates that Inner-SIoU outperformed other IoU-based loss functions on the SSDD. It provided significant improvements, especially for small and large target ships. The enhanced overlap alignment between predicted and ground truth boxes achieved by Inner-SIoU contributed to more precise localization, even for varying ship sizes and shapes.

Table 6. Comparative effects of different IoU on ship detection.

IoU	AP_{50}	AP_{75}	$AP_{50:95}$	AP_S	AP_M	AP_L
GIoU	97.1%	88.0%	72.2%	68.7%	78.8%	70.6%
DIoU	97.2%	87.1%	72.2%	68.5%	79.4%	62.5%
CIoU	97.5%	87.5%	73.0%	69.1%	81.1%	62.4%
SIoU	98.1%	89.0%	72.9%	68.5%	80.8%	69.6%
ShapeIoU	97.3%	87.2%	72.0%	67.9%	79.5%	75.5%
MPDIoU	97.9%	88.1%	72.9%	69.7%	79.5%	68.3%
PIoU	97.1%	87.0%	72.2%	68.8%	78.8%	72.7%
Inner-SIoU	97.1%	88.8%	73.1%	69.8%	79.1%	72.1%

3.4. Comparative experiments

To validate the superiority of the proposed ECF-YOLO model, a comparative study was conducted against state-of-the-art object detection methods, including Faster R-CNN [6], Mask R-CNN [7], YOLOv7 [33], TOOD [34], YOLOv10 [35], YOLOX [36], YOLOv9 [37], YOLO11 [19], YOLOv12 [38] and RT-DETR [39]. The results are summarized in Table 7.

Table 7. Comparison between algorithms.

Model	AP_{50}	AP_{75}	$AP_{50:95}$	**Parameters(M)**	**FLOPs(G)**
Faster-RCNN	96.6%	90.6%	73.3%	60.34	250
Mask-RCNN	97.4%	89.0%	71.8%	62.96	302
TOOD	97.1%	89.0%	73.8%	32.02	174
YOLOv10n	97.2%	85.7%	72.0%	2.69	8.2
YOLOX-tiny	98.0%	87.8%	70.8%	5.03	7.57
YOLOv9t	97.9%	86.9%	70.5%	2.62	10.7
YOLOv8n	97.5%	87.5%	73.0%	3.01	8.1
YOLOv7-tiny	96.8%	81.2%	66.9%	6.01	13.0
YOLO11n	97.4%	85.4%	70.9%	2.58	6.3
RTDETR-l	97.2%	91.9%	75.6%	28.45	100.6
YOLOv12n	97.4%	85.1%	69.3%	2.56	6.3
ECF-YOLO	**98.2%**	**90.3%**	**73.9%**	**2.93**	**7.8**

Table 7 demonstrates that ECF-YOLO achieved superior performance across multiple metrics. While maintaining a lightweight architecture, it obtained the highest AP_{50} of 98.2% among all compared models and the second-best AP_{75} of 90.3%, surpassed only by RTDETR at 91.9%. The proposed model achieved a competitive $AP_{50:95}$ of 73.9%, outperforming most YOLO variants while maintaining significantly lower computational complexity. Notably, ECF-YOLO attained these results with only 2.93M parameters and 7.8G FLOPs-demonstrating 22.8 times higher efficiency compared to Mask R-CNN with 62.96M parameters and 12.8 times higher computational efficiency than RTDETR at 100.6G FLOPs. This exceptional balance between detection accuracy and operational efficiency positioned ECF-YOLO as particularly suitable for deployment in resource-constrained environments.

4. Conclusions

In this study, we address the challenges of detecting ships in SAR imagery with complex backgrounds and small targets by proposing the ECF-YOLO model. The model incorporates several key improvements: The integration of the C2f-EMSCP module, which enhances multi-scale feature extraction and reduces model parameters; the incorporation of the CGFM module, which improves the detection of small target ships through effective feature selection; and the adoption of the Inner-SIoU loss function, which provides smoother gradients and more accurate bounding box localization. Experimental results on the SSDD dataset achieves a favorable balance between detection accuracy and computational efficiency. In the future, we will focus on enhancing generalization capability, lightweight design, and detection precision for more diverse applications.

Use of AI tools declaration

The authors declare we have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was supported by the project "Key Technologies for Intelligent Extraction of Remote Sensing Monitoring of Marine Areas and Islands", grant number SDGP370000000202402001009A 001.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, G. Kuang, BiFA-YOLO: A novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images, *Remote Sens.*, **13** (2021), 4209. https://doi.org/10.3390/RS13214209
2. H. Li, W. Hong, Y. Wu, P. Fan, An efficient and flexible statistical model based on generalized gamma distribution for amplitude SAR images, *IEEE Trans. Geosci. Remote Sens.*, **48** (2010), 2711–2722. https://doi.org/10.1109/TGRS.2010.2041239

3. A. Achim, E. E. Kuruoglu, J. Zerubia, SAR image filtering based on the heavy-tailed Rayleigh model, *IEEE Trans. Image Process.*, **15** (2006), 2686–2693. https://doi.org/10.1109/TIP.2006.877362

4. A. Breloy, G. Ginolhac, F. Pascal, P. Forster, CFAR property and robustness of the lowrank adaptive normalized matched filters detectors in low rank compound Gaussian context, in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, (2014), 301–304. https://doi.org/10.1109/SAM.2014.6882401

5. J. Li, C. Xu, H. Su, L. Gao, T. Wang, Deep learning for SAR ship detection: Past, present and future, *Remote Sens.*, **14** (2022), 2712. https://doi.org/10.3390/RS14112712

6. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

7. K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in *2017 IEEE International Conference on Computer Vision (ICCV)*, (2017), 2980–2988. https://doi.org/10.1109/ICCV.2017.322

8. Z. Cai, N. Vasconcelos, Cascade R-CNN: High quality object detection and instance segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 1483–1498. https://doi.org/10.1109/TPAMI.2019.2956516

9. L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, et al., A survey of deep learning-based object detection, *IEEE Access*, **7** (2019), 128837–128868. https://doi.org/10.1109/ACCESS.2019.2939201

10. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, et al., SSD: Single shot multibox detector, in *Computer Vision-ECCV 2016* , **9905** (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

11. T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, **42** (2020), 318–327. https://doi.org/10.1109/TPAMI.2018.2858826

12. S. Bhattacharjee, P. Shanmugam, S. Das, A deep-learning-based lightweight model for ship localizations in SAR images, *IEEE Access*, **11** (2023), 94415–94427. https://doi.org/10.1109/ACCESS.2023.3310539

13. K. De Sousa, G. Pilikos, M. Azcueta, N. Floury, Ship detection from raw SAR echoes using convolutional neural networks, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **17** (2024), 9936–9944. https://doi.org/10.1109/JSTARS.2024.3399021

14. M. F. Humayun, F. A. Nasir, F. A. Bhatti, M. Tahir, K. Khurshid, YOLO-OSD: Optimized ship detection and localization in multiresolution SAR satellite images using a hybrid data-model centric approach, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **17** (2024), 5345–5363. https://doi.org/10.1109/JSTARS.2024.3365807

15. X. Tang, J. Zhang, Y. Xia, H. Xiao, DBW-YOLO: A high-precision SAR ship detection method for complex environments, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **17** (2024), 7029–7039. https://doi.org/10.1109/JSTARS.2024.3376558

16. J. Shen, L. Bai, Y. Zhang, M. C. Momi, S. Quan, Z. Ye, ELLK-Net: An efficient lightweight large kernel network for SAR ship detection, *IEEE Trans. Geosci. Remote Sens.*, **62** (2024), 1–14. https://doi.org/10.1109/TGRS.2024.3451399

17. S. Zhao, Z. Zhang, W. Guo, Y. Luo, An automatic ship detection method adapting to different satellites SAR images with feature alignment and compensation loss, *IEEE Trans. Geosci. Remote Sens.*, **60** (2022), 1–17. https://doi.org/10.1109/TGRS.2022.3160727

18. S. Zhao, Y. Zhang, Y. Luo, Y. Kang, H. Wang, Dynamically self-training open set domain adaptation classification method for heterogeneous SAR image, *IEEE Geosci. Remote Sens. Lett.*, **21** (2024), 1–5. https://doi.org/10.1109/LGRS.2024.3360006

19. GitHub, Ultralytics, 2023. Available from: https://github.com/ultralytics/ultralytics.

20. K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, C. Xu, GhostNet: More features from cheap operations, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 1577–1586. https://doi.org/10.1109/CVPR42600.2020.00165

21. D. Qin, C. Leichner, M. Delakis, M. Fornoni, S. Luo, F. Yang, et al., MobileNetV4: Universal models for the mobile ecosystem, in *Computer Vision-ECCV 2024*, **15098** (2024), 78–96. https://doi.org/10.1007/978-3-031-73661-2_5

22. Y. Li, J. Hu, Y. Wen, G. Evangelidis, K. Salahi, Y. Wang, et al., Rethinking vision transformers for MobileNet size and speed, in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, (2023), 16889–16900.

23. H. Zhang, C. Xu, S. Zhang, Inner-IoU: More effective intersection over union loss with auxiliary bounding box, preprint, arXiv:2311.02877.

24. T. Zhang, X. Zhang, J. Li, X. Xu, B. Wang, X. Zhan, et al., SAR ship detection dataset (SSDD): Official release and comprehensive data analysis, *Remote Sens.*, **13** (2021), 3690. https://doi.org/10.3390/RS13183690

25. J. Li, C. Qu, J. Shao, Ship detection in SAR images based on an improved faster R-CNN, in *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, (2017), 1–6. https://doi.org/10.1109/BIGSARDATA.2017.8124934

26. Z. Chen, K. Chen, W. Lin, J. See, H. Yu, Y. Ke, et al., PIoU Loss: Towards accurate oriented object detection in complex environments, in *Computer Vision-ECCV 2020*, **12350** (2020), 195–211. https://doi.org/10.1007/978-3-030-58558-7_12

27. Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and better learning for bounding box regression, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 12993–13000. https://doi.org/10.1609/AAAI.V34I07.6999

28. Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, et al., Enhancing geometric factors in model learning and inference for object detection and instance segmentation, *IEEE Trans. Cybern.*, **52** (2022), 8574–8586. https://doi.org/10.1109/TCYB.2021.3095305

29. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 658–666. https://doi.org/10.1109/CVPR.2019.00075

30. Z. Gevorgyan, SIoU Loss: More powerful learning for bounding box regression, preprint, arXiv:2205.12740.

31. S. Ma, Y. Xu, MPDIoU: A loss for efficient and accurate bounding box regression, preprint, arXiv:2307.07662.

32. H. Zhang, S. Zhang, Shape-IoU: More accurate metric considering bounding box shape and scale, preprint, arXiv:2312.17663.

33. C. Wang, A. Bochkovskiy, H. M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 7464–7475. https://doi.org/10.1109/CVPR52729.2023.00721

34. C. Feng, Y. Zhong, Y. Gao, M. R. Scott, W. Huang, TOOD: Task-aligned one-stage object detection, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2021), 3490–3499. https://doi.org/10.1109/ICCV48922.2021.00349

35. A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, et al., YOLOv10: Real-time end-to-end object detection, preprint, arXiv:2405.14458.

36. Z. Ge, S. Liu, F. Wang, Z. Li, J. Sun, YOLOX: Exceeding YOLO series in 2021, preprint, arXiv:2107.08430.

37. C. Wang, I. Yeh, H. M. Liao, YOLOv9: Learning what you want to learn using programmable gradient information, in *Computer Vision-ECCV 2024* , **15089** (2024), 1–21. https://doi.org/10.1007/978-3-031-72751-1_1

38. Y. Tian, Q. Ye, D. Doermann, YOLOv12: Attention-centric real-time object detectors, preprint, arXiv:2502.12524.

39. Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, et al., DETRs beat YOLOs on real-time object detection, in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2024), 16965–16974. https://doi.org/10.1109/CVPR52733.2024.01605