Anchor-Free SAR Ship Instance Segmentation With Centroid-Distance Based Loss

Fei Gao, Yiyang Huo, Jun Wang, Amir Hussain, and Huiyu Zhou

***Abstract*—Instance segmentation methods for synthetic aperture radar (SAR) ship imaging have certain unsolved problems. 1) Most of the anchor-based detection algorithms encounter difficulties in tuning the anchor-related parameters and high computational costs. 2) Different tasks share the same features without considering the differences between tasks, leading to mismatching of the shared features and inconsistent training targets. 3) Common loss functions for instance segmentation cannot effectively distinguish the positional relationships between ships with the same degree of overlap. In order to alleviate these problems, we first adopt a lightweight feature extractor and an anchor-free convolutional network, which effectively help to reduce computational consumption and model complexity. Second, to fully disseminate feature information, a dynamic encoder–decoder is proposed to dynamically transform the shared features to task-specific features in channel and spatial dimensions. Third, a novel loss function based on centroid distance is designed to make full use of the geometrical shape and positional relationship between SAR ship targets. In order to better extract features from SAR images in complex scenes, we further propose the dilated convolution enhancement module, which utilizes multiple receptive fields to take full advantage of the shallow feature information. Experiments conducted on the SAR ship detection dataset prove that the method proposed in this article is superior to the other state-of-the-art algorithms in terms of instance segmentation accuracy and model complexity.**

***Index Terms*—Anchor-free, convolutional neural network (CNN), instance segmentation, synthetic aperture radar (SAR).**

I. Introduction

Synthetic aperture radar (SAR) ship images play a significant role in many aspects, such as water traffic monitoring, fishery monitoring, marine vessel management, intelligence acquisition, and so on [1]. As basic methods for SAR image processing, target detection and recognition have received much attention in recent years [2], [3]. Thanks to the powerful automated feature extraction ability of deep-learning-based methods, accuracy and efficiency of target detection and recognition have been greatly improved, however, they are still incapable of describing the target shape [4], [5]. Therefore, it is necessary to pay more attention to instance segmentation for better descriptions of the target contour in SAR ship images.

Fig. 1 shows illustrative results of ship detection, ship semantic segmentation, and ship instance segmentation.

Ship detection focuses on obtaining the vertical or rotating bounding box of a ship indicating the ship's position. Semantic segmentation of ships attaches attention to the shape of the ships, where a mask is used to describe the ship contour. However, it merely distinguishes the ship category from the background category. Compared to detection and semantic segmentation, instance segmentation is not only dedicated to assigning category labels on a pixel-by-pixel basis, but also to distinguishing different objects of the same category. The category, location, and contour information of the targets are all well obtained. With these characteristics, the ship instance segmentation has profound significance for the application of ship images in multiple fields, and will undoubtedly become a fundamental task of SAR image processing in the future.

Many instance segmentation architectures have been proposed and achieved outstanding performance in the natural scene. Mask R-CNN [6] is a representative method in instance segmentation. A segmentation branch is added to faster R-CNN [7] with region of interest (ROI) pooling replaced by ROI align. To further enhance spatial text information, Chen *et al.* [8] designed Cascade Mask R-CNN, introducing a direct flow of information between mask branches in different stages. PA Net by Liu *et al.* [9] proposed bottom–up path enhancement, adaptive feature pooling, and full connection fusion to improve the performance of instance segmentation. To reduce the errors between mask quality and mask confidence, Huang *et al.* [10] presented an instance segmentation method based on Mask R-CNN named Mask Scoring R-CNN. They added the mask-IOU branch and re-evaluated the mask confidence determined by the classification score, achieving consistent and significant benefits. Precise ROI pooling was suggested by Su *et al.* [11] to solve the accuracy loss of optical remote sensing images due to coordinate quantization, and they introduced high-resolution feature fusion pyramid network to reduce spatial resolution loss in the pyramid network. There are also some methods proposed in remote sensing

Manuscript received August 14, 2021; revised October 2, 2021; accepted October 18, 2021. Date of publication October 29, 2021; date of current version November 17, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61771027, Grant 61071139, Grant 61471019, Grant 61501011, and Grant 61171122. The work of Amir Hussain was supported by the U.K. Engineering, and Physical Sciences Research Council (EPSRC) under Grant EP/M026981/1, Grant EP/T021063/1, and Grant EP/T024917/1. The work of Huiyu Zhou was supported in part by the Royal Society-Newton Advanced Fellowship under Grant NA160342, and in part by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska Curie Grant 720325. *(Corresponding author: Jun Wang.)*

Fei Gao, Yiyang Huo, and Jun Wang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: feigao2000@163.com; hyy1604018154@163.com; wangj203@buaa.edu.cn).

Amir Hussain is with the Cyber and Big Data Research Laboratory, Edinburgh Napier University, EH11 4BN Edinburgh, U.K., and also with the Taibah Valley, Taibah University, Medina 30001, Saudi Arabia (e-mail: a.hussain@napier.ac.uk).

Huiyu Zhou is with the Department of Informatics, University of Leicester, LE1 7RH Leicester, U.K. (e-mail: hz143@leicester.ac.uk).

Digital Object Identifier 10.1109/JSTARS.2021.3123784

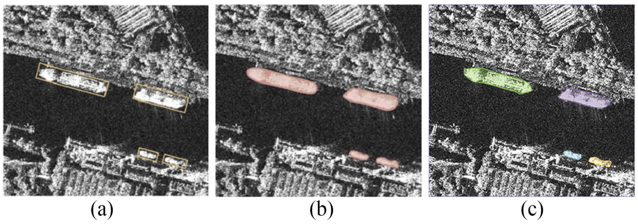


Fig. 1. Illustrative results. (a) Ship detection. (b) Ship semantic segmentation. (c) Ship instance segmentation.

images processing. HQ-ISNet [12] introduced the high resolution feature pyramid network (HRFPN) to maintain high-resolution feature maps in the network and designed a tiny network to replace the original mask branch. Zhang *et al.* [13] extended the original single-scale mask branch into the scale complementary mask branch to deal with the under segmentation problem caused by multiple scales of geospatial instances. These methods all rely on the manually set anchors to generate ROIs through region proposal network (RPN). YOLACT [14] was developed based on the single-stage detection network Retina Net [8]. It directly generates ROI without RPN, but still needs to set anchors in advance. The above deep convolutional neural networks (DCNNs) all adopt an anchor-based mechanism for instance segmentation, leading to some disadvantages when processing SAR ship images.

First, the above methods are sensitive to hyper-parameter settings such as anchor size [15]. SAR ship images have three characteristics: the relatively large aspect ratio, the varied scales, and the dense distribution [16]. Densely set anchors may cause high computational consumption and an imbalance between the distributions of positive and negative samples. To overcome the shortcomings of the anchor box, researchers turned to instance segmentation with anchor-free methods. CenterMask [17] embedded a novel mask branch based on spatial attention guidance into the target detection framework FCOS [18]. The mask is acquired by ROI center prediction, and distance regression between the bounding box boundary and the center point position. PolarMask [19] used FCOS as backbone, formulated the instance segmentation problem as instance center prediction and dense distance regression in a polar coordinate. The anchor-free methods are also introduced in remote sensing images processing. Liu and Di [20] embedded a global context parallel attention module into the anchor-free instance segmentation framework to capture the global information. Polar template mask [21] used a nonuniform angle polar coordinate system to solve the problem with masks not being smooth and accurate enough due to the uniform angle sampling of the ship targets with large aspect to achieve competitive accuracy. Inspired by these methods, this article adopts an anchor-free network based on center point prediction, significantly improving the calculation efficiency.

Second, all the above anchor-based methods encode the position information of the instance into the ROI, and the instance mask is generated by using the full convolutional network named mask head to act on the ROI or global feature map. It is difficult to encode irregular shape information of the ship targets using rectangular ROIs. In view of the problem, the mask head needs a large receptive field to encode enough context information. For instance, in mask R-CNN, four 3*3 convolution kernels with 256 channels are adopted, which significantly increases the computational costs. Therefore, researchers aim to to perform more effective coding to reduce computational consumption. Jia *et al.* [22] introduced the concept of dynamic filter network, which dynamically generates a filter adapted to the network input. Conditionally parameterized convolutions (CondConv) proposed by Yang *et al.* [23] calculates the weighted convolution kernel on the input samples before performing the convolution calculation, which improves the model capacity while maintaining high efficiency. Therefore, we refer to the controller subnetwork design of Tian *et al.* [24], which encodes features such as the relative position and the instance shape into the dynamic mask heads of each instance, effectively reducing the complicated calculations caused by ROI.

Third, although FPN has been proven to improve the model's segmentation performance for multi-scale targets, the adoption of FPN in two-stage anchor-based methods will greatly increase computational consumption. Thus, some two-stage anchor-based methods just perform instance segmentation on the feature map with lowest resolution and strongest semantic information for efficiency, resulting in missed detection of small targets. The parameters also lead to over-fitting in training on the SAR dataset with limited training samples [25]. Generally, researchers train models by fine-tuning the pre-training model on the ImageNet dataset [26]. However, in the process of domain migration, it is easy to introduce learning bias due to different objective functions and huge differences in target distribution. To address these problems, this article uses a lightweight feature extraction network based on the Ghost Net [27] to extract multiscale ship features, which improves the network segmentation speed while improving the generalization performance of the network on a limited-sample dataset without a pretraining model. To extract richer features, we appropriately widen the channels of the Ghost Net. Deepest features are more suitable for the segmentation of large-scale targets due to the limitation of receptive fields. Therefore, our article further presents the dilated convolution feature enhancement module (DEM) to use multiple receptive fields to make full use of the shallow feature information in the high-resolution images.

Compared to the methods based on anchors, anchor-free methods possess a simpler model structure and exhibit a higher speed. However, they still have some shortcomings. For example, PolarMask conducts instance segmentation tasks through center point prediction and dense distance regression, without considering that the features corresponding to the classification task need to remain unchanged with translation and scale changes, while the features related to dense distance regression need to change with positional changes. If two types of tasks share the same feature as input, it will not only impede the propagation of the features, but also lead to inconsistent training targets during training [28]. Liu *et al.* [28] suggested that the gradient flows required for classification tasks and location regression tasks be dispersed, thus, greatly alleviating the problem. Segmenting objects by locations (SOLO) series by Wang *et al.* [29], [30] added two parallel classification and location prediction branches after each layer of the feature pyramid networks (FPN) [31], effectively dispersing the gradient flows from the spatial dimension. However, the required feature information is not specifically selected for different tasks. Inspired by the attention mechanism, Yang *et al.* [32] designed an encoder–decoder (ED) which contains two decoders to enhance the salient features for different tasks, but still decentralizes the gradient flows only from the spatial dimension. Based on their work, our article introduces the dynamic ED. Here, the dynamic factor branch and the encoding and decoding branch are effectively utilized and gradient flows are decentralized from the spatial dimension and the channel dimension, leading to an appropriate allocation of the required feature information for different tasks.

At the same time, the anchor-free methods typically contain multiple parallel branches. Most existing state-of-the-art methods use dice loss [33] as the mask branch loss function. Dice loss only evaluates the similarity between the predicted target and the true target from the overlap ratio between them. However, for the training process of ship targets, there are many positional situations when the predicted target overlaps with the true target in the same proportion. Dice loss cannot be used to effectively distinguish between these positional relationships, if the overlap is fixed. Thus, this article first proposes to use the distance between the centroid of the ship targets to weight the loss function to take advantage of the positional relationships. To make more effective use of the geometry characteristics of ships, we further put forward a corrosion algorithm to generate the central area. Weighting the loss function with the centroid distance and the overlap degree of the central area simultaneously, the network regression can be effectively guided.

To sum up, to solve the problems in the aforementioned instance segmentation methods, in this article, Ghost Net with widen channels as the feature extractor is used to improve calculation efficiency. Simultaneously, we use DEM to enhance shallow features through multiple receptive fields, the multiscale deep and shallow features are then combined with feature fusion networks to enhance the representation of features and improve the model generalization ability. In the process of feature fusion, our article introduces the convolutional block attention module (CBAM) [34] to enhance salient features and suppress background clutters. Following this, the feature map is processed by the dynamic ED to generate features suitable for different tasks. Finally, this article presents a novel loss function based on the geometric characteristics of ships and the positional relationship between ship targets. The experimental results on SSDD [35] show that this method can obtain better segmentation accuracy than the other state-of-the-art instance segmentation methods without a pretraining model.

The rest of this article is arranged as follows. Section II introduces the methods used in this article in detail, which is mainly divided into the feature extraction network based on Ghost Net, the DEM, the feature fusion module, the dynamic ED, the center point based instance segmentation predictor and the centroid distance (CD)-based loss function. Section III provides experimental details and experimental results on SSDD. Finally, Section IV concludes this article.

II. Methodology

Fig. 2 illustrates the detailed architecture of the proposed method in this article which can be divided into four parts from left to right: the feature extraction network, the tree-like feature fusion network, the dynamic ED and the center point-based instance segmentation predictor. First, the SAR image passes a convolution layer with a step size of 2 to reduce the size of the feature map, and then is fed as the input of the feature extractor, through which features of four different scales $\{C_4, C_3, C_2, P_1\}$ are obtained. The resolution of the features reduces to half after each stage. As the deepest feature P_1 contains strong semantic information and large receptive fields, $\{C_4, C_3, C_2\}$ are chosen to perform feature enhancement with DEM to obtain $\{P_4, P_3, P_2\}$. In the feature fusion network, $\{P_4, P_3, P_2, P_1\}$ are combined in the upsampling process to generate multiscale high-resolution feature maps F_{out}. Two feature maps F_{out1} and F_{out2} of the same size are generated in parallel by the dynamic ED. F_{out1} serves as the input of the center point prediction network, F_{out2} not only needs to be the input of the ship size regression network, the ship bias regression network and the dynamic controller generation network, but also needs to be concatenated with the relative coordinate map as the feature map for instance segmentation. In the training stage, the multipart loss functions are calculated according to the center point's information and the mask information of the ship targets. The losses are combined to train the branches jointly. In the inference stage, the output of each branch is combined to realize the ship instance segmentation without using any anchor and nonmaximum suppression (NMS) postprocessing. Next, this article will introduce the structure and the principle of these parts one by one.

A. Feature Extraction Network Based on Ghost Net

Existing instance segmentation methods mostly use DCNNs such as ResNet-50 and ResNet-101 [36] for multilevel feature extraction, segmenting only on shallow low-resolution feature maps to achieve a balance of accuracy and efficiency. However, the target scales in the SAR ship images are diverse. It is necessary to use multiscale, multilevel, high-resolution feature maps to obtain the feature information of multiscale targets. Merely using shallow low-resolution feature maps will inevitably impair the model's ability for multiscale instance segmentation. Therefore, many scholars have begun to devote themselves to efficient network designs to reduce the network complexity, proposing outstanding lightweight networks, such as Mobile Net [37], Shuffle Net [38], and the latest Ghost Net.

According to the Ghost Net theory, trained networks often contain rich or even redundant feature information to ensure that the network fully understands the inputs. To generate these redundant features in a cost-effective way, they proposed a ghost module and the Ghost Net on the basis of the design of Mobile Net v3 [37], replacing the bottle block with the ghost bottleneck, obtaining state-of-the-art performance. In this article, we adopt

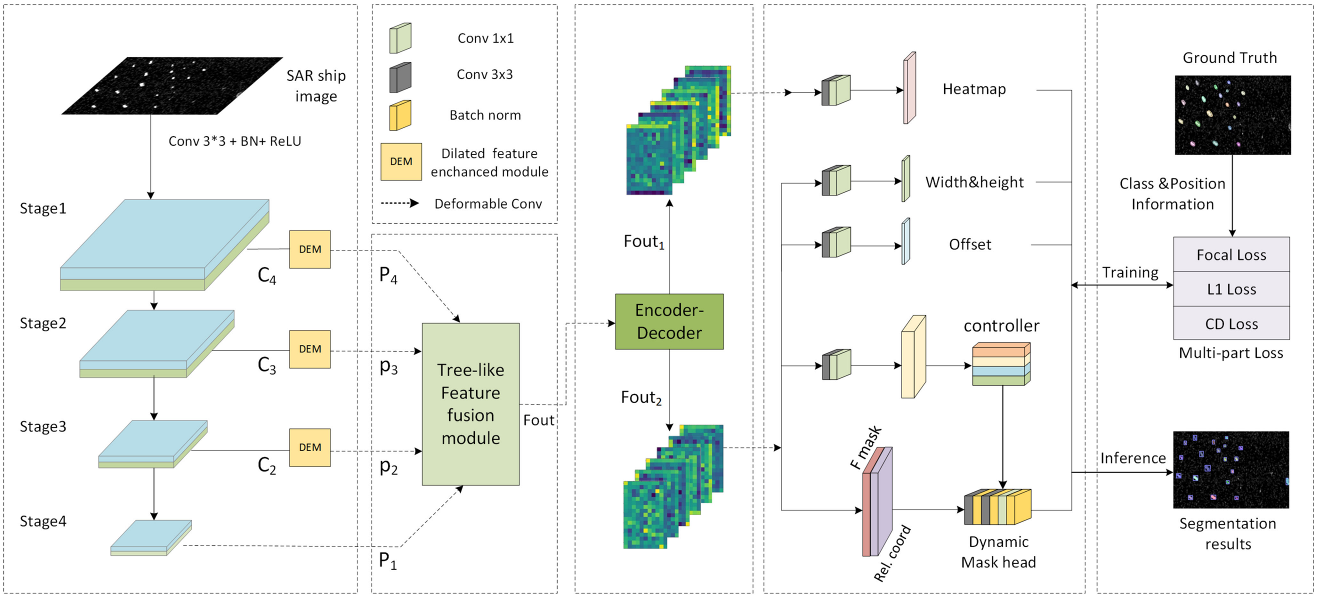


Fig. 2. Overall structure diagram. The feature extraction network, the tree-like feature fusion network, the dynamic encoder–decoder and the center point based segmentation predictor. C_4, C_3, C_2, P_1 are the four different scale extracted features; P_4, P_3, P_2 are the enhanced features. F_{out} is the output fusion feature map produced by the tree-like feature fusion network. F_{out1} and F_{out2} are the feature maps suitable for different tasks.

the lightweight Ghost Net to extract the multiscale features of the SAR images.

The number of the channels in the middle layer, that is, the width of the network, greatly influences the generalization capabilities of the network. Ghost Net is originally designed for target recognition tasks, and the extracted features are not capable for detection and instance segmentation tasks. A wider network allows each layer to acquire richer features, such as different orientations and variant sizes, but also results in a significant increase in model parameters. To achieve a balance between the number of parameters and the fitting ability, Ghost Net gives the empirical value of the widening factor between 0.8 and 2. We follow the thought of Ghost net and use the following rules to adjust the output channels:

$$c_t = \max(d, \lfloor\lfloor \alpha c_{\text{old}} + d/2\rfloor/d\rfloor \times d)$$

$$c_{\text{new}} = \begin{cases} c_t + d & c_t < 0.9\alpha c_{\text{old}} \\ c_t & c_t \geq 0.9\alpha c_{\text{old}} \end{cases} \qquad (1)$$

where d indicates a divisor. In this article, we set $d = 4$ in all experiments. c_{old} and c_{new} denote the original dimension and adjusted dimension of the output features, respectively. α is the adjustment ratio, a typical range for α is (0,2). $\lfloor\cdot\rfloor$ represents the rounding down operation. $c_t + d$ makes sure that round down does not go down by more than 10%. And the adjusted dimension of the output features satisfies that it can be divided by d. For example, when the original dimensions of the output features are $\{16, 24, 40, 80, 112, 160\}$, given $\alpha = 1.3$, the output dimensions will be adjusted to $\{20, 32, 52, 104, 144, 208\}$. In this article, We choose to widen the channels by 1.5 times and give the wide-channel Ghost Net structure in Table I.

According to the size of the input feature map, the extraction processes can be divided into five stages. Except that the first stage contains a standard convolution structure, each stage consists of a set of ghost bottlenecks. Ghost bottleneck is composed of two ghost modules and jump connections, with the similar structure to the residual block in the deep residual network (ResNet) [36]. The former ghost module is used to increase and decrease the number of channels, while the latter can effectively avoid the disappearance of the gradient after deepening the network layers. For the ghost bottleneck with a step size of 2, a depth-wise separable convolution (DSConv) is inserted into

TABLE I
FEATURE EXTRACTION NETWORK STRUCTURE

Stage	Input	Op	Exp	C	SE	S	Out
0	512×512×3	Conv 3×3	-	24	-	2	-
	256×256×24	G neck	16	24	-	1	-
	256×256×24	G neck	48	36	-	2	-
1	128×128×36	G neck	72	36	-	1	-
	128×128×36	G neck	72	60	1	2	P_4
2	64×64×36	G neck	120	60	1	1	-
	64×64×36	G neck	240	120	-	2	C_3
3	32×32×120	G neck	200	120	-	1	-
	32×32×120	G neck	184	120	-	1	-
	32×32×120	G neck	184	120	-	1	-
	32×32×120	G neck	480	168	1	1	-
	32×32×168	G neck	672	168	1	1	-
	32×32×168	G neck	672	240	1	2	C_2
4	16×16×240	G neck	960	240	-	1	-
	16×16×240	G neck	960	240	-	1	-
	32×32×120	G neck	960	240	1	1	-
	32×32×120	G neck	960	240	1	1	C_1

Exp represents the expanded size, Out represents the number of output channels, SE represents whether to use the SE attention module, and S represents the ghost bottleneck step size. G neck is the abbreviation for ghost bottleneck.

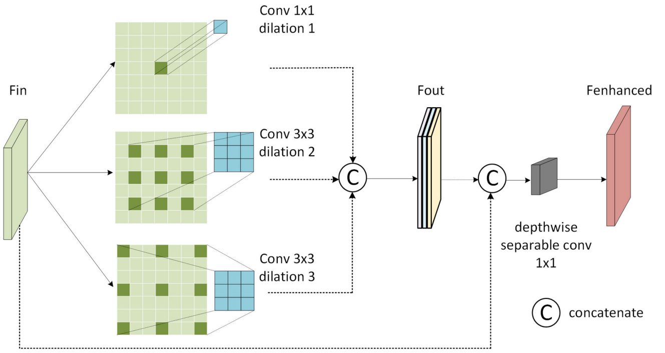


Fig. 3. Structure diagram of the DEM. F_{in} is the input high-resolution shallow feature map; F_{out} is the output result of the three dilated convolution cascades; F_{enhanced} is the final enhanced feature map.

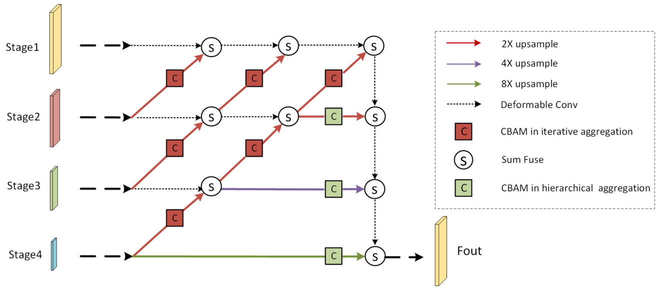


Fig. 4. Structure diagram of the tree-like feature fusion module.

the two ghost modules. By replacing the standard convolution with a combination of a depth-wise convolution and a point-wise convolution, the computational cost of DSConv is reduced by a factor of $(k^2 + d_0)/(d_0 k^2)$ [39]. Where d_0 represents the number of the convolution output channels, and k represents the size of convolution kernel. After the second Ghost module, the rectified linear unit (ReLU) [40] is not used as the activation function to avoid the information loss of manifold of interest.

B. Dilated Convolution Feature Enhanced Module

Shallow features have smaller receptive fields compared to high-level features, and are suitable for segmentation of smaller targets. But these features also contain the characteristic information of large-scale targets. Using different receptive fields to further extract features can better describe multiscale targets and make full use of the high-resolution features [41]. We design the DEM shown in Fig. 3 to further enhance the shallow features in Ghost Net.

Dilated convolution implants regular holes in ordinary convolution to generate a larger receptive field without the increase of parameters and calculations. This process can be expressed as

$$Rec_{\text{dilated}} = \beta \times Rec_{\text{standard}} + 1 \tag{2}$$

where d, Rec_{dilated}, and Rec_{standard} represent the dilated rate, dilated convolution receptive field, and standard convolution receptive field, respectively. Three types of convolution kernels are used in the DEM. The dilated rate of the first 1×1 convolution is 1, with a same receptive field as the standard 1×1 convolution; The second dilated convolution is 3×3 amounts to the standard 5×5 convolution, and the third dilated convolution 3×3 equals to a standard 7×7 convolution. For the input high-resolution feature map $F_{\text{in}} \in \mathbb{R}^{C \times H \times W}$ after parallel convolution processing, the size of the feature map generated by each branch is the same as the input, and the number of the channels is compressed to half. The output $F_{\text{out}} \in \mathbb{R}^{\frac{3}{2}C \times H \times W}$ gets channels 1.5 times that of the input after concatenating parallel branches. In order to fully preserve the information of the original feature map, this article adds a skip connection path to concatenate F_{out} with the original input F_{in}. The size of the final output feature map and the number of the channels are restored to be consistent with the input through depth separable convolution.

To avoid too many holes, leading to discontinuous feature extraction and loss of some effective features, larger dilated rates are not used. At the same time, three parallel dilated convolutions can achieve sufficient feature enhancement. More parallel branches or larger convolution kernels contribute to improvement of the accuracy slightly and will increase the number of the parameters.

C. Tree-Like Feature Fusion Module

Aggregation is a commonly used technique for designing network structures. How to integrate information between different stages and blocks is the research direction of many scholars. The most common aggregation method at present is skip connection. However, this method is limited to simple superposition within the block without information aggregation between the blocks. In the feature extraction network, with the stage going deeper, the resolution of the features gradually decreases, the receptive field and the semantic information increase. The deep-level features are helpful for target classification while bringing harm to the localization ability of target detection. Shallow feature maps have higher resolution, but their low-level features impair the ability of object recognition. In order to obtain high-resolution feature maps with strong semantic information. Yu *et al.* [42] proposed a tree-like feature fusion module to iteratively fuse the feature information. Inspired by their work, we adopt the tree-like feature fusion module to compound multiscale features, and finally output high-resolution multiscale feature maps. The adopted tree-like feature fusion module structure is shown in Fig. 4.

The tree-like feature fusion module consists of iterative aggregation and hierarchical aggregation. As shown in the upper left part of Fig. 4, iterative aggregation is responsible for linking the features of two adjacent stages to make sure that the deep and shallow expressions can be better integrated. Hierarchical aggregation is shown in the lower right part of Fig. 4, which aggregates stages in the tree to retain and combine feature channels. The aggregation in the channel direction realizes semantic fusion, which improves the ability to infer what it is; the aggregation in resolution and scale directions realizes spatial fusion, which contributes more to the ability to infer where it is.

Recent studies have proved that the attention mechanism is helpful for strengthening the salient features of the target and improve the performance of SAR image processing. Inspired by Cui *et al.* [43], we add CBAM to the process of feature upsampling of the additive fusion. The upsampling layer is added after the original CBAM, here we follow the design in [44] and adopt deconvolution in different steps according to different additive fusion process. Red blocks in Fig. 4 denote the CBAM used in the process of iterative aggregation, with deconvolution in steps of 2. Green blocks represent the CBAM used in the process of hierarchical aggregation, with deconvolution in steps of 2, 4, or 8 for different stages. The core idea of CBAM is that the importance of the features in different channels dimension and spatial dimension is different. Thus, the feature maps can be optimized by assigning weights for different channels and spatial positions. CBAM contains two independent submodules, channel attention module (CAM) and spatial attention module (SAM). Given the feature map $F_{in} \in \mathbb{R}^{H\times W\times C}$, where C, H, and W represent the channel number, height, and width of F_{in}. CAM first adopts global maximum pooling and global average pooling to generate two channel description maps $F_{\text{max}}^{c} \in \mathbb{R}^{H\times W\times 1}$ and $F_{\text{ave}}^{c} \in \mathbb{R}^{H\times W\times 1}$, respectively. F_{max}^{c} and F_{ave}^{c} are then sent to a shared two-layer neural network. After the two output features are added on corresponding elements, the weight coefficient W_c is obtained through a sigmoid activation function. Finally, multiply W_c and F_{in} to get the new feature as the input for SAM. The structure of SAM is similar to that of CAM, and generates the weight coefficient W_s. Finally, we multiply W_s with F_{in}, thus, the weights for different channels and spatial positions in the original feature map are assigned. During multiscale feature fusion, features with stronger semantic information play a more vital role in the identification and positioning of ship targets. CBAM helps us to strengthen the saliency features in the high-level features and suppress the background clutters, thereby improving the accuracy of the target response.

D. Dynamic Encoder–Decoder

Although the tree-like feature fusion module effectively improves the model's ability to segment multiscale targets, these methods mainly focus on global feature enhancement, without feature optimization for different tasks. Take classification tasks and position regression tasks as examples, classification tasks require features to remain unchanged during translation and scale changes; position regression tasks involve regression of target sizes and center point offsets, requiring features to be varied corresponding to positions in translation and scale changes. Different tasks use the shared features without considering the differences between tasks will cause mismatching of the shared features and inconsistent training targets.

Liu *et al.* [28] disperse the gradient flow required by the classification task and the position regression task in the spatial dimension, thereby greatly alleviating the above problem. However, copying feature maps directly in the spatial dimension will not acquire the features specifically for different tasks. Inspired by the ability of the attention module to enhance the salient features through the encoding and decoding structure, we design the dynamic ED to disperse the gradient flow required by the classification task and the position regression task from the spatial dimension and the channel dimension. The structure of the dynamic ED is well illustrated in Fig. 5.

The dynamic ED mainly consists of the dynamic factor branch and the encoding–decoding branch. The former adds a global weight to the high-resolution feature map, and assigns different degrees of global information to each task from the spatial dimension. This process can be expressed as

$$\begin{aligned} F_{d1} &= w_1 \times F_1 \\ F_{d2} &= w_2 \times F_2 \end{aligned} \tag{3}$$

$w_1 \in (0,1)$ and $w_2 \in (0,1)$ are dynamically generated during the training process. The encoding–decoding branch is composed of an encoder and two decoders. The encoder first performs adaptive maximum pooling on the input high-resolution feature map to compress the global information of each channel, and then encodes information by compressing the number of channels. This process can be described as

$$\begin{aligned} F_{\text{zip}} &= \text{Adaptivemaxpooling}(F_{\text{in}}) \\ F_{\text{encoder}} &= \text{ReLU}(w_e F_{\text{zip}}) \end{aligned} \tag{4}$$

where $F_{in} \in \mathbb{R}^{C\times H\times W}$ denotes the input feature, $F_{zip} \in \mathbb{R}^{C\times 1\times 1}$ represents the feature map after space compression, and $w_e \in \mathbb{R}^{\frac{C}{r}\times C}$ represents the weight of channel compression in the encoder, r is the compression ratio. Channel compression allows us to learn the degree of dependence of different tasks on the feature information of each channel, make full use of useful features, and suppress useless features. To prevent the loss of feature information due to a large compression rate, we set $r = 2$ here [32]. Then, the encoded feature sequence is employed to two decoders, respectively, restoring the number of channels to C. And the semantic global information is assigned to different tasks from the channel dimension. This process can be illustrated as

$$\begin{aligned} F_{\text{decoder1}} &= \text{Sigmod}(w_{d1} F_{\text{zip}}) \\ F_{\text{decoder2}} &= \text{Sigmod}(w_{d2} F_{\text{zip}}) \end{aligned} \tag{5}$$

$w_{d1} \in \mathbb{R}^{C\times \frac{C}{r}\times W}$ and $w_{d2} \in \mathbb{R}^{C\times \frac{C}{r}\times W}$ denote the weight of the expansion part of the decoder channel. $F_{\text{decoder1}} \in \mathbb{R}^{C\times 1\times 1}$ and $F_{\text{decoder2}} \in \mathbb{R}^{C\times 1\times 1}$ represent the decoded feature sequences. The value in the decoded sequence is normalized to (0,1) through the Sigmod function to decide whether the features in each channel will be enhanced or suppressed accordingly.

Through the dynamic factor and the encoding–decoding branches, the dependence of different tasks on the global information from the spatial and channel dimensions of the feature map can be obtained, and the useless information will be suppressed. Finally, the subsequent features are further aggregated and restored to the same size as that of the input high-resolution feature map. This process can be expressed as

$$\begin{aligned} F_1 &= \text{Conv2d}(F_{d1} \textcircled{c} F_{\text{decoder1}}) \\ F_2 &= \text{Conv2d}(F_{d2} \textcircled{c} F_{\text{decoder2}}) \end{aligned} \tag{6}$$

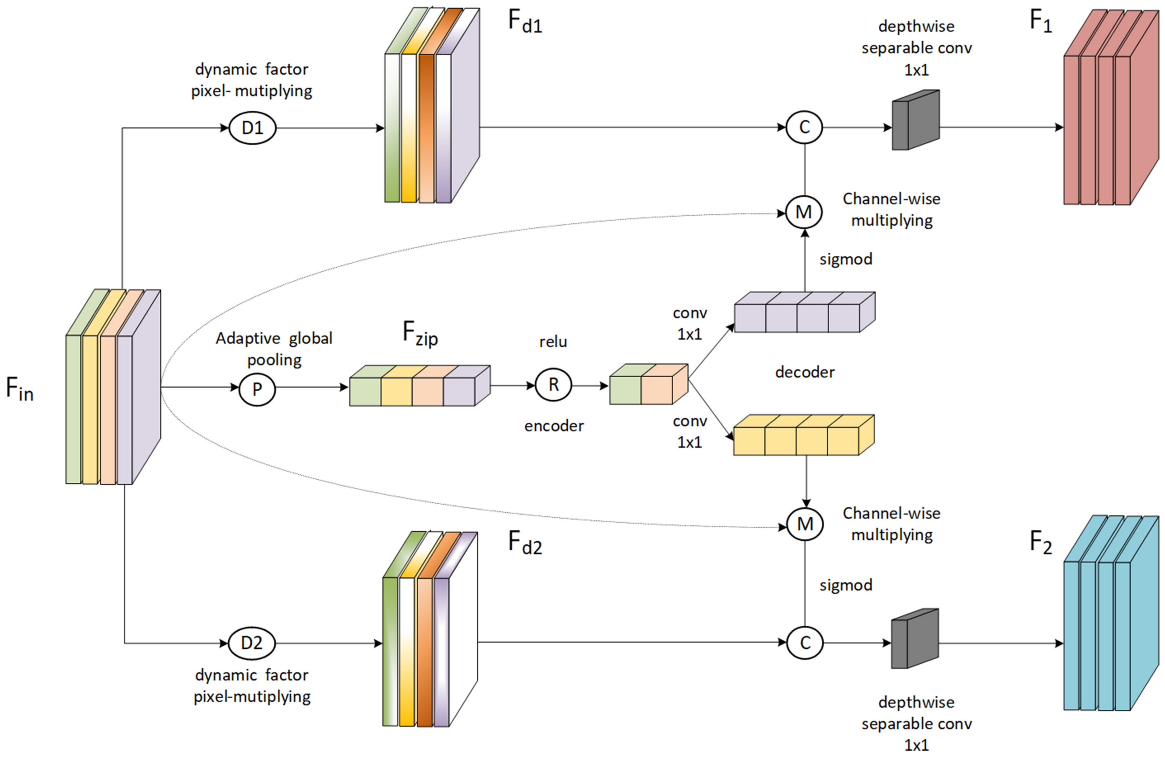


Fig. 5. Dynamic ED structure diagram. F_{in} is the high-resolution feature map after fusion; F_{d1} and F_{d2} are high-resolution feature maps multiplied by dynamic factors; F_1 and F_2 are feature maps suitable for classification and position regression tasks after encoding and decoding.

ⓒ represents concatenate operations, and DSConv is adopted to reduce the number of the parameters. Through feature aggregation, the model not only dynamically allocates suitable features for different tasks from the channel and spatial dimensions, but also achieves the dispersion of gradient flows for different tasks.

E. Center Point-Based Instance Segmentation Predictor

Among the two-stage instance segmentation algorithms, most of them regard the instance segmentation task as a combination of target detection and semantic segmentation, adding a segmentation branch to the target detection network. Recently, some single-stage anchor-free instance segmentation methods have been developed. These methods conduct instance segmentation on ship targets by simultaneously predicting the key points and regressing the size of the targets, without the presetting and correction of the anchor. Therefore, the results can be obtained effectively end to end. A center point based and anchor-free instance segmentation predictor is adopted in this article.

However, there is redundancy in both types of methods in terms of mask generation. For the two-stage methods, the position and shape information of the instance is generally encoded in ROIs, on which the mask head needs to generate the mask. For SAR ships with irregular shapes, the mask head requires a relatively larger receptive field to encode enough context information, which significantly increases the computational consumption. For the single-stage methods, it is necessary to extract the position information of the instance from the global feature map into the mask head corresponding to the predicted center point. For multiple targets contained in one image, mask heads need to be generated at a time, which significantly increases the parameter quantity of the mask head.

In order to solve these problems, we refer to conditional convolutions for instance segmentation (CondInst) and a N dimensions vector named controller is designed to dynamically generate filters in the mask head conditioned on instances, where N denotes the total parameters of the mask head. The controller branch obtains the target position from the center point prediction branch and the characteristics of the target instance from the input feature map, encoding the feature information into the dynamic controller. Then the encoded features will be employed to the mask head, which will act on the global feature map concatenated with relative position information to generate the corresponding instance mask. For $F_{\text{mask}} \in \mathbb{R}^{H_{\text{mask}} \times W_{\text{mask}} \times C_{\text{mask}}}$, C_{mask} is proven to achieve superior performance and using a larger channel dimension cannot improve the performance. The mask head has 169 parameters in total (weights $= (8 + 2) \times 8 + 8 \times 8 + 8 \times 1$ and bias $= 8 + 8 + 1$). So the corresponding dynamic controller $C_{\text{controller}} \in \mathbb{R}^{168 \times 1 \times 1}$.

In addition to the abovementioned dynamic controller branch and global feature map branch, we adopt the center point prediction branch, the ship size regression branch, and the offset regression branch of Center Net. These branches generate a heatmap of center point estimation $F_{hm} \in [0, 1], \mathbb{R}^{H \times W \times 1}$, ship size prediction map $F_{wh} \in \mathbb{R}^{H \times W \times 2}$ and offset prediction map $F_o \in \mathbb{R}^{H \times W \times 2}$, respectively.

The pixel-wise focal loss [45] for ship center prediction is calculated as follows:

$$L_{hm} = -\frac{1}{N}\sum_{xy}\begin{cases}(1-\widehat{y}_{xy})^{\alpha}log(\widehat{y}_{xy}) & y_{xy}=1\\(1-\widehat{y}_{xy})^{\beta}\widehat{y}_{xy}^{\alpha}log(1-\widehat{y}_{xy}) & \text{otherwise}\end{cases} \tag{7}$$

where α and β are the hyper-parameters of focal loss, we adopt the settings in Center Net with $\alpha = 2$, $\beta = 4$. N represents the number of ship targets in a single image, which is cable to normalize the positive samples in the image. Y_{xy} and $\widehat{Y}_{xy}$ denote the elements of the ground truth map and the center estimation heatmap, respectively.

At the predicted center point of each ship target, the length and width are regressed by the size regression branch. L1 loss is adopted to calculate the regression loss. The calculation process can be described by

$$L_{wh} = -\frac{1}{N}\sum_{k=1}^{N}(|w_{kt} - w_k| + |h_{kt} - h_k|) \tag{8}$$

where w_{kt} and h_{kt} represent the actual length and width of the k_{th} ship target, w_k and h_k are the predicted length and width.

The prediction maps are downsampled by four times compared to the original input SAR image, introducing a certain discrete error when calculating the center coordinates. We adopt the offset regression branch to compensate these errors and L1 loss is used for this branch

$$L_o = -\frac{1}{N}\sum_{k=1}^{N}\left(\left|O_{xk} - \left|\frac{X_{kt}}{r} - x_k\right|\right| + \left|O_{yk} - \left|\frac{Y_{kt}}{r} - y_k\right|\right|\right) \tag{9}$$

where O_{x_k} and O_{y_k} represent the center point deviation of the k_{th} ship predicted by the offset regression branch. X_{kt} and Y_{kt} are the coordinates of the center point in the original image, r represents the downsampling ratio, which is 4 in our method. x_k and y_k are the horizontal and vertical coordinates of the center point of the ship predicted above. The supervision is only conducted on each ship center.

The mask branch uses the CD loss designed for ships, which will be described in detail in the following part.

F. Centroid-Distance-Based Loss

Dice loss is first proposed in V-Net [33] and widely used in medical image segmentation. The calculation of dice loss can be expressed by

$$L_{\text{Dice}} = 1 - \frac{2\,|X \cap Y| + 1}{|X| + |Y| + 1} \tag{10}$$

where $|X \cap Y|$ represents the intersection between X and Y and equals to the dot product between the predicted map and the ground truth. $|X|$ and $|Y|$, respectively, represent the number of elements in X and Y. The coefficient of the numerator is 2 due to the repeated calculations of common elements between X and Y. To avoid the case where the denominator is 0 and reduce overfitting during training, we add factor 1 to the numerator and denominator.

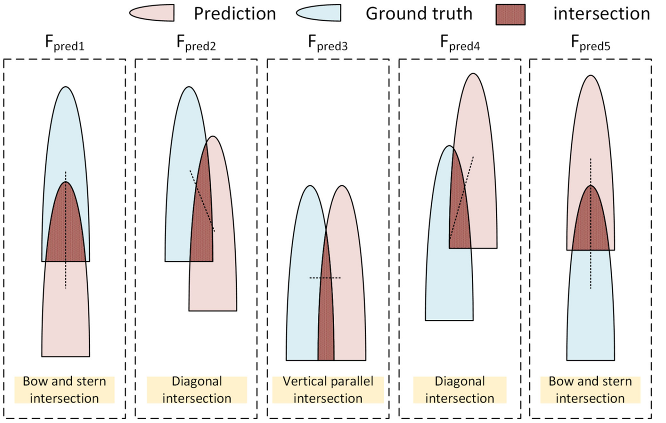


Fig. 6. F_{pred} represents the prediction ship. From left to right, several positional relationships such as the intersection of the bow and the stern, the intersection of the diagonals, and the intersection of the vertical and parallel are shown, respectively.

However, dice loss merely calculates the similarity between the predicted map and the ground truth based on the degree of overlap. From the form of dice loss, we can conclude that once the overlap degree of predicted map and ground truth is the same, the result of the dot multiplication keeps invariant, leading to a constant loss function value.

Assume that there is only a single target in the ship image shown in Fig. 6. Where $F_{\text{pred}} \in [0, 1]$ represents the predicted ship map, $F_{GT} \in \{0, 1\}$ represents the ground truth. Given that the results of the dot product under these positional relationships are identical

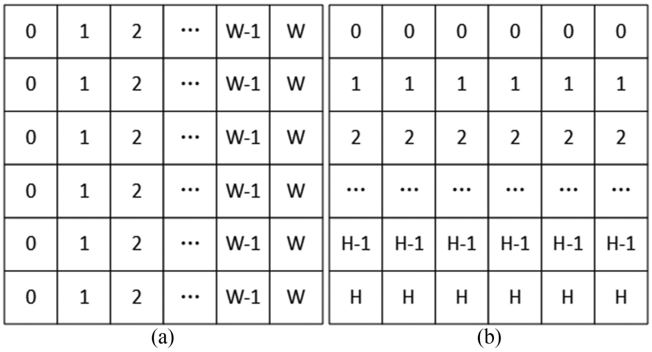
$$\sum_{xy} y_{xy}^{p} y_{xy}^{gt} = S_{\text{const}},$$

$$y_{xy}^{p} \in F_{\text{pred}i}, y_{xy}^{gt} \in F_{gt}, i \in [1, 5] \tag{11}$$

where y_{xy}^{p} represents the element value at the position (x, y) in the i_{th} prediction map, y_{xy}^{gt} represents the element value at the position (x, y) in the ground truth, and S_{const} is a constant. At this time, dice loss is incapable of making effective distinguish between these relationships.

During the process of ship training, we are more inclined to F_{pred3}, when the predicted ship is vertical parallel to the true ship, which is much closer to the expected training results. Dice loss is not qualified to meet the purpose that a smaller loss should be assigned to this positional relationship compared to the intersection of diagonals and the intersection of the bow and the stern. Therefore, in the case of vertical parallel intersection, indicators that are superior to other situations should be designed to build a corresponding loss function.

As shown by the dotted line in Fig. 6, the distance between the centroids of the ship is different in these cases. For the ships with large aspect ratios, the smallest center distance occurs with the positional relationship in F_{pred3}. Therefore, the loss function can be weighted by the distance of the center points. Since only the positional relationship of the ship is considered here, we use



0	1	2	···	W-1	W
0	1	2	···	W-1	W
0	1	2	···	W-1	W
0	1	2	···	W-1	W
0	1	2	···	W-1	W
0	1	2	···	W-1	W

(a)

0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
···	···	···	···	···	···
H-1	H-1	H-1	H-1	H-1	H-1
H	H	H	H	H	H

(b)

Fig. 7. Grid point graphs G_X and G_Y, where H and W are the width and length of the image, respectively.

the centroid to represent the center of the ships. The ground truth map is a simple binary graph, the center of gravity is the same as the centroid; $F_{\text{pred}3} \in [0, 1]$ stands for the predicted map, which must be binarized first to calculate the centroid. The calculation for the binarized prediction map $F_{\text{pred}_{\text{bin}}}$ can be expressed by

$$y_{xy}^{p_{\text{bin}}} = \begin{cases} 1 & y_{xy}^{p} > \text{Threshold} \\ 0 & y_{xy}^{p} \leq \text{Threshold} \end{cases},$$
$$y_{xy}^{p_{\text{bin}}} \in F_{\text{pred}_{\text{bin}}}, y_{xy}^{p} \in F_{\text{pred}} \tag{12}$$

where $y_{xy}^{p_{\text{bin}}}$ represents the element value in the (x, y) position of the binarized prediction map $F_{\text{pred}_{\text{bin}}}$. Threshold is set as 0.5 in this article.

For the ground truth map and the binarized prediction map, obtaining the centroid coordinates by point-by-point calculation will greatly reduce the calculation efficiency. Therefore, we design two grid point graphs G_X and G_Y, as shown in Fig. 7.

In this article, the length and width of the ground truth map and the binarized prediction map are the same and equal to 128. Their centroid coordinates can be easily calculated using two grid point maps. The calculation formula can be described as follows:

$$B_{X_{\text{pred}}} = \sum_{XY} \left(X_{xy}^{P_{\text{bin}}} y_{xy}^{G_X}\right)$$
$$B_{Y_{\text{pred}}} = \sum_{XY} \left(Y_{xy}^{P_{\text{bin}}} y_{xy}^{G_Y}\right)$$
$$B_{X_{GT}} = \sum_{XY} \left(X_{xy}^{GT} y_{xy}^{G_X}\right)$$
$$B_{Y_{GT}} = \sum_{XY} \left(Y_{xy}^{GT} y_{xy}^{G_Y}\right) \tag{13}$$

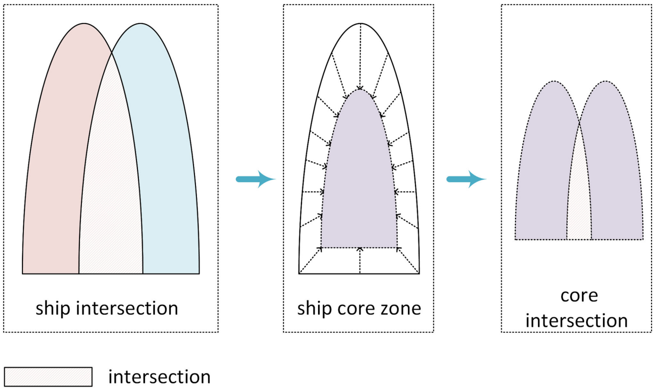


Fig. 8. Diagram of the central area intersection. From left to right, the intersection of the predicted ship map and the true ship map, the principle of the central area generation and the intersection of the central area are shown, respectively.

where $(B_{X_{\text{pred}}}, B_{Y_{\text{pred}}})$ stands for the centroid coordinates of the binarized prediction map, $(B_{X_{GT}}, B_{Y_{GT}})$ stands for the centroid coordinates of the ground truth map, $(X_{xy}^{P_{\text{bin}}}, Y_{xy}^{P_{\text{bin}}})$ and $(X_{xy}^{GT}, Y_{xy}^{GT})$ represent the coordinates in the (x, y) position of the binarized prediction map and the ground truth, respectively. $(y_{xy}^{G_X}, y_{xy}^{G_Y})$ represent the element value in the (x, y) position of the two grid point graphs G_X and G_Y, respectively. The centroid distance and the normalized centroid distance between the predicted ship and the true ship can be expressed as

$$D_{\text{bary}} = \sqrt{(B_{X_{\text{pred}}} - B_{X_{GT}})^2 + (B_{Y_{\text{pred}}} - B_{Y_{GT}})^2}$$
$$D_{\text{bary}_{\text{norm}}} = \frac{D_{\text{bary}}}{\sqrt{2}W} \tag{14}$$

where W represents the width of the predicted map. Although the loss function weighted by the normalized centroid distance can generate a small loss for the vertical parallel intersection, representing the position of the ship by only one point still fails to make full use of the ship's geometry. Therefore, we further propose the concept of the centroid region, using the central region to represent the entire ship. When the degree of overlap between the predicted ship and the true ship is the same, the positional relationship with a larger overlap of the centroid regions will be given a smaller loss. Therefore, the loss function can be further weighted by the degree of overlap of the centroid region of the ship as shown in Fig. 8.

This article introduces a corrosion algorithm to obtain the centroid region of the ship target. The algorithm consists of multiple corrosion processes, the single execution can be expressed as follows:

$$F_{\text{min}} = \text{minpool2}d(F_{in}) = (-1) \cdot \text{maxpool2}d(-F_{\text{in}})$$
$$F_{\text{contour}} = \text{ReLU}(\text{maxpool2d}(F_{\text{min}}) - F_{\text{min}})$$
$$F_{\text{out}} = \text{ReLU}(F_{\text{in}} - F_{\text{contour}}) \tag{15}$$

where F_{in} represents the binarized prediction map, in which the ship target equals to 1 and the background equals to 0. To obtain the smooth ship contour F_{contour}, the prediction map is processed by a serial of minimum pooling and maximum pooling operations. Using the difference between the binarized prediction map and the contour map, the centroid region can be retained. Smaller centroid region will be gradually generated to characterize the geometry and position of the ship through multiple successive corrosions. Fig. 9 shows the influence of the convolution kernel size and the number of the corrosion process on the generated region.

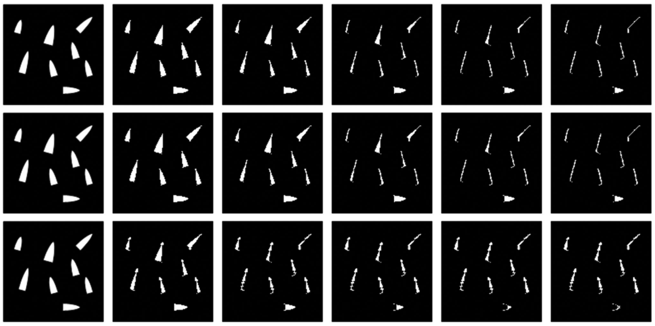


Fig. 9. Corrosion results. The topmost row shows the results for the minimum pooling and the maximum pooling with convolution kernel 3×3 and 5×5, the second row shows the results for convolution kernel 3×3 and 7×7 and the third row shows the results for convolution kernel 5×5 and 7×7. (a) Original image. (b) After one corrosions. (c) After two corrosions. (d) After three corrosions. (e) After four corrosions. (f) After five corrosions.

The topmost and second rows prove that the convolution kernel size of the maximum pooling does not significantly affect the experimental results. However, the ship target narrows to a line after five corrosions. Comparing to the others, the third row where the convolution kernel sizes of minimum pooling and maximum pooling are 5×5 and 7×7, respectively, performs worse, with ship target losing geometric shape and centroid region composing of interval points. Column (f) illustrates a smaller kernel size of minimum pooling contributes to better corrosion results. Therefore, in order to prevent the ship losing its geometric shape during the training process and reduce the parameters quantity as much as possible, we select the convolution kernel size setting in the first row and perform four consecutive corrosions. The binarized predicted ship map and the ground truth map after corrosion are represented by $F_{C_{\text{pred}}}$ and $F_{C_{GT}}$. The degree of overlap between them is represented by the intersect over union (IOU) of the two areas

$$IOU_c = \frac{|F_{C_{\text{pred}}} \cap F_{C_{GT}}|}{|F_{C_{\text{pred}}} \cup F_{C_{GT}}|} = \frac{|F_{C_{\text{pred}}} \cap F_{C_{GT}}|}{|F_{C_{\text{pred}}}| + |F_{C_{GT}}| - |F_{C_{\text{pred}}} \cap F_{C_{GT}}|}. \quad (16)$$

Since the IOU metric is used as the evaluation index in instance segmentation, our algorithm uses the centroid distance and the degree of overlap between central regions to weight the IOU loss. Similar to dice loss, the IOU loss for the predicted target X and the true target Y can be calculated as

$$L_{\text{iou}} = 1 - \frac{|X \cap Y|}{|X \cup Y|} = 1 - \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}. \quad (17)$$

In order to prevent the denominator from being zero and reduce overfitting, we add a constant 1 to the numerator and denominator. Therefore, for the predicted ship map and the ground truth map containing N ship targets, loss function weighted by centroid distance and the overlap degree of the central region can be expressed as

$$L_{\text{iou}} = \left(1 - \frac{|F_{C_{\text{pred}}} \cap F_{C_{GT}}| + 1}{|F_{C_{\text{pred}}} \cup F_{C_{GT}}| + 1}\right)$$
$$L_{\text{mask}} = \left(1 + \beta_1 \sum_{i=1}^{N} \frac{D_{\text{bary}_i}}{N} + \beta_2 IOU_C\right) L_{\text{iou}} \quad (18)$$

where β_1 and β_2 stand for the two weights, D_{bary_i} is the centroid distance between the ith predicted ship target and true ship target. In the experiment, we choose $\beta_1 = 6$ and $\beta_2 = 4$ empirically, which can obtain best experimental results. The joint loss of the anchor-free instance segmentation predictor is the sum of the above four entities

$$L_{\text{total}} = \alpha_1 L_{hm} + \alpha_2 L_{wh} + \alpha_3 L_o + \alpha_4 L_{\text{mask}}. \quad (19)$$

In this article, the scale of each part of the loss function is quite different to the others. The loss of each part needs to be unified to the same order of magnitude [46]. We conduct experiments and set the hyper-parameters in the loss function to $\alpha_1 = 0.5$, $\alpha_2 = 0.05$, $\alpha_3 = 1$, $\alpha_4 = 7$.

III. Experiments

In this section, we conduct experiments on SSDD to evaluate the effectiveness of our method. First, the SSDD dataset used in this article and the experimental setting will be introduced in detail. Then, the evaluation metrics adopted in the evaluation is described. Next, we show the visualization results of comparison experiments to verify the effectiveness of our method. Finally, the results of ablation experiments of each module and network width comparison experiment results are given.

A. Dataset Description and Experimental Settings

In this article, experiments are carried out on SSDD. The detailed information of SSDD is shown in Table II. SSDD contains 1160 multiresolution, multipolarization, and multiscene SAR images. Each image contains 2.12 ship targets on average. Multiresolution coverage of 1, 3, 5, 7, 10, and 15 m ensures better adaptability of the training model. The training and the test sets are divided in a ratio of 8:2 in this experiment, containing 928 SAR images and 232 SAR images, respectively [47]. The senses include inshore and offshore scenes. Compared to the offshore scenes, the inshore scenes are influenced more by land clutters, leading to much harder segmentation. In order to validate the segmentation performance of the proposed model in different scenes, the test set is divided into inshore and offshore parts, containing 39 and 193 SAR images, respectively.

In the training process, we randomly initialize the parameters of the feature extraction network Ghost Net. Adaptive moment estimation (Adam) [48] is adopted as the training optimizer, the weight decay of which is set to 0.0005. The initial learning rate is set to 1.25×10^{-4}, where the learning rate drops by 10 times at the 100th epoch, the number of small batches of random gradient descent is set to 4 and a total of 140 epochs are trained. The experiments are implemented using the deep

TABLE II
DETAILED INFORMATION OF SSDD

Satellite	Polarization	Imaging Mode	Average Size	Quantity	Location	Scene
RadarSat-2,Sentinel-1,TerraSAR-X	HH,VV,HV,VH	Interferometric, wide swath, Spotlight, Strip	481×331	1160	Yantai, China, Visakhapatnam, India	Inshore, Offshore

learning framework Pytorch [49], and the comparison algorithms except CenterMask are conducted under the MMDet framework [50], CenterMask is conducted under the Detectron2 framework [51]. All of our experiments are carried out on a platform configured with Ubuntu 18.04 system, 16 G memory, and Tesla P100 GPU.

B. Evaluation Metrics

For instance segmentation of SAR images, mask IOU is defined by the overlap ratio of the predicted mask and the ground truth mask, which is used to evaluate the accuracy of the instance segmentation. The calculation formula of mask IOU is as follows:

$$\mathrm{IOU}_{\mathrm{mask}} = \frac{\mathrm{mask}_{\mathrm{pred}} \cap \mathrm{mask}_{gt}}{\mathrm{mask}_{\mathrm{pred}} \cup \mathrm{mask}_{gt}} \tag{20}$$

where $\mathrm{mask}_{\mathrm{pred}}$ and mask_{gt} stand for the predicted mask and the ground truth mask, respectively.

The current common dataset evaluation metrics are Pascal Visual Object Classes (Pascal VOC) [52] and Microsoft Common Objects in Context (MS COCO) [53]. The latter is adopted in this article to quantitatively evaluate the performance of the models. MS COCO's evaluation criteria are abundant and comprehensive, and targets with various sizes in an identical category are calculated separately due to their wide disparity in AP. Pascal VOC's mAP calculation standard is based on the IOU threshold of 0.5, while MS COCO's mAP calculation standard contains more detailed IOU threshold settings, such as AP, AP_{50}, and AP_{75}. AP_{50} represents the calculation under the IOU threshold of 0.5. AP_{75} more strictly represents the calculation under the IOU threshold of 0.75. AP is the primary challenge metric and is averaged across all 10 IOU thresholds from 0.5 to 0.95 with the step of 0.05. The model's ability to segment multiscale targets is evaluated using AP_S, AP_M, and AP_L. These three indicators correspond to small targets with an area less than 32^2 pixels, medium targets with area between 32^2 pixels and 64^2 pixels and large targets with an area exceeding 64^2 pixels, respectively. A larger AP value indicates a higher prediction accuracy of the instance mask and a better instance segmentation effect. To comprehensively evaluate the model's performance, the precision-recall (PR) curve is also introduced. The more areas the PR curve covers, the better the model performs.

Besides, we adopted some metrics such as model size, floating point operation (FLOPs), and parameters to evaluate the time complexity and space complexity of the model. Model size refers to the size of storage space required to save the model during training. In actual calculation, model size includes network architecture information and optimizer information, in addition to the amount of parameters. FLOPs can be used to measure the time complexity of the model. It can solve the problem that the processing time of different models on different hardware platforms cannot be directly compared. Parameters refer to the total weight of all parameterized layers, and it is only related to the size of the convolution kernel, the number of channels, and the number of layers, representing the model's space complexity. The more parameters, the more training data is needed to avoid overfitting.

C. Comparison With Other Methods

To verify the effectiveness of our method, we choose Mask R-CNN, Cascade Mask R-CNN, GC Net [46], DCN [54], Yolact, and CenterMask for comparison, which have achieved outstanding performance in the field of instance segmentation. These DCNN-based methods are introduced as follows.

1) Mask R-CNN: Mask R-CNN is a classic two-stage deep learning instance segmentation method based on Faster R-CNN. A mask branch is employed to predict the mask for each ROI. In addition, ROI pooling is replaced by ROI Align.

2) Cascade Mask R-CNN: Cascade Mask R-CNN is a two-stage deep learning instance segmentation algorithm combining the characteristics of Mask R-CNN and Cascade R-CNN. Each Cascade structure contains a parallel mask branch to generate masks pixel by pixel.

3) GC Net: GC Net absorbs the advantages of nonlocal network (NLNet) and squeeze excitation network (SENet), providing a simple, fast and effective method for global context modeling.

4) DCN: Deformable convolution is proposed in DCN, the scale and direction of which can be changed to acquire receptive fields of various scales and shapes.

5) Yolact: Yolact is a single-stage full-convolution real-time instance segmentation algorithm based on anchor boxes. It achieves instance segmentation independent of the feature location processing steps such as ROI pooling and ROI Align.

6) CenterMask: Centermask is an efficient and real-time instance segmentation method based on anchor-free and proposal-free one stage object detector FCOS. A spatial attention module is added to the mask branch to focus on meaningful pixels.

Table III shows the quantitative instance segmentation performance of different methods in the inshore and offshore scenes

TABLE III
INSTANCE SEGMENTATION PERFORMANCE OF DIFFERENT METHODS

Model	Backbone	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Mask R-CNN	ResNet-50	offshore	0.636	0.944	0.807	0.613	0.736	0.600
		inshore	0.419	0.762	0.424	0.456	0.360	0.000
Cascade Mask R-CNN	ResNet-50	offshore	0.641	0.946	**0.823**	0.623	0.735	0.700
		inshore	0.443	0.762	0.510	0.474	0.401	0.000
GC Net	ResNet-50	offshore	0.634	0.953	0.799	0.616	0.719	**0.800**
		inshore	0.418	0.763	0.473	0.464	0.374	0.000
DCN	ResNet-50	offshore	0.629	0.915	**0.823**	0.605	0.729	**0.800**
		inshore	0.426	0.756	0.464	0.457	0.390	0.000
Yolact	ResNet-50	offshore	0.609	0.968	0.747	0.591	0.707	0.700
		inshore	0.406	0.740	0.431	0.452	0.348	0.000
CenterMask	ResNet-50	offshore	0.643	0.975	0.799	**0.633**	0.731	0.550
		inshore	0.431	0.766	0.491	0.410	0.287	0.000
Ours	1.5xGhost Net	offshore	**0.648**	**0.981**	0.799	0.623	**0.743**	0.600
		inshore	0.460	0.806	0.483	0.479	0.457	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

of SSDD. It can be seen from Table III that our method leads other methods by a large margin, both in inshore or offshore scenes. Yolact has the worst instance segmentation performance compared to other methods. Surprisingly, the offshore performance of CenterMask is relatively better compared to other anchor-based methods, with AP close to Cascade Mask-RCNN. And CenterMask has more advantages in the instance segmentation of small targets, achieves topmost precision among comparative methods. The instance segmentation accuracy of Cascade Mask R-CNN in inshore scenes is significantly better than other comparative methods, but there is still a 1.7% gap with our method. To be specific, the performance of Cascade Mask R-CNN is close to ours when the IOU threshold is 0.75, while our method achieves a 5% advantage for the wider IOU threshold of 0.5 and middle size targets segmentation. GC Net and DCN achieve better performance than Yolact, but the overall performance is relatively poor compared to Mask R-CNN and Cascade Mask R-CNN. The reason why Yolact perform worse than other two-stage methods is that the parallel anchor-based branches may cause misclassification or inaccurate positioning of the bounding box. For instance, when there are multiple overlapping instances in a certain position, Yolact may not be able to locate them through the prototype mask it has learned. Therefore, to completely eliminate the influence of the anchor, we use an anchor-free instance segmentation predictor. In order to take advantage of the extracted feature information, we enhance the shallow features and transform the task-specific features in channel and spatial dimensions. As a result, compared to others, our method makes a balance between multiscale ship targets instance segmentation.

The PR curves are illustrated in Fig. 10 to comprehensively show the instance segmentation performance of different methods in different scenes. It can be observed from Fig. 10(a) and (b) that the PR curve of our method is away from the other methods, indicating that its instance performance is the best no matter it is in inshore or offshore scene. Among the six comparative methods, the PR curve of CenterMask seems to cover more areas than the others. The offshore PR curve of the DCN covers the least area, illustrating that the deformable convolution does not help improve the offshore instance segmentation performance. The PR curve of Yolact shows this method is suitable for instance segmentation in offshore scenes. In summary, the results verify the superior performance of our method.

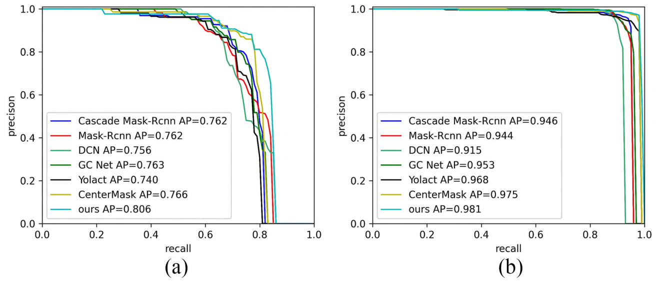


Fig. 10. PR curves of different methods in different scenes. (a) PR curves for inshore scenes. (b) PR curves for offshore scenes.

In addition, Fig. 11 compares the parameters quantity, model size, and the FLOPs of above methods. We are sorry that CenterMask project under the Detectron2 framework does not support the computation of FLOPs and parameters quantity, so only model size is compared for CenterMask. As shown in the figure, Cascade Mask R-CNN far exceeds other comparison methods in the number of parameters (245.49 M), model size and FLOPs because of multiple IOU thresholds and postprocessing of the regression box. GC Net adds a global context module on the basis of Mask R-CNN, resulting in a slight increase in model size and FLOPs. Compared to Mask R-CNN, the number of the parameters of DCN is reduced to a certain extent due to the adaption of a smaller convolution kernel to obtain multiple receptive fields. Among the six comparison methods, Yolact and CenterMask have relatively smallest model sizes and require fewest FLOPs, showing better efficiency. However, our

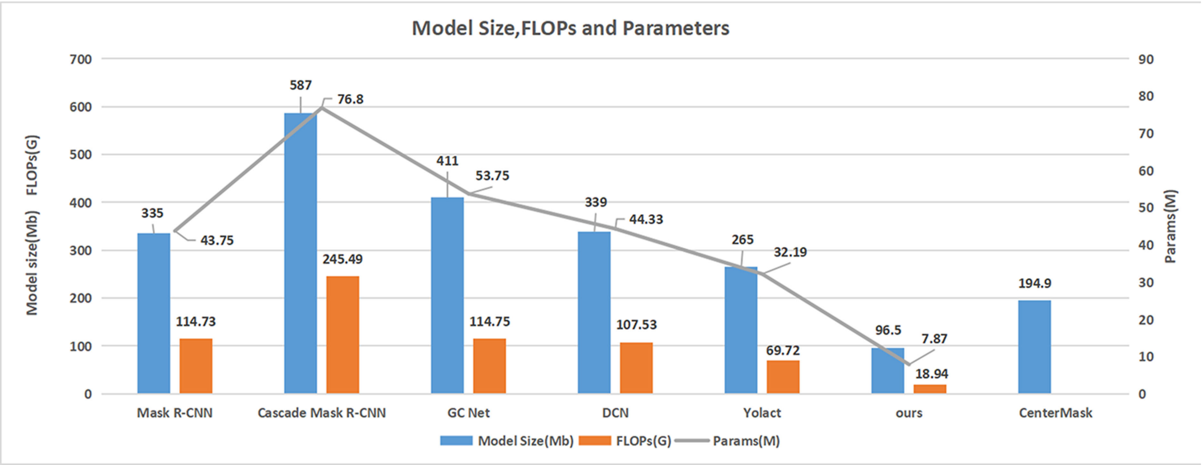


Fig. 11. Comparison the number of parameters, model size, and FLOPs.

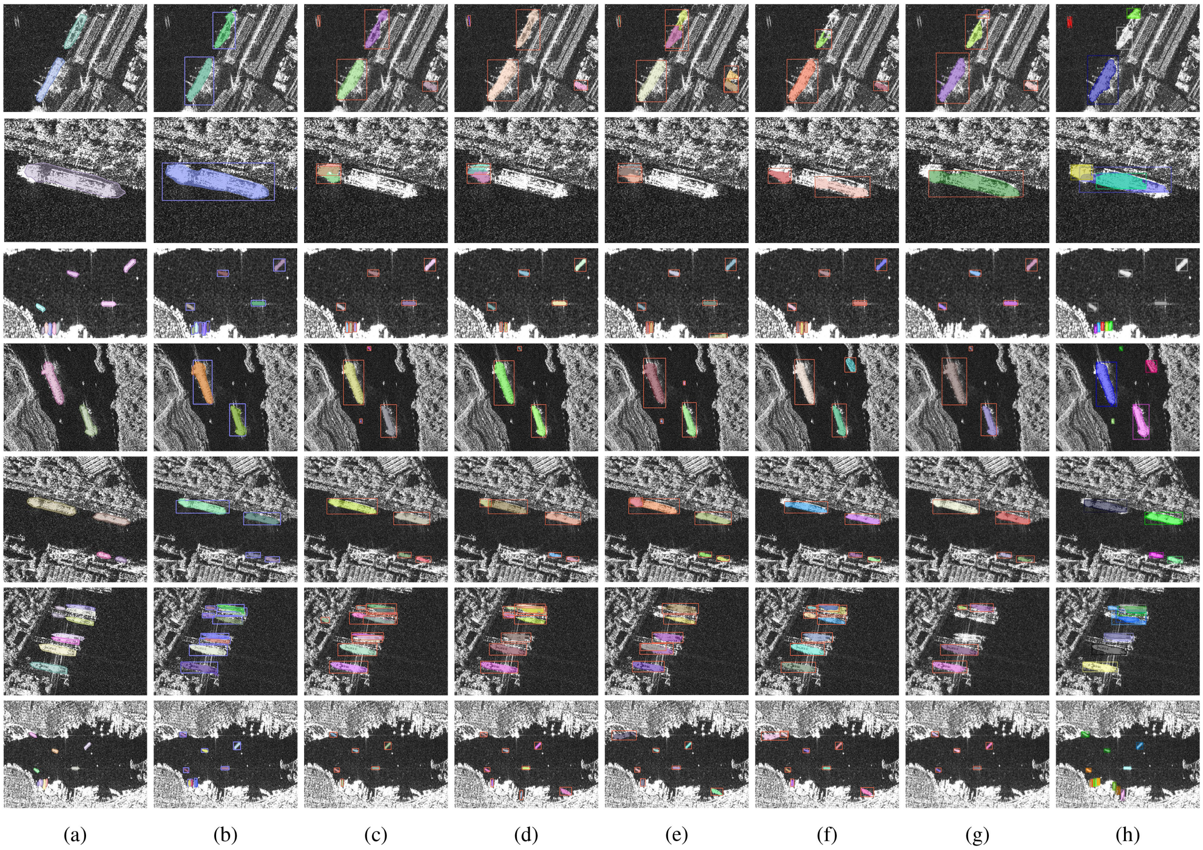


Fig. 12. Instance segmentation results of different methods in inshore scenes. (a) Ground truth. (b) Our method. (c) Cascade Mask R-CNN. (d) Mask R-CNN. (e) DCN. (f) GC Net. (g) Yolact. (h) CenterMask.

method still leads them by a large margin. The number of the parameters in our methods is only one fourth of that in Yolact, and the required FLOPs is far fewer than Yolact. It demonstrates the superior efficiency of our method. In short, the experiment results show that our method is efficient in computation and light in model size, thanks to the adopted lightweight extraction network, tree-like feature extraction structure, and center point based segmentation predictor.

In order to visually compare our method with the other methods, in Figs. 12 and 13, several SAR images for comparison in inshore and offshore scenes are, respectively, given. Columns (a)–(h) represent the ground truth, the segmentation

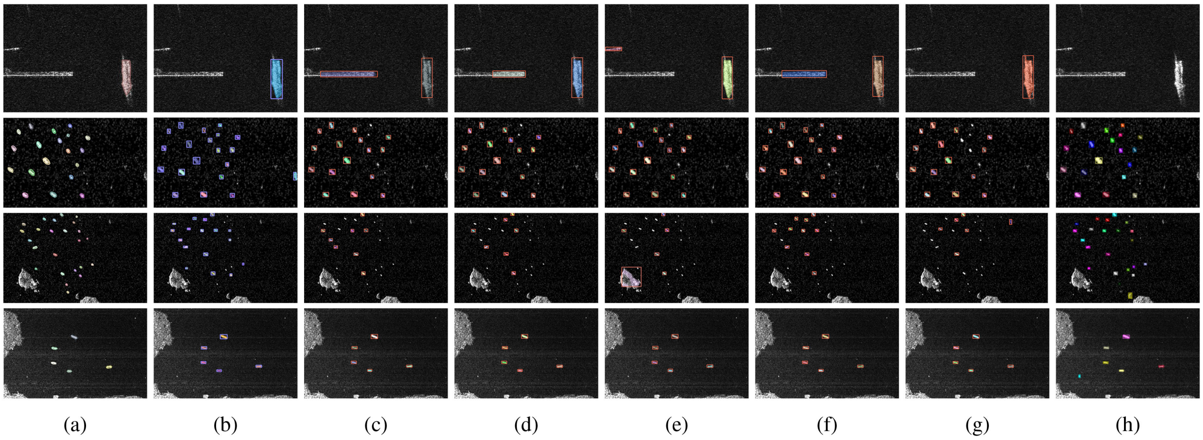


Fig. 13. Instance segmentation results of different methods in offshore scenes. (a) Ground truth. (b) Our method. (c) Cascade Mask R-CNN. (d) Mask R-CNN. (e) DCN. (f) GC Net. (g) Yolact. (h) CenterMask.

results of the method in this article, the Cascade Mask R-CNN segmentation results, the Mask R-CNN segmentation results, the DCN segmentation results, the GC Net segmentation results, the Yolact segmentation results, and the segmentation results of CenterMask, respectively.

From Figs. 12(b) and 13(b), we can see that the proposed method segments ship targets more accurately, not only the densely distributed ship targets with relatively large aspect ratio in inshore, but also the scattered distribution ship targets of a small scale in offshore scenes. There are fewer false alarms and missed targets, to be specific, there is a false alarm on the third row in Fig. 13(b). And the generated mask is smoother compared to other methods. It can be seen from Figs. 12(c) and 13(c) that Cascade Mask R-CNN has more false alarms in inshore scenes and misses some targets in offshore scenes. In the third row in Fig. 13(c), many missed targets occurs. The instance segmentation results of Mask R-CNN in Figs. 12(d) and 13(d) show more false alarms in inshore scenes, and the land clutter in the second row in Fig. 13(d) is mistaken as a ship target. For the results of DCN in Figs. 12(e) and 13(e), many false alarms occur in inshore scenes, and densely distributed ship targets are recognized as a whole. In the second and third row and in Fig. 12(d), there are about eight missed targets and two obvious false alarms. In Figs. 12(f) and 13(f), for GC Net, the instance segmentation performance is unsatisfactory, several medium-sized targets are segmented to multiple targets, small-sized targets are mostly missed. The instance segmentation results of Yolact shows fewer false alarms in inshore scenes, and the generated mask is smoother than the other two-stage methods, mainly because of the abandon of repooling operation. The instance segmentation performance of CenterMask showed in column (h) is satisfactory, especially for small ship targets. The figure on the third row in Fig. 12 contains five scattered distribution ship targets on the coastline, the instance segmentation results show that only our methods and CenterMask can obtain the five targets' contour and position precisely. It is mainly because of the proposed spatial attention guided mask, which can help the mask predictor to focus on informative pixels but also suppress noise. To summarize, we achieve more accurate instance segmentation results and generate smoother masks than other methods in both inshore and offshore scenes, which proves that the CD loss can better guide the regression of the network and verifies the effectiveness of our proposed algorithm.

D. Ablation Experiments

To illustrate the effectiveness of each module and quantitatively judge the improvement of each module, this article reports ablation experiments on CBAM, the DEM, the dynamic ED, and the CD loss.

Table IV shows the results of CBAM ablation experiment. By comparing the model parameters, the instance segmentation accuracy and other indicators before and after removing the CBAM, we can draw the following conclusions.

1) The instance segmentation accuracy in inshore and offshore is improved with the adoption of CBAM, increased by 1% and 3.5%, respectively.
2) CBAM improves the feature representation ability in the fusion process and the model's ability to segment multiscale targets.
3) CBAM has almost no influence on the number of parameters, with FLOPs and model size increased by 1.6 G and 2.5Mb, respectively.

Table V shows the ablation experiment results of the DEM. By comparing the model parameters, the instance segmentation accuracy and other indicators before and after removing the DEM, we can draw the following conclusions.

1) The instance segmentation accuracy whether it is in inshore or offshore is superior to the performance when the multiscale features are directly combined, showing that the DEM is beneficial to improve accuracy.
2) Although the model's ability to segment large-scale targets has been weakened to some extent, the instance segmentation capabilities of small and medium-sized targets have been effectively enhanced, indicating that the DEM can make full use of shallow feature information to balance the segmentation performance of multiscale targets.

TABLE IV
ABLATION EXPERIMENT ON CBAM

CBAM	**Model Size(Mb)**	**FLOPs(G)**	**Params(M)**	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	96.50	18.94	7.87	offshore	**0.648**	0.981	**0.799**	**0.623**	0.743	0.600
				inshore	0.460	0.806	0.483	0.479	0.457	0.000
	94.00	17.39	7.85	offshore	0.638	**0.985**	0.795	0.610	**0.748**	0.600
				inshore	0.425	0.800	0.420	0.457	0.389	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

TABLE V
ABLATION EXPERIMENT ON DEM

DEM	**Model Size(Mb)**	**FLOPs(G)**	**Params(M)**	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	96.50	18.94	7.87	offshore	**0.648**	**0.981**	**0.799**	**0.623**	**0.743**	0.600
				inshore	0.460	0.806	0.483	0.479	0.457	0.000
	89.70	16.61	7.47	offshore	0.628	0.977	0.755	0.602	0.730	**0.610**
				inshore	0.430	0.750	0.470	0.460	0.386	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

TABLE VI
ABLATION EXPERIMENT ON ED

ED	**Model Size(Mb)**	**FLOPs(G)**	**Params(M)**	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	96.50	18.94	7.87	offshore	**0.648**	**0.981**	**0.799**	**0.623**	**0.743**	0.600
				inshore	0.460	0.806	0.483	0.479	0.457	0.000
	94.20	17.01	7.84	offshore	0.632	0.979	0.780	0.605	0.738	0.600
				inshore	0.434	0.781	0.465	0.456	0.404	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

TABLE VII
ABLATION EXPERIMENT ON CD LOSS

CD Loss	**Model Size (Mb)**	**FLOPs(G)**	**Params(M)**	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
✓	96.50	18.94	7.87	offshore	**0.648**	**0.981**	**0.799**	**0.623**	**0.743**	0.600
				inshore	0.460	0.806	0.483	0.479	0.457	0.000
	96.50	18.94	7.87	offshore	0.622	0.977	0.749	0.593	0.736	0.600
				inshore	0.415	0.750	0.391	0.433	0.400	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

3) Since the feature map has been kept the same size as the input in the convolution, the model size, and FLOPs increase by 6.8 Mb and 2.3 G, respectively.

Table VI shows the results of the dynamic ED ablation experiment. By comparing the model parameters, the instance segmentation accuracy and other indicators before and after removing the dynamic ED, we can draw the following conclusions.

1) After removing the dynamic ED, the instance segmentation accuracy of the model dropped by 1.6% and 1.1%, respectively, in inshore scenes and offshore scenes.
2) The dynamic ED significantly improves the model's instance segmentation ability of medium-sized targets under high IOU thresholds, indicating that the module can effectively allocate task-specific features.
3) The dynamic ED has a very slight influence on the model size and the parameters quantity, one possible reason is the adoption of the depth separable convolution and efficient design of the module.

Table VII shows the results of ablation experiments on the CD loss. By comparing the model parameters and instance segmentation accuracy when using CD loss and dice loss, the following conclusions can be drawn.

1) The instance segmentation accuracy with CD loss is significantly superior to the performance of using dice loss, with the average accuracy increased by 3.0% and 9.3% in inshore and offshore scenes, respectively, when the IOU threshold is 0.75. This shows that for closely distributed inshore ships, the loss function designed in this article according to the ship's geometric shape and positional

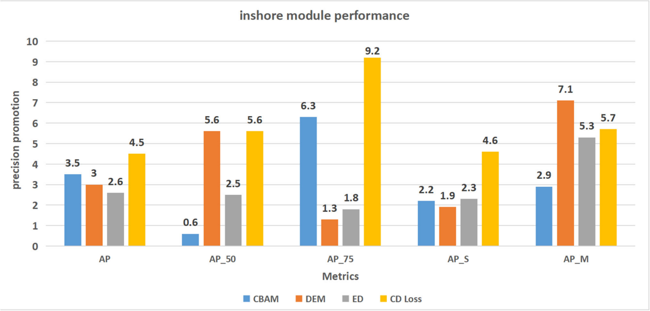


Fig. 14. Precision promotion of each module in inshore scenes.

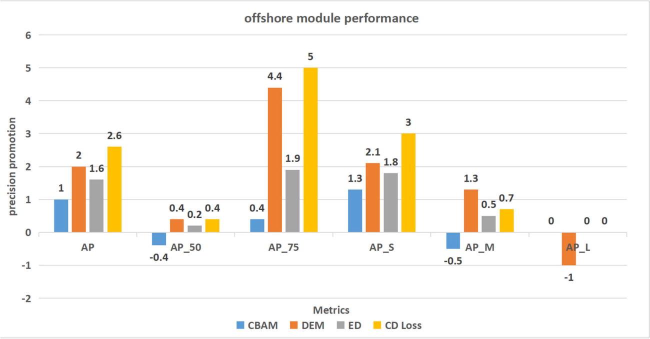


Fig. 15. Precision promotion of each module in offshore scenes.

relationship can be more effective in guiding the regression of training.

2) CD loss has nothing to do with the parameter quantity, model size, and FLOPs.

Figs. 14 and 15 compare the influence of each module on the instance segmentation performance in inshore and offshore scenes in the form of histograms. The following conclusions can be drawn.

1) The CD loss has the most significant improvement in the performance of the model in inshore or offshore scenes.
2) Compared to the other modules, CBAM slightly reduces the model's ability to segment small-scale targets in offshore scenes with small IOU thresholds but effectively improves the model's segmentation accuracy when the IOU threshold is greater than 0.75. Overall, it promotes the instance segmentation performance.
3) The DEM improves the instance segmentation accuracy of medium-sized and small targets. But the model's instance segmentation accuracy for large-scale targets drops by 1%, mainly because, in order to improve the performance of multiscale segmentation, only the three shallow features are enhanced, while the high-level features with larger receptive fields suitable for large targets instance segmentation are somewhat weakened.
4) The dynamic ED has a more balanced improvement in model performance, especially for inshore medium-scale targets. It proves that allocating features for different tasks can effectively help the regression of training goals.

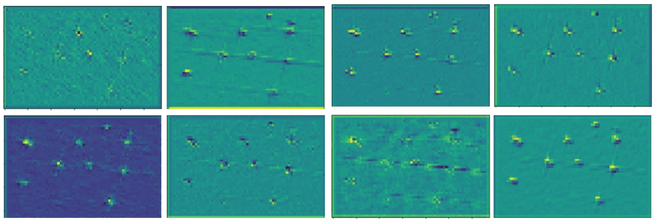


Fig. 16. Visualization results of features learned from different channels.

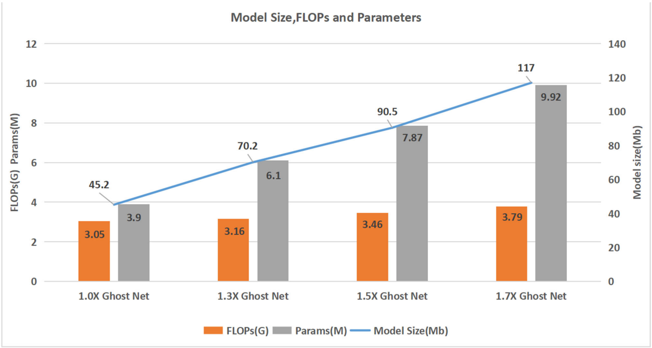


Fig. 17. Comparison of parameters, model size and FLOPs of different width.

E. Comparison Experiments of the Network's Width

Ghost Net uses a cost-efficient method to generate rich features and realizes the lightweight of the network. Ghost Net performs well in the field of image recognition. However, compared to the recognition task, instance segmentation tasks require more complex features, not only the category information, but also the position information, shape information, and so on. The width of the network has a key influence on the richness of features that each layer of the network can learn and handle. A wide network allows each layer to learn richer features, such as different directions and different frequencies. If the width is too small, the fitting ability of the network will be deficient and the instance segmentation performance will be degraded as a result. Fig. 16 shows the visualized features which are selected from the stage 4 output extracted from the backbone Ghost Net, more specifically, the output of the ghost cheap operation in third ghost bottleneck of stage 4. It can be seen that different channels have different emphases. Some focus on category information, and there is no obvious difference between ships, while some focus on characteristics such as target scale, and there are more obvious differences between the ships.

In order to extract rich features for instance segmentation tasks, we widen the number of the channels in the Ghost Net middle layer. However, a wider width may lead to a larger model size and a square increase in the parameters. In order to make a balance between the instance segmentation efficiency and the generalization ability of the network, this article reports comparative experiments on the model size, FLOPs, parameters quantity, and instance segmentation accuracy under different network widths.

Fig. 17 gives the results of the FLOPs, parameters quantity and model size under different widths. It can be seen from the figure

TABLE VIII
INSTANCE SEGMENTATION PERFORMANCE OF DIFFERENT WIDTH

Model	**Scene**	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
$1.0 \times GhostNet$	inshore	0.404	0.732	0.417	0.449	0.366	0.000
	offshore	0.638	0.987	0.768	0.616	0.728	0.617
$1.3 \times GhostNet$	inshore	0.422	0.785	0.446	0.466	0.370	0.000
	offshore	0.651	0.978	0.812	0.630	0.741	0.650
$1.5 \times GhostNet$	inshore	**0.460**	**0.806**	**0.483**	**0.479**	**0.457**	0.000
	offshore	0.648	0.981	0.799	0.623	0.743	0.600
$1.7 \times GhostNet$	inshore	0.431	0.799	0.433	0.461	0.396	0.000
	offshore	0.653	0.978	0.827	0.631	0.740	0.600

Bold items denote the optimal inshore values in the columns, the underlined items represent the optimal offshore values in the columns.

that the model size and parameters quantity have an exponential increase as the network is widened. The FLOPs also gradually increase as the width increases. Table VIII shows the influence of the network's width on the instance segmentation performance of the network. We can see that broadening the network by 1.3 times is already possible to obtain good enough instance segmentation performance in offshore scenes. A wider network does not make significant improvement in performance. When it is widened by 1.7 times, the instance segmentation accuracy for large targets is reduced to a certain extent. It shows that our method reaches the best generalization ability on SSDD in the interval of (1.3,1.7). The instance segmentation of inshore ship targets is more challenge. When the network is widened by 1.5 times, the model's instance segmentation accuracy for inshore medium-scale targets is about 6% higher than the accuracy with other widths. To conclude, as the width of the network increases, the instance segmentation performance of the network first increases due to the improvement of the fitting ability, and then degrades because of the degradation of the generalization ability. The efficiency drops as the increase of parameter quantity. To make a balance between performance and efficiency in inshore and offshore scenes for multiscale targets, we choose to widen the network by 1.5 times.

F. Loss Function Weights Selection

The algorithm proposed in our article contains multiple parallel branches, each of which have a corresponding loss function. In our article, the scale of each part of the loss function is quite different to the others. We follow the principle in [46] to design the loss function. The loss of each part needs to be unified to the same order of magnitude to make sure that the loss of each part of the training process has roughly the same convergence speed, which can avoid the loss of the smaller gradient from being dominated by the loss of the larger gradient and therefore enhance the generalization of the model. In our article, the mask loss function is described as

$$L_{\text{mask}} = \left(1 + \beta_1 \sum_{i=1}^{N} \frac{D_{\text{bary}_i}}{N} + \beta_2 IOU_C\right) L_{\text{iou}}. \tag{21}$$

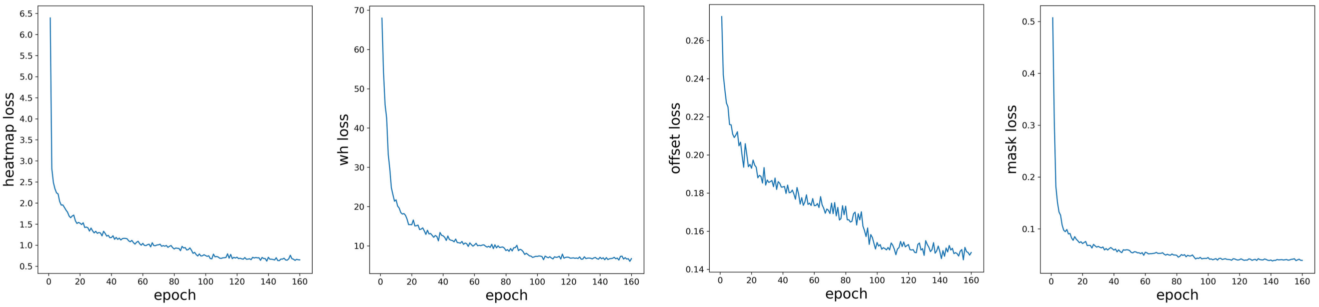
During the training process, the ratio of the values of $\sum_{i=1}^{N} \frac{D_{\text{bary}_i}}{N}$ and IOU_C is approximately $2:3$, to balance the influence of the centroid distance and the degree of overlap of the central area on the mask loss function, we empirically designed two weights β_1 and β_2, which have a ratio of $3:2$. Actually, in our algorithm, the final mask loss $\alpha_4 L_{\text{mask}}$ is jointly controlled by α_4 and the ratio of β_1 and β_2. Once the ratio is fixed, we can adjust α_4 to change the order of magnitude of mask loss in the overall losses. We selected $\beta_1 = 6$ and $\beta_2 = 4$ in our algorithm.

The joint loss of the anchor-free instance segmentation predictor is the sum of 4 entities, as (1) shows. To select the values of α_1, α_2, α_3, and α_4, the curves of the four losses over different iterations are drawn in Fig. 18 to denote the heatmap loss, weight and height (wh) loss, offset loss, and mask loss, respectively.

As the iteration increases, the heatmap loss drops from around 6.5 to 0.7; whereas, the wh loss drops from around 68 to 6; The magnitude of the offset loss is smallest, from around 0.28 to 0.15; and the mask loss drops from around 0.5 to 0.04. The magnitude of the wh loss is much larger than that of the other losses. The heatmap loss, wh loss, and mask loss guide the regression of classification, target size, and mask, respectively. To make sure the multipart tasks achieve balanced performance, we follow the aforementioned principle and loss curves to set the value ratio of α_1, α_2, and α_4 to be 0.5:0.05:7. The offset loss has relatively less impact on the performance of different tasks, so in our article, we empirically set the value ratio of α_1, α_2, α_3, and α_4 to be 0.5:0.05:1:7.

To verify whether the losses with our selected weights are capable of guiding the regression effectively, we conduct comparative experiments under different α_4, the weight of mask loss α_4 is set as 1, 3, 5, 7, 10, and 70, respectively. The instance segmentation performance in inshore and offshore scenes are shown in Table IX.

When α_4 is set as 7, the instance segmentation performance of our method is best. The AP of our method increases with α_4 increasing from 1 to 7, which demonstrates the effectiveness of our design principle. However, the AP drops sharply if α_4 doubles the order of magnitude, the reason might be that: as the weight of mask loss increases, the proportion of heatmap loss and wh loss drop, thus, the recognition accuracy and size regression accuracy decrease.



(a) (b) (c) (d)

Fig. 18. Multiple loss function curves. (a) Heatmap loss curve. (b) Weight and height loss curve. (c) Offset loss curve (d) Mask loss curve.

TABLE IX
INSTANCE SEGMENTATION PERFORMANCE UNDER DIFFERENT α_4

Model	Scene	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
$\alpha_4 = 1$	offshore	0.621	**0.982**	0.753	0.598	0.716	0.600
	inshore	0.377	0.758	0.371	0.419	0.320	0.000
$\alpha_4 = 3$	offshore	0.632	0.976	0.781	0.607	0.726	0.600
	inshore	0.408	0.751	0.421	0.432	0.400	0.000
$\alpha_4 = 5$	offshore	0.638	0.978	0.795	0.613	0.734	0.600
	inshore	0.423	0.761	0.441	0.452	0.395	0.000
$\alpha_4 = 7$	offshore	**0.648**	0.981	**0.799**	**0.623**	0.743	0.600
	inshore	0.460	0.806	0.483	0.479	0.457	0.000
$\alpha_4 = 10$	offshore	0.644	0.976	0.789	0.617	**0.747**	**0.700**
	inshore	0.435	0.786	0.468	0.449	0.438	0.000
$\alpha_4 = 70$	offshore	0.625	0.973	0.783	0.601	0.729	0.192
	inshore	0.357	0.662	0.380	0.403	0.293	0.000

Bold items denote the optimal offshore values in the columns, the underlined items represent the optimal inshore values in the columns.

IV. CONCLUSION

To overcome a number of key shortcomings of the existing SAR ship instance segmentation methods, this article presented a novel loss function and an anchor-free instance segmentation network based on the center point prediction. The main contributions are as follows.

1) We address the inability of dice loss to distinguish between different ship positional relationships under the same overlap degree, by proposing a loss function weighted by the centroid distance and the overlap degree of the central area named CD loss. CD loss efficiently combines geometric characteristics and positional relationships of the ship targets.
2) To improve the instance segmentation accuracy of multiscale targets, we widened the feature extraction network and further proposed the DEM to enhance the shallow features.
3) CBAM was introduced in the feature fusion process to extract salient features of different scales, thereby enhancing the ability of feature representation and suppressing clutter.
4) To solve the feature mismatching and training targets inconsistency problems caused by different tasks sharing features, we presented a dynamic ED to transform task-specific features and guide the regression of the network.
5) To reduce the parameters in DCNNs and improve the instance segmentation efficiency, we adopted a center point based instance segmentation predictor to generate instance masks end-to-end, without the need for preset anchors.

The experimental results on SSDD show that our method can achieve better instance segmentation accuracy and efficiency compared to other state-of-the-art algorithms, both in inshore or offshore scenes.

REFERENCES

[1] D. J. Crisp, “The state-of-the-art in ship detection in synthetic aperture radar imagery,” *Org. Lett.*, vol. 35, no. 42, pp. 2165–2168, Jul. 2004.

[2] F. Zhang, Y. Liu, Y. Zhou, Q. Yin, and H.-C. Li, “A lossless lightweight CNN design for SAR target recognition,” *Remote Sens. Lett.*, vol. 11, no. 5, pp. 485–494, 2020.

[3] Z. Yue *et al.*, “A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition,” *Cogn. Comput.*, vol. 13, no. 4, pp. 795–806, 2021.

[4] F. Gao, T. Huang, J. Sun, W. Jun, A. Hussain, and E. Yang, "A new algorithm of SAR image target recognition based on improved deep convolutional neural network," *Cogn. Comput.*, vol. 11, pp. 809–824, 2019.

[5] X. Sun, P. Wang, C. Wang, Y. Liu, and K. Fu, "PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 173, pp. 50–65, 2021.

[6] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 91–99, 2015.

[8] K. Chen *et al.*, "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4969–4978.

[9] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8759–8768.

[10] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6402–6411.

[11] H. Su, S. Wei, M. Yan, C. Wang, J. Shi, and X. Zhang, "Object detection and instance segmentation in remote sensing imagery based on precise mask R-CNN," in *Proc. IGARSS IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 1454–1457.

[12] H. Su *et al.*, "HQ-ISNet: High-quality instance segmentation for remote sensing imagery," *Remote Sens.*, vol. 12, no. 6, 2020, Art. no. 989.

[13] T. Zhang *et al.*, "Semantic attention and scale complementary network for instance segmentation in remote sensing images," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2021.3096185.

[14] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9156–9165.

[15] Q. He, X. Sun, Z. Yan, and K. Fu, "DABNet: Deformable contextual and boundary-weighted network for cloud detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3045474.

[16] F. Ma, F. Gao, J. Sun, H. Zhou, and A. Hussain, "Attention graph convolution network for image segmentation in big SAR imagery data," *Remote Sens.*, vol. 11, no. 21, 2019, Art. no. 2586.

[17] Y. Lee and J. Park, "Centermask: Real-time anchor-free instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13903–13912.

[18] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9626–9635.

[19] E. Xie *et al.*, "Polarmask: Single shot instance segmentation with polar representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12190–12199.

[20] X. Liu and X. Di, "Global context parallel attention for anchor-free instance segmentation in remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.3023124.

[21] Z. Huang, S. Sun, and R. Li, "Fast single-shot ship instance segmentation based on polar template mask in remote sensing images," in *Proc. IGARSS IEEE Int. Geosci. Remote Sens. Symp.*, 2020, pp. 1236–1239.

[22] X. Jia, B. De Brabandere, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks for predicting unobserved views," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 1–2.

[23] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "Condconv: Conditionally parameterized convolutions for efficient inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1307–1318.

[24] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 282–298.

[25] X. Sun, B. Wang, Z. Wang, H. Li, H. Li, and K. Fu, "Research progress on few-shot learning for remote sensing image interpretation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 2387–2402, Jan. 2021.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[27] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu, "GhostNet: More features from cheap operations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1577–1586.

[28] G. Song, Y. Liu, and X. Wang, "Revisiting the sibling head in object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11560–11569.

[29] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 649–665.

[30] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOV2: Dynamic, fast instance segmentation,"in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–17.

[31] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

[32] R. Yang, Z. Pan, X. Jia, L. Zhang, and Y. Deng, "A novel CNN-based detector for ship detection based on rotatable bounding box in SAR images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1938–1958, Jan. 2021.

[33] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

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

[35] J. Li, C. Qu, and J. Shao, "Ship detection in SAR images based on an improved faster R-CNN," in *Proc. SAR Big Data Era: Models, Methods Appl.*, 2017, pp. 1–6.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[37] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[38] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6848–6856.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[40] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist. JMLR Workshop Conf. Proc.*, 2011, pp. 315–323.

[41] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016. [Online]. Available: https://arxiv.org/abs/1511.07122

[42] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.

[43] Z. Cui, Q. Li, Z. Cao, and N. Liu, "Dense attention pyramid networks for multi-scale ship detection in SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8983–8997, Nov. 2019.

[44] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019. [Online]. Available: http://arxiv.org/abs/1904.07850

[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[46] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1971–1980.

[47] J. Ren, "Ann vs. SVM: Which one performs better in classification of MCCS in mammogram imaging," *Knowl.-Based Syst.*, vol. 26, pp. 144–153, 2012.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[49] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 8026–8037, 2019.

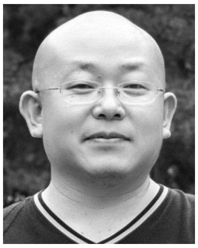
[50] K. Chen *et al.*, "MMDetection: Open MMLab detection toolbox and benchmark," 2019. [Online]. Available: http://arxiv.org/abs/1906.07155

[51] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2. 2019," vol. 2, no. 3, 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[52] M. Everingham, L. van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[53] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[54] J. Dai *et al.*, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 764–773.



Fei Gao received the B.S. degree in electrical automation and the M.S. degree in electromagnetic measurement technology and instrument from Xi’an Petroleum Institute, Xi’an, China, in 1996 and 1999, respectively, and the Ph.D. degree in signal and information processing from Beihang University, Beijing, China, in 2005.

He is currently a Professor with the School of Electronic and Information Engineering, Beihang University. His research interests include target detection and recognition, image processing, deep learning for applications in remote sensing.

Yiyang Huo received the B.S. degree in electronic and information engineering in 2020 from Beihang University, Beijing, China, where he is currently working toward the M.E. degree in information and communication engineering.

His current research activities include target detection, instance segmentation, and remote sensing image processing.

Jun Wang received the B.S. degree in communication engineering from North Western Polytechnical University, Xi’an, China, in 1995 and the M.S. and Ph.D. degrees in signal and information processing from the Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 1998 and 2001, respectively.

He is currently a Professor with the School of Electronic and Information Engineering, BUAA. His research has resulted in over 40 papers in journals, books, and conference proceedings. His research interests include signal processing, DSP/FPGA real-time architecture, target recognition and tracking, and so on.

Amir Hussain received the B.Eng. degree and the Ph.D. degree in electronic and electrical engineering from the University of Strathclyde, Scotland, U.K., in 1992 and 1997, respectively.

Following Postdoctoral and Senior Academic Positions with the West of Scotland (1996–1998), Dundee (1998–2000), and Stirling Universities (2000–2018), respectively, he joined Edinburgh Napier University, Edinburgh, U.K., as founding Head of the Cognitive Big Data and Cybersecurity (CogBiD) Research Lab and the Centre for AI and Data Science. His research interests include cognitive computation, machine learning, and computer vision.

Huiyu Zhou received the B.Eng. degree in radio technology from the Huazhong University of Science and Technology, Wuhan, China, the M.S. degree in biomedical engineering from University of Dundee , Dundee, U.K., and the Ph.D. degree in ratio technology, biomedical engineering, and computer vision from Heriot-Watt University, Edinburgh, U.K.,in 1990, 2002 and 2006, respectively.

He is currently a Professor with the School of Computing and Mathematical Sciences, University of Leicester, Leicester, U.K. His research interests include medical image processing, computer vision, intelligent systems, and data mining.