Computer Systems
Science and Engineering

Tech Science Press

Doi:10.32604/csse.2025.059634

ARTICLE

OD-YOLOv8: A Lightweight and Enhanced New Algorithm for Ship Detection

Zhuowei Wang[1,*], **Dezhi Han**[1], **Bing Han**[2] **and Zhongdai Wu**[2]

[1]School of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China
[2]Shanghai Ship and Shipping Research Institute Co., Ltd., Shanghai, 200135, China

*Corresponding Author: Zhuowei Wang. Email: 15257584005@163.com

Received: 13 October 2024; Accepted: 28 February 2025; Published: 09 April 2025

ABSTRACT: Synthetic Aperture Radar (SAR) has become one of the most effective tools in ship detection. However, due to significant background interference, small targets, and challenges related to target scattering intensity in SAR images, current ship target detection faces serious issues of missed detections and false positives, and the network structures are overly complex. To address this issue, this paper proposes a lightweight model based on YOLOv8, named OD-YOLOv8. Firstly, we adopt a simplified neural network architecture, VanillaNet, to replace the backbone network, significantly reducing the number of parameters and computational complexity while ensuring accuracy. Secondly, we introduce a dynamic, multi-dimensional attention mechanism by designing the ODC2f module with ODConv to replace the original C2f module and using GSConv to replace two down-sampling convolutions to reduce the number of parameters. Then, to alleviate the issues of missed detections and false positives for small targets, we discard one of the original large target detection layers and add a detection layer specifically for small targets. Finally, based on a dynamic non-monotonic focusing mechanism, we employ the Wise-IoU (Intersection over Union) loss function to significantly improve detection accuracy. Experimental results on the HRSID dataset show that, compared to the original YOLOv8, OD-YOLOv8 improves mAP@0.5 and mAP@0.5–0.95 by 2.7% and 3.5%, respectively, while reducing the number of parameters and GFLOPs by 72.9% and 4.9%, respectively. Moreover, the model also performs exceptionally well on the SSDD dataset, with AP and AP50 increasing by 1.7% and 0.4%, respectively. OD-YOLOv8 achieves an excellent balance between model lightweightness and accuracy, making it highly valuable for end-to-end industrial deployment.

KEYWORDS: Object detection; YOLOv8; VanillaNet; Wise-IoU; lightweight

1 Introduction

With the rapid development of the global marine economy, the marine industry has become an important component of global economic civilization development [1]. Traditional methods involve maritime staff monitoring images to supervise ships and gather information. However, this approach is susceptible to visual fatigue and interference from complex environments, leading to potential safety hazards [2]. With the continuous advancement of deep learning, deep learning methods have become the mainstream approach for target detection in remote sensing images due to their outstanding feature representation capabilities [3]. However, due to factors such as high traffic intensity in some sea areas and complex climatic conditions, the accuracy of ship identification still has shortcomings and poses considerable risks [4].

As an active microwave imaging sensor, the Synthetic Aperture Radar (SAR) system has unique advantages in applications such as environmental monitoring, disaster monitoring, resource exploration, ocean monitoring, crop yield estimation, mapping, and military fields [5]. SAR, with its all-weather, all-day,

weather-independent, wide coverage, and altitude-independent characteristics [6], has been widely used in both military and civilian domains.

SAR ship detection algorithms can be divided into traditional ship detection algorithms and deep learning-based ship detection algorithms. Spectral Residual (SR) [7] and Constant False Alarm Rate (CFAR) [8] are the most representative traditional detection algorithms. They primarily perform statistical analysis on image pixels using thresholds [9]. The CFAR method automatically adjusts the threshold based on a given false alarm rate and distinguishes targets from the calculated background threshold using the estimated statistical distribution [10]. However, these algorithms generally lack sufficient feature extraction capabilities, heavily rely on the statistical distribution of ocean clutter, and are easily affected by speckle noise and complex environments. The complexity and significant computational load of these methods also fail to meet real-time detection requirements effectively [11]. In recent years, with the continuous development of deep learning technology, its applications in image segmentation, image classification, and object recognition have become increasingly widespread, covering areas such as autonomous driving, medical image analysis, and facial recognition. Using deep learning for ship target detection has become the current trend. However, ship target detection is often influenced by factors such as weather, lighting, ship size variations, and background interference. Deep learning models, by learning from massive datasets, can more effectively detect ships of different scales, thereby improving detection accuracy and robustness in complex environments.

Currently, target detection algorithms are mainly divided into two categories: two-stage and one-stage algorithms. Two-stage target detection algorithms involve two steps: generating candidate regions and then classifying and finely locating the targets. Common two-stage detection algorithms include R-CNN [12] and Faster R-CNN [13]. These algorithms can more precisely locate and classify targets, usually offering higher accuracy and better adaptation to complex scenes compared to one-stage algorithms. However, the model structure of two-stage algorithms is more complex, and generating candidate regions requires substantial computational resources, which is not conducive to real-time monitoring. Representative one-stage target detection algorithms include YOLO [14], SSD [15], and RetinaNet [16]. Unlike two-stage algorithms, one-stage detection algorithms do not need an additional candidate region generation step but directly predict the location and category of targets across the entire image. Overall, one-stage algorithms are characterized by their end-to-end, simple, and fast nature, making them suitable for scenarios that demand real-time monitoring.

Currently, most target detection algorithms improve detection accuracy by stacking numerous network structures and increasing computational load. However, as the number of network layers increases, gradients may gradually diminish to near zero during backpropagation, making deeper networks difficult to train. ResNet alleviates the issue of gradient vanishing by introducing residual blocks, allowing gradients to flow directly to shallower layers, thereby enhancing training stability and network performance. Nevertheless, ResNet requires storing feature maps from previous layers in memory during training, resulting in substantial off-chip memory traffic, which is disadvantageous for real-time monitoring in resource-constrained environments [17].

Additionally, massive network structures are not conducive to real-time ship detection and fail to meet the requirements for end-to-end industrial deployment. Although model lightweighting can reduce computational load and parameters, it may lead to a decrease in accuracy to some extent, thereby reducing detection precision. Most existing ship target detection models have not successfully balanced accuracy and efficiency. How to maintain high efficiency while ensuring that the model has sufficient generalization capability and robustness to adapt to various complex situations has become an urgent problem to solve.

To address the issues in SAR-based ship target detection, this paper proposes a new lightweight algorithm called OD-YOLOv8, which is an improvement based on YOLOv8. The main research work and contributions of this paper are summarized as follows:

(1) The original complex 10-layer backbone network of YOLOv8 has been discarded in favor of the elegant backbone architecture VanillaNet, recently proposed by Huawei Noah's Ark Lab and the University of Sydney. This new structure avoids self-attention, high depth, and numerous residual operations, resulting in a lightweight and efficient network architecture that fully demonstrates the charm of minimalism in deep learning.

(2) We have improved the neck of YOLOv8 by integrating ODConv into the C2f module, resulting in a new network structure called ODC2f. Additionally, we replaced the convolution operations in the downsampling stages with GSConv, which enhances the model's detection capability while maintaining its lightweight nature.

(3) We removed the original detection head designed for large objects and added a new detection head specifically for small objects, thereby enhancing the detection capability for small targets. Additionally, we employed the WIoU loss function to mitigate the adverse gradient issues caused by low-quality samples. WIoU uses the outlier degree as the regression loss value for bounding boxes, rather than relying solely on the Intersection over Union (IoU). This effectively suppresses the interference from background boxes, thereby improving the performance of object detection.

(4) Extensive experiments were conducted on the HRSID and SSDD datasets, resulting in significant performance improvements. On the HRSID dataset, the mAP@0.5 increased from 0.898 to 0.925, and the mAP@0.5–0.95 rose from 0.655 to 0.69, representing improvements of 2.7% and 3.5%, respectively. Additionally, the number of parameters was dramatically reduced from 3.011 million to 816 thousand, which is only 27.1% of the original amount. The GFlops decreased from 8.2 to 7.8, a reduction of 4.9%. Notably, the model also achieved excellent results on the SSDD dataset, with AP and AP50 increasing by 1.7% and 0.4%, respectively. This model successfully balances lightweight design with high accuracy.

The remainder of this paper is organized as follows: In Section 2, we introduce the related work on ship target detection. Section 3 discusses the proposed methods. In Section 4, we analyze the experimental results. Finally, Section 5 summarizes our work and provides an outlook for future research.

2 Related Work

To address the issues of low detection accuracy caused by multi-scale, complex environments, and background interference, as well as the severe problems of missed detections and false detections of small targets and high model complexity in ship detection, many scholars have made improvements and attempts.

2.1 Research on Improving Low Detection Accuracy

In the research on improving low detection accuracy, Cui et al. [18] proposed a novel detection method based on a Dense Attention Pyramid Network (DAPN) using SAR images. This method connects Convolutional Block Attention Modules (CBAM) to each cascaded feature map through a pyramid structure, thereby improving detection accuracy. Zhang et al. [19] designed a new Cross-Scale Region Prediction Perception Network (CSRP-Net) and introduced a Cross-Scale Self-Attention (CSSA) module to tackle complex environments. Wang et al. [20] designed a component called the Dilated Convolution Feature Enhancement Module (DFEM) and integrated it into the backbone network, proposing a new Feature Pyramid Network called NAS-FPN. Li et al. [21] proposed an Attention-Guided Balanced Feature Pyramid Network (A-BFPN) and designed a Channel Attention-Guided Fusion Network (CAFN) model to reduce aliasing effects in feature maps. Yang et al. [9] introduced an improved single-stage object detection

framework that combines the ideas of RetinaNet and Rotatable Bounding Boxes (RBox). Zhao et al. [22] designed an Attention Perception Pyramid Network (ARPN), which adopts Convolutional Block Attention Modules (CBAM) and Receptive Field Blocks (RFB) to construct a top-down fine-grained feature pyramid.

2.2 Research on Severe Missed and False Detections of Small Items

To address the issue of serious missed and false detections of small objects, Guo et al. [23] designed a single-pole detector called CenterNet++. They proposed a head enhancement module to mitigate the impact of complex backgrounds and designed a feature refinement module to enhance the detection of small objects. Zhu et al. [24] proposed an anchor-free detection method based on FCOS (Fully Convolutional One-Stage Object Detection), redesigning sample definition and feature extraction to reduce the influence of anchor effects on detection. Liu et al. [25] improved YOLOv5 and introduced YOLO-Extract. By integrating a coordinate attention mechanism into the network, the model better captures important features in both spatial and channel dimensions. Additionally, a residual network was introduced to better capture fine-grained features, thereby improving the detection of small objects. Wang et al. [26] proposed an improved real-time framework based on YOLOv3, focusing on enhancing the detection accuracy of small objects in remote sensing datasets. This method simplifies the network structure by removing some large and medium-sized layers and increasing the detection weight for small objects to achieve better detection performance for small objects.

2.3 Research on Implementing Model Lightweighting

To achieve model lightweighting, Chen et al. [27] proposed a lightweight network model based on YOLOv3, significantly reducing the number of parameters by leveraging the ShuffleNetv2 network. Fan et al. [28] introduced a detection algorithm based on an improved RetinaNet, enhancing network depth and feature extraction capability by incorporating grouped convolutions and attention mechanisms. Ma et al. [29] proposed a lightweight model named YOLOv8n-ShuffleNetv2-Ghost-SE. This model replaces the Conv module with the Ghost module, substitutes the C2f module in the Neck part with C2fGhost, and alternately connects the ShuffleNetv2 base module and downsampling module, improving the backbone and achieving model lightweighting while enhancing detection speed. Yang et al. [30] proposed a lightweight detection algorithm that combines feature enhancement and attention mechanisms to balance model size and accuracy. They replaced standard convolutions with depthwise separable convolutions (DSConv) to significantly reduce computational complexity. Additionally, they designed and added a Dual Path Attention Gate (DPAG) module and a Feature Enhancement Module (FEM) to improve detection accuracy. Jiang et al. [31] proposed a lightweight forest pest image recognition model based on YOLOv8. This model adopts the Slim Neck design paradigm, replacing conventional convolutions in the neck with lightweight convolutions (GSConv), significantly reducing the computational load. They also introduced the CBAM attention mechanism and improved the loss function to compensate for the accuracy loss caused by lightweighting.

2.4 YOLOv8

The YOLO algorithm has achieved remarkable results in tasks such as object detection, instance segmentation, and image classification. Ultralytics released YOLOv8 [32] on January 10, 2023. Compared to its predecessors YOLOv5 and YOLOv7, YOLOv8 features higher accuracy and faster inference speed, and it is an anchor-free algorithm.

YOLOv8 is composed of a backbone network, neck, and head, as shown in Fig. 1.

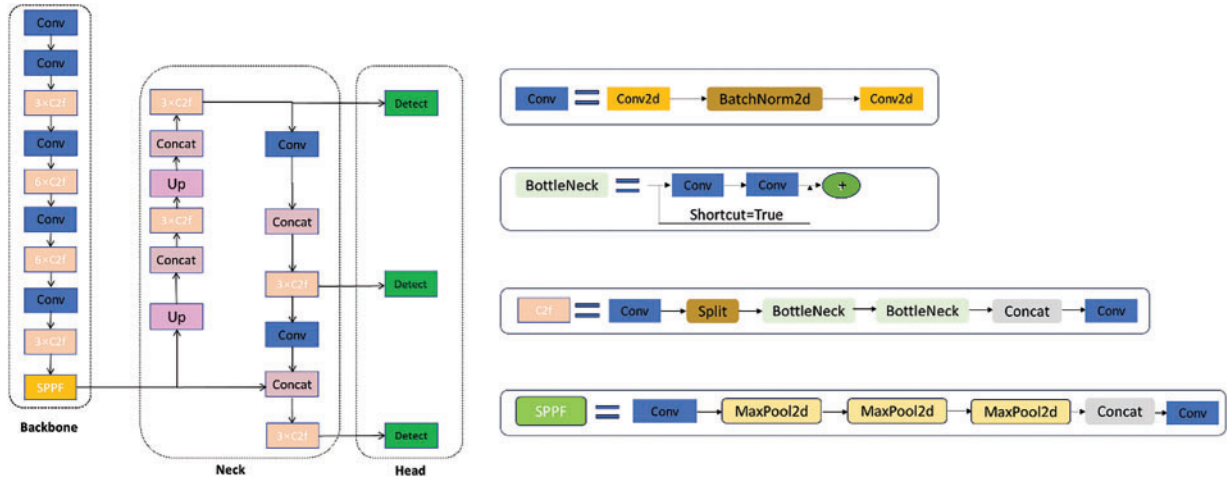

Figure 1: The network structure of YOLOv8

The backbone network consists of convolutional layers, C2f layers, and an SPPF layer [33]. The convolutional layers operate in three steps: first, they process the input information through convolution operations; next, they perform batch normalization; and finally, they apply the SiLU activation function to generate the output [32]. The C2f layer draws inspiration from the ELAN concept in YOLOv7, replacing the original C3 module to obtain richer gradient flow information [34]. As the final layer of the backbone network, the SPPF layer uses three max-pooling operations with a kernel size of 5×5 to aggregate feature maps and then passes the result to the neck layer [35].

In YOLOv8, the primary function of the neck network is to integrate features at different scales [36]. The neck adopts an FPN-PAN structure, combining the advantages of a Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). By introducing the path aggregation network mechanism, it better integrates multi-scale features, improving detection accuracy and robustness. At the same time, the convolution in the upsampling stage is removed, significantly reducing inference time and making the model more streamlined.

YOLOv8 adopts the current mainstream decoupled head structure, where the tasks of object classification and localization are handled separately [37]. The model uses three detection heads to detect large, medium, and small objects, respectively. The detection head for the localization task is evaluated using Bbox Loss, with the loss function comprising two parts: CIoU and DFL (Distribution Focal Loss). For the classification task, the detection head uses the binary cross-entropy (BCE) loss function and is evaluated using Varifocal Loss (VFL).

The network structure of YOLOv8 is shown in Fig. 1.

3 Main Methods

3.1 OD-YOLOv8

To address the issue of YOLOv8's poor performance in detecting small objects, while also speeding up detection and inference, and making the network as lightweight as possible, we propose a novel object detection algorithm, OD-YOLOv8, based on the YOLOv8n baseline model that balances accuracy and speed.

First, we improved the backbone network. YOLOv8's 10-layer backbone network, while focusing on feature extraction, is relatively deep with many layers, leading to a large amount of computation. We

replaced the original backbone with a 5-layer VanillaNet network, which, through its minimalist design, significantly reduces the number of parameters and computational load while maintaining decent accuracy. By eliminating a lot of complex computations, this approach highlights the elegance of minimalism in deep learning.

Next, we replaced the neck with ODNeck using ODConv. ODConv leverages multi-dimensional attention mechanisms and uses a parallel strategy to learn complementary attention along the four dimensions of the convolutional kernel, improving the model's accuracy and robustness. We also employed GSConv, a convolution method combining global and local adaptive attention mechanisms, which enhances the model's adaptability while reducing the number of parameters.

To address the serious problem of missing and false detections of small objects, we added a new detection head, making small objects harder to escape detection. To balance the computational load, we removed one detection head dedicated to large objects. Experimental results show that not only did the detection accuracy for large objects not decrease, but it improved by 6.1%, proving the rationality of this adjustment.

Finally, we introduced Wise-IoU as the loss function, which uses a weighted Intersection over Union (IoU) approach to guide the model to focus more on samples of regular quality, alleviating the harmful gradient issues caused by extreme samples and enhancing the model's generalization capability.

The structure of the OD-YOLOv8 model is shown in Fig. 2.

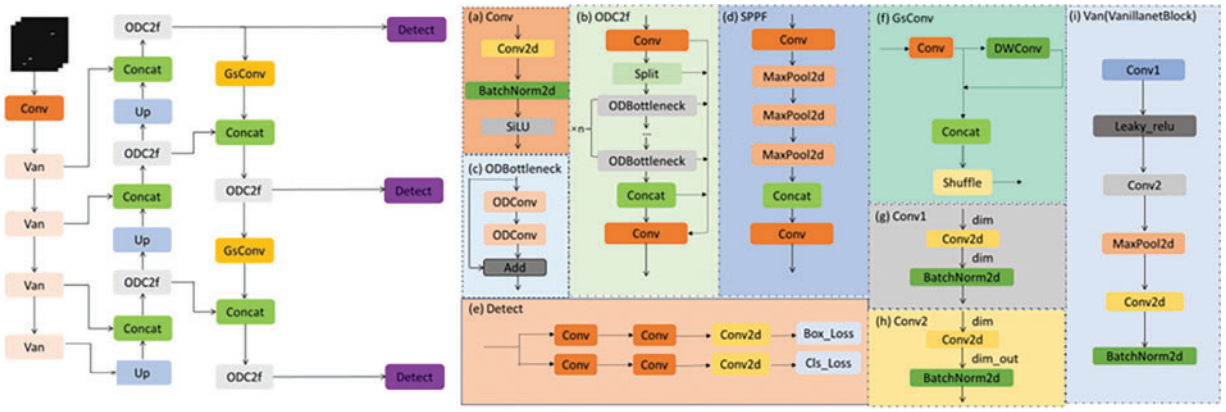

Figure 2: The network structure of OD-YOLOv8

3.1.1 VanillaNet

As neural networks evolve, increasing layers and complexity for accuracy has become common. However, this increases structural complexity, hindering real-time detection. To address this, Huawei Noah's Ark Lab and the University of Sydney introduced VanillaNet, a simple yet powerful architecture [38]. It maintains high accuracy while reducing the model depth and parameters, ideal for resource-constrained environments. VanillaNet balances accuracy and speed, showcasing minimalism in deep learning [38].

VanillaNet-6 is known for its six-layer convolutional design, which effectively extracts and outputs features through downsampling, channel doubling, average pooling, and fully connected layers. Convolution is based on the vanillanetBlock and involves: Conv2d for feature extraction, feature channel normalization,

Leaky ReLU activation function, transformation dimension and maximum pooling, Conv2d and renormalization, etc. These steps simplify the training process. The network structure of VanillaNet-6 is as follows (Fig. 3):

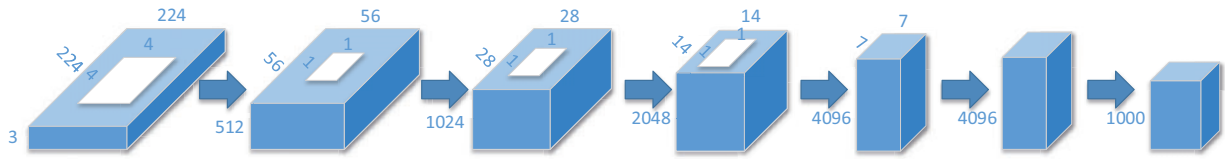

Figure 3: VanillaNet-6 model architecture diagram

Although the original backbone of YOLOv8 has good feature extraction capabilities, its network structure is relatively complex and contains redundancy. Inspired by the concept of VanillaNet, we designed an improved five-layer backbone based on VanillaNet to replace the original backbone of YOLOv8, thereby achieving a lightweight model. Our five-layer backbone network structure is as follows (Fig. 4):

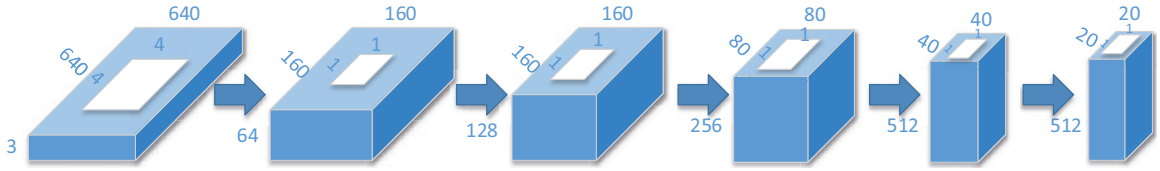

Figure 4: The schematic diagram of the backbone we designed

In the first layer of our backbone, we use a convolution operation with a stride and kernel size of 4 to adjust the number of channels to 64, while reducing the size of the output feature map to one-fourth of the original. The second layer has a convolution stride of 1, which does not change the size of the feature map but doubles the number of channels to 128 and merges the output with the neck. The operations of the third and fourth layers are similar, continuing to double the number of channels while halving the size of the output feature map by setting the convolution stride to 2, and merging the output with the neck. The fifth layer does not further increase the number of channels, and other operations are similar to the second and third layers. The convolution kernel size for the second to fifth layers is set to 1, aiming to retain as much information of the feature map as possible with minimal computational cost.

To enhance integration with the neck part, reduce feature loss, and improve detection accuracy, we have implemented the following design: The output from the last vanillanetBlock is upsampled and fused with the output from the third vanillanetBlock. After another ODC2f operation and upsampling, this fused result is combined with the output from the second vanillanetBlock. The same steps are repeated until fusion with the output from the first vanillanetBlock. The results of these three fusions, after undergoing an ODC2f operation, serve as inputs for the large, medium, and small object detection heads, respectively.

This design aims to fully utilize feature information from different layers and improve the model's ability to detect objects of various sizes through gradual fusion and ODC2f operations. Experimental results show that this backbone design is well-compatible with the newly added small object detection head, achieving good results while also making the model lightweight.

3.1.2 ODC2f

Currently, most neural networks use static convolution kernels as a general training paradigm. However, when processing different input information, this convolution method results in fixed convolution kernels for each filter, which can easily lead to information loss. Recent research on dynamic convolution has found that using a linear combination of multiple convolution kernels as the learning target, and dynamically weighting them through an attention mechanism, makes the convolution operation dependent on the input [39]. This approach can enhance the fusion of contextual information, thereby significantly improving detection accuracy. The operation of dynamic convolution is defined as follows:

$$y = \left(\alpha_{w1} w1 + \ldots + \alpha_{wi} wi + \ldots + \alpha_{wn} wn\right) * x \tag{1}$$

here, wi represents the i-th set of convolution filters, and α_{wi} is the weighting coefficient for wi. α_{wi} is calculated by an attention function that depends on the input features.

Traditional dynamic convolution methods such as CondConv [40] and DyConv [41] only focus on the number of convolution kernels, while ODConv, as a multidimensional attention mechanism, learns complementary attention in four dimensions: kernel size, input channel number, output channel number, and convolution kernel number, enhancing feature adaptability and information fusion, improving efficiency and generalization ability.

The network structure of ODConv is shown in Fig. 5.

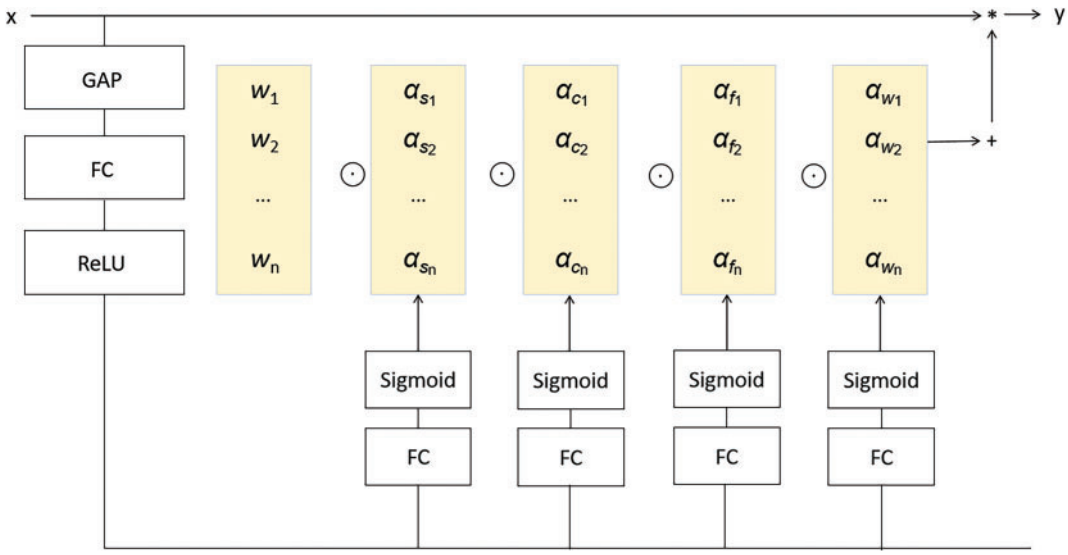

Figure 5: The structure of ODConv

The output y of ODConv can be expressed using the following formula:

$$y = \left(\alpha_{w1} \otimes \alpha_{f1} \otimes \alpha_{c1} \otimes \alpha_{s1} \otimes W_1 + \ldots + \alpha_{wi} \otimes \alpha_{fi} \otimes \alpha_{ci} \otimes \alpha_{si} \otimes W_1 + \ldots + \alpha_{wn} \otimes \alpha_{fn} \otimes \alpha_{cn} \otimes \alpha_{sn} \otimes W_n\right) \tag{2}$$

where α_{wi} represents the i-th group of convolution filters, α_{fi} represents the learnable weight for the output dimension, α_{ci} represents the learnable weight for the input dimension, and α_{si} represents the learnable

weight for the spatial dimension. ⊗ denotes the weighted operation across different dimensions of the convolution filters.

In highly complex environments and adverse weather conditions, such as storms, typhoons, waves, and ocean currents, the accuracy and reliability of detection are often significantly affected. ODConv, as a dynamic multi-dimensional convolution method, can effectively extract target features by dynamically adjusting the direction and weight of convolutional kernels, thereby enhancing the distinction between targets and backgrounds. This is particularly beneficial for handling SAR images with high noise levels. Given that ships have varying positions and postures in the ocean, ODConv's multi-directional characteristics allow it to adapt to different target directions and shape changes, capturing both local and global information of the targets more effectively, thus improving detection accuracy. Moreover, in adverse weather conditions like strong winds and heavy rain, ODConv can dynamically adjust convolution operations to effectively suppress the impact of noise on detection results, thereby enhancing detection robustness.

Therefore, to further improve the performance of the target detection model, we decided to improve the bottleneck part of the neck C2f module. Specifically, we replaced all the conventional convolution operations in this part with ODConv. Through this replacement, we successfully constructed a new feature extraction module named ODC2f. This improvement has brought significant results: the accuracy and efficiency of feature processing have been greatly improved, and the generalization ability of the network has also been significantly enhanced. This is particularly critical for detecting small objects, as it provides us with considerable performance gains. The structure of ODC2f and ODBottleneck is shown in Fig. 6.

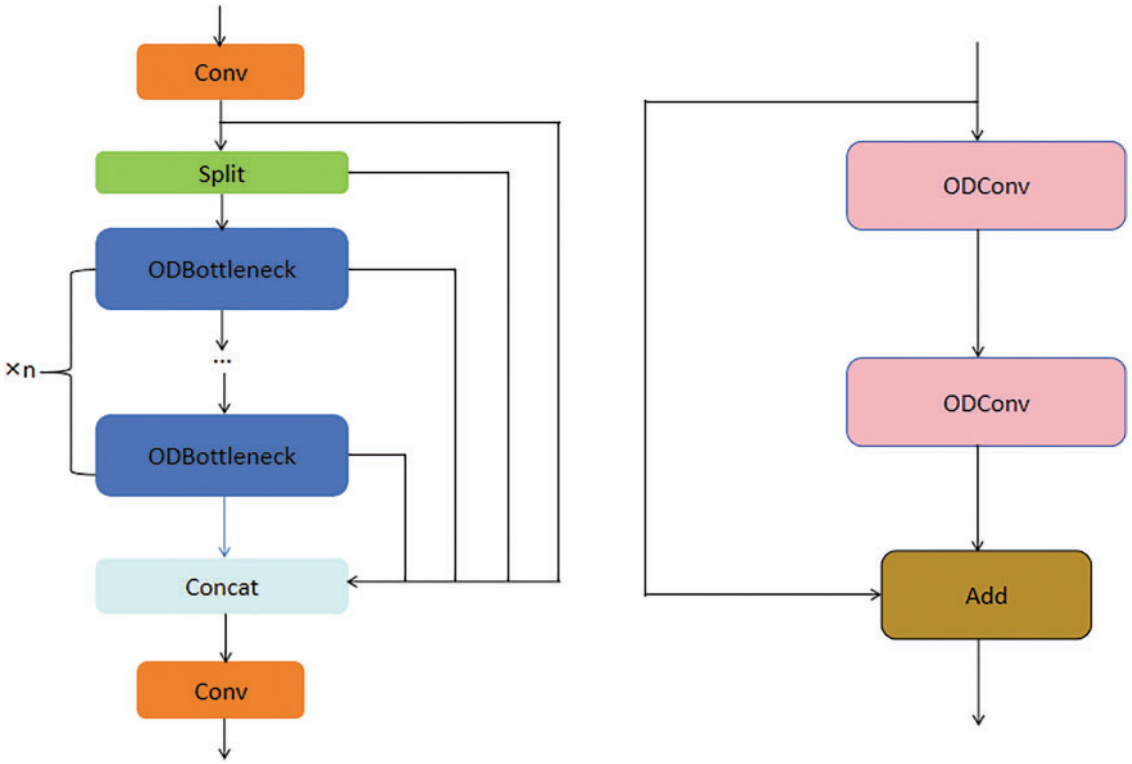

Figure 6: ODC2f (left) and ODBottleneck (right)

3.1.3 GSConv

GSConv was first applied in the field of autonomous driving [42]. Autonomous driving requires strict precision and speed, and traditional lightweight networks like MobileNets [43–45] and ShuffleNets [46,47], significantly reducing the number of model parameters, sacrificing a considerable amount of accuracy. Against this backdrop, GSConv emerged, aiming to balance model accuracy and speed.

GSConv first downsamples the input information through a standard convolution, followed by the use of depthwise separable convolution (DW separable convolution). Then, the results of the two convolutions (SC and DSC) are concatenated. Finally, a shuffle operation is performed to rearrange the corresponding channels of the first two convolutions to be adjacent. The details of GSConv are illustrated as follows (Fig. 7):

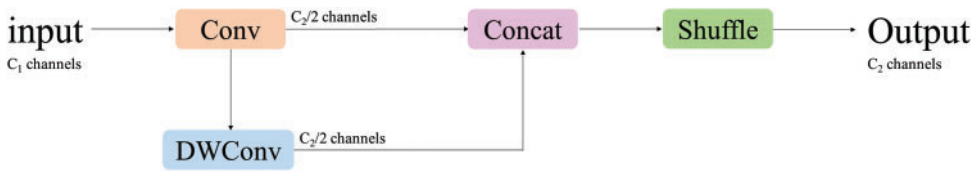

Figure 7: GSConv

If GSConv is used at all stages of the network, it may lead to an excessively deep network, thereby increasing data flow resistance and significantly increasing inference time. Therefore, we choose to replace only the two standard convolutions in the neck downsampling process with GSConv. This helps to better learn features and enhance the correlation between local features. By doing so, we achieve a lightweight model while maintaining accuracy.

3.1.4 Improvements in Detection Head

Before the improvement, YOLOv8 had three detection heads corresponding to feature map sizes of 80 × 80, 40 × 40, and 20 × 20. The 80 × 80 feature map was used for detecting small objects larger than 8 × 8, the 40 × 40 feature map for medium objects larger than 16 × 16, and the 20 × 20 feature map for large objects larger than 32 × 32.

However, due to the large downsampling factor in YOLOv8, deeper feature maps struggle to capture information about small objects, leading to deficiencies in small object detection and increased occurrences of missed detections and false positives. To address this, in the FPN module of the neck, after the second upsampling step that generates the 80 × 80 feature map, we performed an additional upsampling operation. We then fused the result with the second Van layer in the backbone network to generate a large-scale 160 × 160 feature map. This was output to the head, where we added a 160 × 160 detection head to enhance the detection of small objects.

To balance the computational load introduced by the new small object detection head, we removed one of the original detection heads used for large objects (the one corresponding to the 20 × 20 feature map), thus eliminating redundant computations.

The schematic diagram of our improved detection head is shown in Fig. 8.

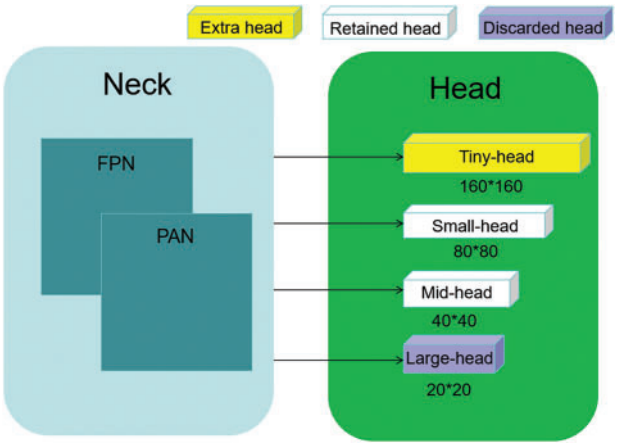

Figure 8: Improvements in detection head

3.1.5 Wise-IoU

In the traditional YOLOv8 detection algorithm, CIoU (Complete Intersection over Union) is commonly used to evaluate the accuracy of bounding boxes. CIoU provides a comprehensive evaluation by considering the overlap of boxes, the distance between their center points, and the aspect ratio. However, the training data may contain low-quality samples, and overemphasis on these samples can affect detection performance. Furthermore, CIoU is overly sensitive to output parameters and object proportions. To address these issues, we have introduced the Wise-IoU loss function, which uses "outlierness" to evaluate anchor boxes. This approach reduces the interference of extreme samples and allows the model to focus more on ordinary samples, thereby improving detection performance [48].

The definition of the overall network loss is as follows:

$$L = W_{box}L_{box} + W_{cls}L_{cls} + W_{obj}L_{obj} \tag{3}$$

$$L_{box} = L_{WIoU} \tag{4}$$

where L_{box}, L_{cls} and L_{obj} represent the bounding box loss, classification loss, and confidence loss, respectively. W_{box}, W_{cls} and W_{obj} denote the weights for the corresponding types of loss, respectively. The total loss L is obtained by computing the weighted sum of the above three losses. Additionally, L_{box} here specifically refers to L_{WIoU}.

The bounding box regression model is shown in Fig. 9. In this model, the predicted bounding box is represented as $C_p(x, y, w, h)$, and the ground truth bounding box is represented as $C_{gt}(x_{gt}, y_{gt}, w_{gt}, h_{gt})$. The relevant calculation formula for L_{WIoU} is as follows:

$$L_{WIoU} = rL_{IoU}R_{WIoU} \tag{5}$$

$$L_{IoU} = 1 - IoU = 1 - \frac{w_i h_i}{wh + w_{gt}h_{gt} - w_i h_i} \in [0,1] \tag{6}$$

$$R_{WIoU} = exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(w^2{}_R + h^2{}_R)^*}\right) \in [1, e) \tag{7}$$

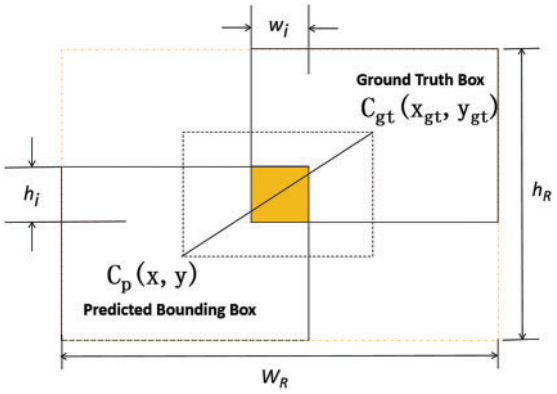

Figure 9: Bounding box regression model

The IoU metric is used to measure the degree of overlap between the anchor box and the target box. R_{WIoU} improves the L_{IoU} of ordinary anchor boxes, while L_{IoU} reduces the R_{WIoU} of high-quality anchor boxes, and reduces the focus on the distance between the center points when the overlap is good. w_R and h_R are the length and width of the minimum bounding box. To avoid the impact of R_{WIoU} on gradient convergence, ${w^2}_R$ and ${h^2}_R$ are separated from the calculation and marked with *. The formula for calculating r is as follows:

$$r = \frac{\beta}{\partial \alpha^{\beta-\partial}} \tag{8}$$

$$\beta = \frac{L^*_{IoU}}{L_{IoU}} \in [0, +\infty) \tag{9}$$

where r represents the non-monotonic focusing factor, we introduce β to describe the degree of outlierness of the predicted bounding box quality. In the formula, L^*_{IoU} and L_{IoU} respectively denote the monotonic focusing factor and the moving average of momentum m. Both ∂ and β are hyperparameters that we set. Through experiments, we found that replacing the loss function with Wise-IoU significantly enhances the robustness of our model, greatly improving the performance of ship detection.

4 Experiment

In this section, the performance validation experiments for the proposed SAR ship detection algorithm will be conducted. First, we will introduce the experimental environment and hardware equipment. Next, we will describe the two SAR image datasets used—HRSID [49] and SSDD [50]. Then, we will verify the effectiveness of the employed modules through ablation experiments. Finally, we will compare our algorithm with other existing object detection algorithms to validate its effectiveness.

To address the issue of model overfitting, we implemented a series of measures. We applied data augmentation techniques by adjusting the color, hue, saturation, and brightness of images to generate more training samples, increasing the diversity of the data and allowing the model to perform well under various lighting conditions. Additionally, we performed geometric transformations on the images, including translation, scaling, and horizontal flipping, altering the spatial position and size of the images. This enabled the model to learn how objects appear in different positions and scales, thereby enhancing its generalization ability. We also employed mosaic augmentation, which involves combining multiple images into one for training, resulting in a single image containing multiple different scenes and objects, greatly enriching the diversity of the training samples. Furthermore, we introduced weight decay, adding a penalty term related

to the weights in the loss function to prevent the weights from becoming too large. This helps control the complexity of the model and avoids overfitting to noise in the training data.

4.1 Experimental Environment and Parameter Settings

Our experiments were conducted in the following environment: The hardware setup includes an NVIDIA RTX 3080 Ti GPU (12 GB memory) and a 12-core Intel Xeon(R) Silver 4214R CPU, running on the Ubuntu 20.04 operating system. The experiments utilized the mm detection [51] framework, developed with PyTorch 2.0.0 and Python 3.8. CUDA 11.8 was used for GPU acceleration. During training, we employed the AdamW optimizer for a total of 150 epochs, with a batch size of 16. The initial and final learning rates were both set to 0.01. The software and hardware parameter settings for the experiment are shown in Table 1.

Table 1: Software and hardware parameter settings for the experiment

Platform	Configuration
Memory	90 G
CPU	Intel Xeon Silver 4214R
GPU	NVIDIA RTX 3080 Ti
GPU accelerator	CUDA 11.8
Programming language	Python 3.8
Development environment	PyTorch 2.0.0

4.2 Introduction to the Dataset

In this paper, we evaluate the OD-YOLOv8 network using the HRSID [49] and SSDD [50] datasets. Below is a brief introduction to these two datasets.

HRSID is a high-resolution, large-scale SAR ship dataset released in 2020. This dataset includes 5604 images, each sized 800 × 800, acquired from Sentinel-1 and TerraSAR-X. It contains a total of 16,951 ship targets and is widely used for tasks such as instance segmentation, semantic segmentation, and ship detection. The dataset consists of large, medium, and small ships, which account for 54.5%, 43.5%, and 2% of the total dataset, respectively. On average, each image contains three ships. Compared to SSDD, the SAR images in the HRSID dataset have a higher resolution (below 3 m), with ship features being more accurate and detailed. We divided the dataset into training, validation, and test sets in a 7:1:2 ratio, containing 3922, 561, and 1121 images, respectively.

SSDD, constructed by Li et al., is the first published and publicly available dataset for ship detection in SAR images. The images in this dataset are sourced from RadarSat-2, Sentinel-1, and TerraSAR-X sensors, comprising a total of 1160 images with 2456 ships. The ship sizes range from 7 × 7 to 211 × 298, with an average of 2.11 ships per image. The dataset includes various sea conditions (favorable and unfavorable), polarization modes (HH, HV, VV, and VH), resolutions (ranging from 1 to 15 m), and offshore scenes (complex and simple). We divided the dataset into training and test sets in an 8:2 ratio, containing 928 and 232 images, respectively.

Comparison of parameters between HRSID and SSDD datasets is shown in Table 2, and examples of images from the HRSID and SSDD datasets are shown in Fig. 10.

Table 2: Comparison of parameters between HRSID and SSDD datasets

Dataset categories	HRSID	SSDD
Data source	TerraSAR-X, Sentinel-1	TerraSAR-X, RadarSat-2, Sentinel-1
Resolution	0.5–3	1–15
Image size	800 × 800	500 × 500
Polarization mode	HH, VV, HV	HH, HV, VV, VH
Number of vessels	16,951	2456
Training set size	3922	928
Test set size	1121	232

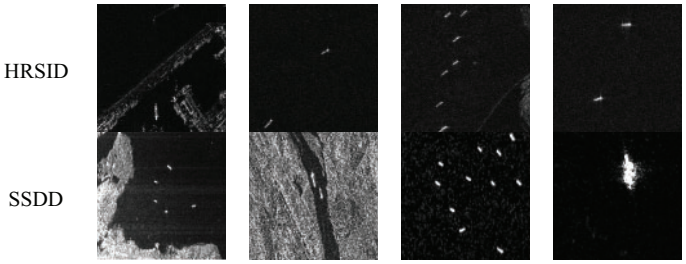

Figure 10: HRSID and SSDD dataset examples

4.3 Evaluation Metrics

We use Average Precision (AP) as our primary evaluation metric and adopt the COCO evaluation metrics. The calculation method for mAP50–95 involves computing 10 mAP values at IOU thresholds ranging from 50% to 95% (in 5% increments) and then averaging these 10 values. AP_{50} and AP_{75} are the AP values at IOU thresholds of 0.5 and 0.75, respectively, which measure the accuracy of the model in detecting objects. AP_L, AP_M, and AP_S represent the AP values for large, medium, and small object detection, respectively. As common performance evaluation metrics, *TP* (True Positives), *FP* (False Positives), *TN* (True Negatives), and *FN* (False Negatives) represent correctly classified positive samples, incorrectly classified positive samples, correctly classified negative samples, and incorrectly classified negative samples, respectively. The calculation formulas for Recall (*R*), Precision (*P*), and mean Average Precision (mAP) are as follows:

$$R = \frac{TP}{TP + FN} \tag{10}$$

$$P = \frac{TP}{TP + FP} \tag{11}$$

$$\text{mAP} = \int_0^1 P(R)\, dR \tag{12}$$

4.4 Ablation Experiments

To validate the effectiveness of each proposed strategy, we conducted ablation experiments on the HRSID dataset. The experimental results show that improving the detection head, optimizing the Wise-IoU loss function, and introducing the ODC2f module significantly enhance detection accuracy. Although replacing the Van backbone network slightly sacrifices some accuracy, it significantly reduces the number of parameters and improves the model's detection and inference speed. Moreover, while maintaining other accuracies, the use of GSConv significantly boosts the detection capability for large ships by 7.8% and reduces

the computational load and the number of floating-point operations. The experimental results are shown in Table 3.

Table 3: Ablation experiments

Method	AP	AP50	AP75	APS	APM	APL	Params	GFlops
YOLOv8 (Baseline)	65.5	89.8	75.7	53.1	77.8	47.5	3,011,043	8.2
Replace the head	68.1	91.4	79.3	58.6	78.5	38.7	2,070,595	11.9
Replace the head+Wise-IoU	69	92.8	80.7	59.6	78.4	37	2,070,595	11.9
Replace the head+Wise-IoU+Van	67.6	91.8	78.4	57.6	77.5	38.7	836,866	9.2
Replace the head+Wise-IoU+Van+GSConv	67.8	91.7	78.3	57.3	77.9	46.5	793,219	9
Replace the head+Wise-IoU+Van+GSConv+ODC2f (Our model)	69	92.5	79.9	59.1	78.5	53.6	816,189	7.8

From Table 3, it can be observed that after improving the detection head, the model's AP, AP_{50}, and AP_{75} have increased by 2.6%, 1.6%, and 1.6%, respectively, compared to the baseline model. The newly added small target detection head better captures the feature information of small targets, enhances the sensitivity to small targets, and significantly improves the AP_S by 5.5%. Removing the large target detection head greatly reduces the number of parameters, by about 31.2%. Our improved loss function Wise-IoU alleviates the harmful gradient caused by extreme samples and focuses more on the quality of regular samples, which significantly improves the detection accuracy of the model, with AP, AP_{50}, and AP_{75} increasing by 0.9%, 1.4%, and 1.4%, respectively, and AP_S also increasing by 1%, showing significant results. Although the replacement of the backbone network caused a slight decrease in detection accuracy, with AP, AP_{50}, and AP_{75} decreasing by 1.4%, 1%, and 2.3%, respectively, AP_L increased by 1.7%. The replacement of VanillaNet abandoned complex operations and shortcuts, resulting in a significant reduction in the number of parameters and FLOPs. The number of parameters decreased by approximately 59.6%, and FLOPs decreased by 22.7%, showing a significant lightweight effect. By replacing with GSConv, the model weight is further reduced, with the number of parameters and FLOPs reduced by about 5.2% and 2.2%, respectively. This helps to better learn features and enhance the correlation between local features while achieving lightweighting, resulting in a significant 7.8% improvement in AP_L. This perfectly compensates for the impact of removing large target detection heads on the accuracy of large target detection, achieving a win-win situation of lightweighting and performance improvement. The ODC2f module significantly improves the accuracy and efficiency of feature processing without increasing model complexity by introducing ODConv dynamic multi-dimensional convolution, enhancing the model's feature adaptability and generalization ability. After adding the ODC2f module, the accuracy of the model has been significantly improved, with AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L increasing by 1.2%, 0.8%, 1.6%, 1.8%, 0.6%, and 7.1%, respectively. After combining all the above modules, our model has achieved significant improvements compared to the baseline model YOLOv8, with each metric improving by 2.5%, 2.7%, 4.2%, 6%, 0.7%, and 8.1%, while reducing the number of parameters and FLOPs by 72.9% and 4.9%, respectively. This verifies the effectiveness of each module and achieves a balance between model lightweighting and accuracy.

4.5 Ablation Experiments

We conducted extensive comparative experiments on the HRSID and SSDD datasets to validate the effectiveness of our proposed method. After comparing with eight state-of-the-art object detection algorithms, we found that our model not only improves accuracy but also achieves significant lightweight performance, thereby confirming the effectiveness and advancement of our model. To further test the generalization ability of the model and its detection performance in complex scenarios, we conducted comparative experiments on nearshore and offshore ships using the SSDD dataset and achieved ideal results, thus verifying the robustness and reliability of our model.

4.5.1 Comparative Experiments Conducted on the HRSID Dataset

To verify the effectiveness of our designed model, we conducted experiments on the HRSID dataset using 8 common object detection algorithms, including DINO (2022), EfficientNet (2019), TOOD (2021), LD (2022), DyHead (2021), FCOS (2019), DDOD (2021), VFNet (2020), and YOLOv8 (2023). We performed a comparative evaluation of the detection metrics for each model, and the results are shown in Table 4 (The bolded data represents the best-performing data in the comparison model).

Table 4: Comparison and evaluation of detection metrics for various models on HRSID

Method	AP	AP50	AP75	APS	APM	APL	Params (M)	GFlops
DINO (2022)	0.588	0.864	0.696	0.546	0.651	**0.617**	47.54	179
EfficientNet (2019)	0.437	0.708	0.489	0.272	0.628	0.033	18.339	106
TOOD (2021)	0.645	0.886	0.731	0.513	0.78	0.454	32.018	123
LD (2022)	0.414	0.644	0.465	0.236	0.609	0.035	19.239	96.86
DyHead (2021)	0.64	0.875	0.722	0.495	0.783	0.527	38.89	68.052
FCOS (2019)	0.518	0.766	0.59	0.338	0.701	0.247	32.113	123
DDOD (2021)	0.65	0.886	0.737	0.511	0.785	0.513	32.196	111
VFNet (2020)	0.624	0.853	0.701	0.472	0.777	0.471	32.709	118
YOLOv8 (2023)	0.655	0.898	0.757	0.531	0.778	0.475	3.011	8.2
OD-YOLOv8 (ours)	**0.69**	**0.925**	**0.799**	**0.591**	**0.785**	0.536	**0.816**	**7.8**

We have selected eight object detection algorithms that are among the most popular and advanced in the past five years. However, each of these algorithms has its shortcomings. DINO, with its Transformer architecture, excels at capturing global information and shows significant advantages in small object detection, with the AP_S metric notably ahead of most models. However, its performance in complex weather conditions is average, and it relies on pre-training data and data augmentation strategies. Although the Transformer architecture provides some robustness against background interference, its large number of parameters (Params) and high computational demand (GFlops) make the model complex and bulky, which may be a limitation in real-time detection and resource-constrained environments. As a convolutional neural network for image classification, EfficientNet achieves significant simplification by balancing model complexity and performance through adjusting depth, width, and resolution. However, this simplification comes at the cost of reduced accuracy, with only 27.2% and 3.3% on AP_S and AP_L, respectively. Its performance is not ideal in high background interference and complex weather conditions, and its effectiveness for small object detection is limited. As a one-stage object detector, TOOD enhances detection performance through task alignment strategies, which is particularly effective for small object detection. The task alignment strategy also helps reduce the impact of background noise, showing excellent performance under background

interference. However, the model's parameter count is 32.018 M, which is still relatively large and has significant room for lightweight improvement. LD improves classification and localization accuracy through a label decoupling mechanism. Its advantage lies in having a lower parameter count compared to most models, resulting in faster inference and training speeds, but it performs poorly in small and large object detection. DyHead enhances detection performance by dynamically adjusting network parameters, showing good results in large object detection. However, the dynamic adjustment mechanism increases the model's computational complexity and memory usage, which can be a problem in resource-limited environments. As an anchor-free object detection method, FCOS does not require predefined anchor boxes, simplifying model design and implementation. However, since every pixel participates in prediction, it is prone to background false detections. DDOD is a dynamic dense object detection method that can dynamically adjust detector parameters based on the features of the input image, making it more adaptable to different scenarios. Through dense feature sampling, DDOD performs well in small object detection, but its dense feature extraction and dynamic adjustments may still require high computational resources, especially in high-resolution images or real-time applications, making the model relatively large. With its zoom mechanism, VFNet can locate objects more accurately and performs exceptionally well in complex backgrounds, exhibiting strong robustness. However, the zoom mechanism and adaptive weight allocation may increase computational overhead, particularly in high-resolution images or real-time applications, requiring high computational resources. Compared to these popular algorithms of the past five years, our algorithm not only significantly improves accuracy but also achieves model lightweighting, resulting in impressive outcomes.

4.5.2 Comparison Experiments Conducted on the SSDD Dataset

To further validate the effectiveness of our model, we conducted extensive comparative experiments on the SSDD dataset. Similar to our approach to HRSID, we compared our method with eight state-of-the-art object detection algorithms. The experimental results are shown in Table 5 (The bolded data represents the best-performing. Data in the comparison model).

Table 5: Comparison and evaluation of detection metrics of various models on the SSDD

Method	AP	AP50	AP75	APS	APM	APL
DINO (2022)	0.588	0.864	0.696	0.546	0.651	0.617
EfficientNet (2019)	0.549	0.894	0.612	0.552	0.596	0.378
TOOD (2021)	0.62	0.939	0.718	**0.62**	0.645	0.367
LD (2022)	0.53	0.875	0.597	0.533	0.537	0.378
DyHead (2021)	0.61	0.952	0.715	0.594	0.656	0.572
FCOS (2019)	0.477	0.824	0.529	0.532	0.385	0.245
DDOD (2021)	0.626	0.945	0.742	0.618	0.657	0.506
VFNet (2020)	0.561	0.896	0.637	0.565	0.59	0.424
YOLOv8 (2023)	0.629	0.98	0.727	0.577	0.726	0.65
OD-YOLOv8 (ours)	**0.646**	**0.984**	**0.749**	0.597	**0.733**	**0.654**

Our designed model achieved AP, AP_{50}, and AP_{75} scores of 64.6%, 98.4%, and 74.9%, respectively, on the SSDD dataset, representing improvements of 1.7%, 0.4%, and 2.2% over the baseline model. Additionally, there were varying degrees of improvement in AP_S, AP_M, and AP_L metrics, with a significant 2-point increase in AP_M. Compared to eight other object detection algorithms, our model performed best across most metrics, except for small vessel detection, where it was outperformed by TOOD. Notably, for large vessel detection,

our model's AP_L reached 65.4%, significantly surpassing other methods. However, its lower performance in small vessel detection might be due to the cost of lightweighting the backbone network, and the relatively small size of the SSDD dataset, which may lack sufficient representativeness. In the future, we plan to introduce more efficient attention mechanisms and loss functions to enhance the model's ability to detect small targets. Overall, the experimental results demonstrate that our model performs exceptionally well on the SSDD dataset, fully highlighting its effectiveness and robustness.

4.5.3 Comparison Experiments in Nearshore and Offshore Scenarios (under the SSDD Dataset)

To further validate the generalization and detection capabilities of our model in complex scenarios, we conducted comparative experiments between our model and eight different object detection algorithms in both nearshore and offshore environments. Nearshore detection is easily influenced by the presence of buildings and water reflections, leading to strong background noise that makes it difficult to distinguish ship targets from the background. Additionally, nearshore areas often have shallow waters, where radar signal scattering can be affected by underwater terrain and water quality variations, thereby impacting detection performance. Consequently, the experiments demonstrate that both our model and traditional object detection algorithms perform significantly worse in nearshore detection compared to offshore detection. The experimental results indicate that our method is superior to others. The results for nearshore and offshore scenarios are shown in Tables 6 and 7, respectively (The bolded data represents the best-performing data in the comparison model).

Table 6: Comparative evaluation of detection metrics for various models in nearshore scenarios of SSDD

Method	AP	AP50	AP75	APS	APM	APL
DINO (2022)	0.398	0.683	0.395	0.357	0.474	0.441
EfficientNet (2019)	0.357	0.695	0.345	0.343	0.472	0.254
TOOD (2021)	0.409	0.789	0.358	0.389	0.484	0.33
LD (2022)	0.322	0.633	0.276	0.316	0.357	0.304
DyHead (2021)	0.462	0.852	0.431	0.424	0.53	0.513
FCOS (2019)	0.237	0.498	0.201	0.29	0.199	0.104
DDOD (2021)	0.442	0.82	0.434	0.403	0.538	0.376
VFNet (2020)	0.339	0.707	0.279	0.34	0.394	0.193
YOLOv8 (2023)	0.531	0.918	0.548	0.464	0.636	0.534
OD-YOLOv8 (ours)	**0.566**	**0.945**	**0.648**	**0.506**	**0.661**	**0.654**

Table 7: Comparative evaluation of detection metrics for various models in offshore scenarios of SSDD

Method	AP	AP50	AP75	APS	APM	APL
DINO (2022)	0.583	0.935	0.657	0.501	0.692	0.73
EfficientNet (2019)	0.598	0.969	0.687	0.572	0.665	0.577
TOOD (2021)	0.609	0.977	0.678	0.58	0.704	0.441
LD (2022)	0.559	0.969	0.612	0.544	0.617	0.458
DyHead (2021)	0.604	0.988	0.674	0.572	0.703	0.643
FCOS (2019)	0.523	0.948	0.525	0.548	0.479	0.379
DDOD (2021)	0.606	0.986	0.681	0.571	0.69	0.631

(Continued)

Table 7 (continued)

Method	AP	AP50	AP75	APS	APM	APL
VFNet (2020)	0.568	0.972	0.597	0.539	0.674	0.635
YOLOv8 (2023)	**0.652**	0.986	0.741	0.584	**0.755**	0.737
OD-YOLOv8 (ours)	0.648	**0.988**	**0.778**	**0.593**	0.726	**0.781**

4.5.4 Visual Analysis

To visually demonstrate the effectiveness of our object detection approach, we conducted a visual analysis of the HRSID dataset. By comparing the visual results with nine common detection algorithms, the findings further validate the effectiveness of our method. In the figure, red boxes represent ground truth, green boxes indicate correctly detected objects, and yellow and blue boxes denote false positives (FP) and false negatives (FN), respectively. The detection results are shown in Fig. 11.

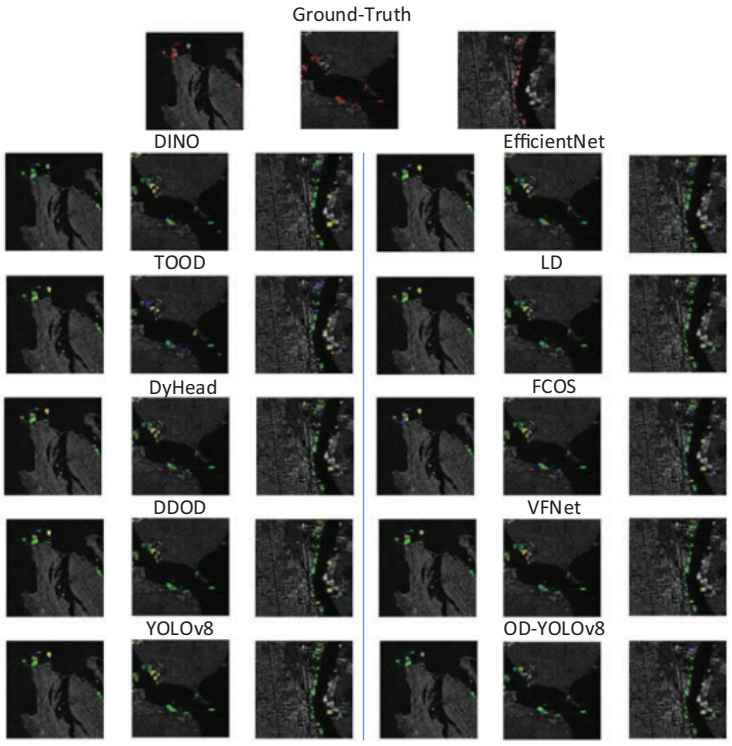

Figure 11: Visualization results

Through experimentation, we have found that in most scenarios, our improved model exhibits overall improvements in both missed detections and false detections when compared to eight common object detection algorithms and the original YOLOv8 model. Notably, improvement is particularly significant when detecting small ships. However, in some cases involving dense clusters of ships and complex background interference, its performance may sometimes be inferior to other models. Nevertheless, in most scenarios, the number of missed and false detections has decreased.

5 Conclusion

To address the issues of difficulty in detecting small vessels, severe background interference, and limited hardware resources in current SAR ship detection, most existing models generally focus solely on improving accuracy or lightweighting, making it challenging to balance resource utilization and detection performance. To practically improve the accuracy of ship detection and achieve real-time detection under the constraints of edge device deployment, we have designed and proposed a new model based on YOLOv8, named OD-YOLOv8.

Specifically, to address the serious issues of missed and false detections of small targets, we added 160 × 160 small target detection head and removed the 20 × 20 large target detection head. This approach eliminates redundant computations while enhancing the model's sensitivity to small targets. To reduce resource consumption and achieve model lightweighting, we replaced the backbone network with Huawei's latest research result—VanillaNet. We also replaced the loss function with Wise-IoU based on a dynamic non-monotonic mechanism, making the model more focused on the quality of ordinary samples and reducing the impact of extreme sample data on the model, thereby significantly improving the model's robustness and generalization ability. Additionally, we replaced the convolution in the neck downsampling stage with GSConv, enhancing the model's expressive capability while reducing computational load. Finally, we utilized ODConv—a multi-dimensional dynamic convolution—to significantly enhance context information fusion by addressing four dimensions: spatial kernel size, input channels, output channels, and the number of convolution kernels, thereby markedly improving the convolution's feature extraction capability. We evaluated the model on the HRSID dataset, and the results showed that AP and AP_{50} improved by 3.5% and 2.7%, respectively, while the number of parameters and GFLOPS decreased by 72.9% and 4.9%, respectively. The model also performed excellently on the SSDD dataset, with AP and AP_{50} improving by 1.7% and 0.4%, respectively. We further compared this method with other mainstream object detection algorithms, validating the effectiveness of our designed model.

Compared to YOLOv8n, we've achieved a balance between detection accuracy and computational efficiency. By replacing the detection head, we've reduced parameters by 31.2% while boosting AP_{50} by 1.6%. Using Wise-IoU, we've increased AP50 by 1.4% without sacrificing efficiency. Although replacing the backbone network resulted in a 1% precision loss, it significantly reduced parameters and GFlops by 59.6% and 22.7%, respectively. Swapping GSConv in the neck led to a minimal 0.1% precision drop, with parameter and GFlops reductions of 5.5% and 2.2%. Introducing the ODC2f module decreased GFlops by 13.3% and improved precision by 0.8%. Our research is highly beneficial for edge device deployment.

Despite our model having achieved remarkable results in lightweight design and demonstrated good performance in detection accuracy, there is still room for further optimization. Looking ahead, we intend to continue focusing on resource consumption while exploring in-depth from multiple dimensions. Firstly, we plan to introduce more efficient attention mechanisms and loss functions to enhance the model's ability to detect small targets and improve overall detection accuracy. Secondly, we will strive to expand the applicability of the model, enabling it to better adapt to complex marine environments and accommodate more diversified datasets. In this process, we will also integrate multimodal data and adaptive learning techniques to enhance the model's generalization ability and robustness. Lastly, we will actively explore the potential of the model in practical industrial deployments, especially its performance in real-time and resource-constrained environments, aiming to provide more reliable solutions for practical applications. These improvements are not only expected to further enhance detection performance but also potentially drive technological innovations in the field of SAR ship target detection, expanding its application prospects in other related fields, and ultimately bringing broader application value and far-reaching impacts to this field.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Open Research Fund Program of State Key Laboratory of Maritime Technology and Safety in 2024. This research also received partial funding from the National Natural Science Foundation of China (Grant No. 52331012) and the Natural Science Foundation of Shanghai (Grant No. 21ZR1426500).

Author Contributions: Conceptualization: Zhuowei Wang; methodology: Zhuowei Wang, Dezhi Han; software: Zhuowei Wang, Bing Han; validation: Zhuowei Wang, Dezhi Han and Zhongdai Wu; formal analysis: Zhuowei Wang, Zhongdai Wu; investigation: Zhuowei Wang; resources: Zhuowei Wang; data curation: Zhuowei Wang; writing and original draft preparation: Zhuowei Wang, Dezhi Han, Bing Han and Zhongdai Wu; writing—review and editing: Zhuowei Wang, Dezhi Han, Bing Han and Zhongdai Wu; visualization: Zhuowei Wang; supervision: Dezhi Han, Bing Han and Zhongdai Wu; project administration: Dezhi Han; funding acquisition: Dezhi Han, Bing Han and Zhongdai Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data presented in this study are available on request from the corresponding author.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Sun C, Li X, Zou W, Wang S, Wang Z. Chinese marine economy development: dynamic evolution and spatial difference. Chin Geogr Sci. 2018;28(1):111–26. doi:10.1007/s11769-017-0912-8.
2. Jiang Z, Su L, Sun Y. YOLOv7-Ship: a lightweight algorithm for ship object detection in complex marine environments. J Mar Sci Eng. 2024;12(1):190. doi:10.3390/jmse12010190.
3. Yang Z, Zhang P, Wang N, Liu T. A lightweight theory-driven network and its validation on public fully polarized ship detection dataset. IEEE J Sel Top Appl Earth Obs Remote Sens. 2024;17(1):3755–67. doi:10.1109/JSTARS.2024.3354271.
4. Han Y, Guo J, Yang H, Guan R, Zhang T. SSMA-YOLO: a lightweight YOLO model with enhanced feature extraction and fusion capabilities for drone-aerial ship image detection. Drones. 2024;8(4):145. doi:10.3390/drones8040145.
5. Asiyabi RM, Ghorbanian A, Tameh SN, Amani M, Jin SG, Mohammadzadeh A. Synthetic aperture radar (SAR) for ocean: a review. IEEE J Sel Top Appl Earth Obs Remote Sens. 2023;16(5):9106–38. doi:10.1109/JSTARS.2023.3310363.
6. Gong Y, Zhang Z, Wen J, Lan G, Xiao S. Small ship detection of SAR images based on optimized feature pyramid and sample augmentation. IEEE J Sel Top Appl Earth Obs Remote Sens. 2023;16:7385–92. doi:10.1109/JSTARS.2023.3302575.
7. Wang H, Xu F, Chen S. Saliency detector for SAR images based on pattern recurrence. IEEE J Sel Top Appl Earth Obs Remote Sens. 2016;9(7):2891–900. doi:10.1109/JSTARS.2016.2521709.
8. Robey FC, Fuhrmann DR, Kelly EJ, Nitzberg R. A CFAR adaptive matched filter detector. IEEE Trans Aerosp Electron Syst. 1992;28(1):208–16. doi:10.1109/7.135446.
9. Yang R, Pan Z, Jia X, Zhang L, Deng Y. A novel CNN-based detector for ship detection based on rotatable bounding box in SAR images. IEEE J Sel Top Appl Earth Obs Remote Sens. 2021;14:1938–58. doi:10.1109/JSTARS.2021.3049851.
10. Bao W, Huang M, Zhang Y, Xu Y, Liu X, Xiang X. Boosting ship detection in SAR images with complementary pretraining techniques. IEEE J Sel Top Appl Earth Obs Remote Sens. 2021;14:8941–54. doi:10.1109/JSTARS.2021.3109002.

11. Zhao C, Fu X, Dong J, Qin R, Chang J, Lang P. SAR ship detection based on end-to-end morphological feature pyramid network. IEEE J Sel Top Appl Earth Obs Remote Sens. 2022;15(6):4599–611. doi:10.1109/JSTARS.2022.3150910.
12. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014 Jun 23–28; Columbus, OH, USA. p. 580–7.
13. Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. doi:10.1109/TPAMI.2016.2577031.
14. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. p. 779–88.
15. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, et al. SSD: single shot MultiBox detector. In: Computer Vision-ECCV 2016: 14th European Conference; 2016 Oct 11–14; Amsterdam, The Netherlands. Cham, Switzerland: Springer. p. 21–37.
16. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision; 2017 Oct 22–29; Venice, Italy. p. 2999–3007.
17. Li G, Zhang J, Wang Y, Liu C, Tan M, Lin Y, et al. Residual distillation: towards portable deep neural networks without shortcuts. Adv Neural Inf Process Syst. 2020;33:8935–46.
18. Cui Z, Li Q, Cao Z, Liu N. Dense attention pyramid networks for multi-scale ship detection in SAR images. IEEE Trans Geosci Remote Sens. 2019;57(11):8983–97. doi:10.1109/TGRS.2019.2923988.
19. Zhang L, Liu Y, Huang Y, Qu L. Regional prediction-aware network with cross-scale self-attention for ship detection in SAR images. IEEE Geosci Remote Sens Lett. 2022;19:1–5. doi:10.1109/LGRS.2022.3212073.
20. Wang H, Han D, Cui M, Chen C. NAS-YOLOX: a SAR ship detection using neural architecture search and multi-scale attention. Connect Sci. 2023;35(1):1–32. doi:10.1080/09540091.2023.2257399.
21. Li X, Li D, Liu H, Wan J, Chen Z, Liu Q. A-BFPN: an attention-guided balanced feature pyramid network for SAR ship detection. Remote Sens. 2022;14(15):3829. doi:10.3390/rs14153829.
22. Zhao Y, Zhao L, Xiong B, Kuang G. Attention receptive pyramid network for ship detection in SAR images. IEEE J Sel Top Appl Earth Obs Remote Sens. 2020;13:2738–56. doi:10.1109/JSTARS.2020.2997081.
23. Guo H, Yang X, Wang N, Gao X. A CenterNet++ model for ship detection in SAR images. Pattern Recognit. 2021;112(7):107787. doi:10.1016/j.patcog.2020.107787.
24. Zhu M, Hu G, Zhou H, Wang S, Feng Z, Yue S. A ship detection method via redesigned FCOS in large-scale SAR images. Remote Sens. 2022;14(5):1153. doi:10.3390/rs14051153.
25. Liu Z, Gao Y, Du Q, Chen M, Lv W. YOLO-extract: improved YOLOv5 for aircraft object detection in remote sensing images. IEEE Access. 2023;11:1742–51. doi:10.1109/ACCESS.2023.3233964.
26. Wang Q, Shen F, Cheng L, Jiang J, He G, Sheng W, et al. Ship detection based on fused features and rebuilt YOLOv3 networks in optical remote-sensing images. Int J Remote Sens. 2021;42(2):520–36. doi:10.1080/01431161.2020.1811422.
27. Chen D, Ju Y. Ship detection in SAR image based on improved YOLOv3. Syst Eng Electron. 2021;43(4):937–43. doi:10.12305/j.issn.1001-506X.2021.04.10.
28. Fan W, Zhao S, Guo L. Ship detection algorithm based on improved RetinaNet. J Comput Appl. 2022;42(7):2248.
29. Ma B, Hua Z, Wen Y, Deng H, Zhao Y, Pu L, et al. Using an improved lightweight YOLOv8 model for real-time detection of multi-stage apple fruit in complex orchard environments. Artif Intell Agric. 2024;11(4):70–82. doi:10.1016/j.aiia.2024.02.001.
30. Yang G, Wang J, Nie Z, Yang H, Yu S. A lightweight YOLOv8 tomato detection algorithm combining feature enhancement and attention. Agronomy. 2023;13(7):1824. doi:10.3390/agronomy13071824.
31. Jiang T, Chen S. A lightweight forest pest image recognition model based on improved YOLOv8. Appl Sci. 2024;14(5):1941. doi:10.3390/app14051941.
32. Wang G, Chen Y, An P, Hong H, Hu J, Huang T. UAV-YOLOv8: a small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios. Sensors. 2023;23(16):7190. doi:10.3390/s23167190.

33. Huang Z, Li L, Krizek GC, Sun L. Research on traffic sign detection based on improved YOLOv8. J Comput Commun. 2023;11(7):226–32. doi:10.4236/jcc.2023.117014.
34. Lou H, Duan X, Guo J, Liu H, Gu J, Bi L, et al. DC-YOLOv8: small-size object detection algorithm based on camera sensor. Electronics. 2023;12(10):2323. doi:10.3390/electronics12102323.
35. Wang X, Gao H, Jia Z, Li Z. BL-YOLOv8: an improved road defect detection model based on YOLOv8. Sensors. 2023;23(20):8361. doi:10.3390/s23208361.
36. Wu T, Dong Y. YOLO-SE: improved YOLOv8 for remote sensing object detection and recognition. Appl Sci. 2023;13(24):12977. doi:10.3390/app132412977.
37. Zhang LJ, Fang JJ, Liu YX, Feng LH, Rao ZQ, Zhao JX. CR-YOLOv8: multiscale object detection in traffic sign images. IEEE Access. 2024;12:219–28. doi:10.1109/ACCESS.2023.3347352.
38. Chen H, Wang Y, Guo J, Tao D. Vanillanet: the power of minimalism in deep learning. Adv Neural Inf Process Syst. 2023;36:7050–64.
39. Li C, Zhou A, Yao A. Omni-dimensional dynamic convolution. arXiv:2209.07947. 2022.
40. Yang B, Bender G, Le QV, Ngiam J. Condconv: conditionally parameterized convolutions for efficient inference. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems; 2019 Dec 8–14; Vancouver, BC, Canada.
41. Chen Y, Dai X, Liu M, Chen D, Yuan L, Liu Z. Dynamic convolution: attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA. p. 11030–9.
42. Li H, Li J, Wei H, Liu Z, Zhan Z, Ren Q. Slim-neck by GSConv: a better design paradigm of detector architectures for autonomous vehicles. arXiv:2206.02424. 2022.
43. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. 2017.
44. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L. MobileNetV2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
45. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for MobileNetV3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019 Oct 27–Nov 2; Seoul, Republic of Korea.
46. Zhang X, Zhou X, Lin M, Sun J. An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA.
47. Ma N, Zhang X, Zheng H, Sun J. ShuffleNet V2: practical guidelines for efficient CNN architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV); 2018 Sep 8–14; Munich, Germany.
48. Tong Z, Chen Y, Xu Z, Yu R. Wise-IoU: bounding box regression loss with dynamic focusing mechanism. arXiv:2301.10051. 2023.
49. Wei S, Zeng X, Qu Q, Wang M, Su H, Shi J. HRSID: a high-resolution SAR images dataset for ship detection and instance segmentation. IEEE Access. 2020;8:120234–54. doi:10.1109/ACCESS.2020.3005861.
50. Zhang T, Zhang X, Li J, Xu X, Wang B, Zhan X, et al. SAR ship detection dataset (SSDD): official release and comprehensive data analysis. Remote Sens. 2021;13(18):3690. doi:10.3390/rs13183690.
51. Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, et al. MMDetection: open mmlab detection toolbox and benchmark. arXiv:1906.07155. 2019.