An Enhanced Lightweight Convolutional Neural Network for Ship Detection in Maritime Surveillance System

Yifan Yin, Xu Cheng, *Member, IEEE*, Fan Shi, *Member, IEEE*, Meng Zhao, Guoyuan Li, *Senior Member, IEEE*, and Shengyong Chen, *Senior Member, IEEE*

***Abstract*—With the extensive application of artificial intelligence, ship detection from optical satellite remote sensing images using deep learning technology can significantly improve detection accuracy. However, the existing methods usually have complex models and huge computations, which makes them difficult to deploy on resource-constrained devices, such as satellites. To solve this problem, this article proposes an enhanced lightweight ship detection model called ShipDetectionNet to replace the standard convolution with improved convolution units. The improved convolution unit is implemented by applying depthwise separable convolution to replace standard convolution and further using the pointwise group convolution to replace the point convolution in depthwise separable convolution. In addition, the attention mechanism is incorporated into the convolution unit to ensure detection accuracy. Compared to the latest YOLOv5s, our model has a comparable performance in mean average precision, while the number of parameters and the model size are reduced by 14.18% and 13.14%, respectively. Compared to five different lightweight detection models, the proposed ShipDetectionNet is more competent for ship detection tasks. In addition, the ShipDetectionNet is evaluated on four challenging scenarios, demonstrating its generalizability and effectiveness.**

***Index Terms*—Attention mechanism, depthwise separable convolution (DS-Conv), maritime surveillance system, optical satellite remote sensing image, pointwise group convolution (G-Conv), ship detection.**

I. Introduction

MARINE traffic is increasingly crucial for economical and social development, especially in the epidemic of COVID-19. Ensuring the safety of marine traffic is of great significance because it can benefit seaborne trade and defend illegal activities, such as maritime terrorism, smuggling, and marine pollution [1]. Nowadays, remote-sensing-imagery-based intelligent ocean surface ship surveillance has become a hot research topic.

Synthetic aperture radar (SAR) images are widely used in the early stage of ship detection because SAR images have the advantages of all-weather observations [2]. However, there are some drawbacks: 1) SAR is vulnerable to the high level of intrinsic noise (speckle); 2) the extraction of geometric features is difficult; and 3) the classification of ship type is hard [3]. Because of these shortcomings of SAR, optical satellite remote sensing image has attracted more attention, as it has higher resolution and retains more geometric details that are not available in SAR images.

Owing to the low resolution of early optical satellite remote sensing image, the ship in the image will be regarded as a point during the detection task, and then, the methods such as constant false alarm rate [4], [5] and generalized likelihood ratio test [6] are applied. However, these methods are easily affected by climate and need to build complex equations. The solution process is time consuming, which is not conducive to real-time application [7]. With improved resolutions of optical satellite remote sensing images, template matching has become a common detection algorithm for ship detection [8], [9]. In this method, a template database is first established manually according to the information of the training image, and then, testing images are matched with the established template database to obtain the detection results. The template matching method is simple and has good results in detecting closely docked ships. However, it is challenging to deal with various complex scenes and requires a lot of effort to establish the template database, and the generalization ability is thereby poor. Especially when the noise is high, it is easy to match incorrectly, which makes the recognition accuracy low.

The emergence of convolutional neural networks (CNNs) opened up a new approach to ship detection [10]. A large number of excellent convolution neural networks were enthusiastically developed, such as VGG [11], GoogleNet [12], [13], You Only Look Once (YOLO) [14]–[17]. Nevertheless, there are some challenges in applying the CNN to ship detection based on optical satellite remote sensing images. First, the CNNs are usually very large; they can be used on ground stations to detect ships. However, collecting data on satellites and transmitting them to

Manuscript received 30 January 2022; revised 3 April 2022, 30 May 2022, and 21 June 2022; accepted 26 June 2022. Date of publication 30 June 2022; date of current version 27 July 2022. This work was supported by the National Natural Science Foundation of China under Grant 62020106004, Grant 61906133, and Grant 92048301. *(Corresponding author: Xu Cheng.)*

Yifan Yin, Fan Shi, Meng Zhao, and Shengyong Chen are with the Engineering Research Center of Learning-Based Intelligent System (Ministry of Education), Tianjin University of Technology, Tianjin 300384, China (e-mail: yifan1996@stud.tjut.edu.cn; shifan@email.tjut.edu.cn; mzhao@ieee.org; sy@ieee.org).

Xu Cheng is with Smart Innovation Norway, 1776 Halden, Norway (e-mail: xu.cheng@ieee.org).

Guoyuan Li is with the Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Ålesund, Norway (e-mail: guoyuan.li@ntnu.no).

Digital Object Identifier 10.1109/JSTARS.2022.3187454

the ground stations for processing could be time consuming. To shorten the time delay of ship information extraction, it is necessary to migrate the ship detection model from the ground to the spaceborne platform (i.e., NVIDIA Jetson TX2). At present, spaceborne hardware resources are limited (i.e., memory size of 8 GB), so it is a challenge to achieve accurate and fast ship detection on a lightweight satellite [18]–[20]. For example, the famous model YOLOv4, which has a superior performance in ship detection, is built on deep, dense, and the number of network layers. However, the number of parameters of YOLOv4 is as high as 60M. The huge network puts forward high requirements for hardware. Second, the CNNs used in marine ship detection are based on the standard convolution (S-Conv). One of the limitations of the S-Conv is that it has a large number of parameters and needs lots of computations. This will delay the detection speed and bring challenges to real-time tasks because of the complex operation. Third, there are various scenes in the optical satellite remote sensing images. Ships may be densely docked in ports and may also have the characteristics of long strips, arbitrary orientations. More seriously, the weather condition is another factor for detecting ships from the optical remote sensing images. These factors make the general CNN detection effect poor.

To address the challenges mentioned above, this article proposes an enhanced CNN-based ship detection network inspired by YOLO, named ShipDetectionNet. The main contributions of this article are as follows.

1) The ShipDetectionNet is designed with a compact size by using an improved convolution unit to replace the S-Conv for parameter reduction but ensuring detection accuracy. The improved convolution is implemented by applying depthwise separable convolution (DS-Conv) to replace S-Conv and further using the pointwise group convolution (G-Conv) to replace the point convolution in DS-Conv. In addition, the channel attention model is incorporated into the convolution unit to ensure detection accuracy.
2) We use optical satellite remote sensing images of marine ships as datasets and comprehensively evaluate the proposed ShipDetectionNet. We compare it with the 11 state-of-the-art baselines and five lightweight models on the whole satellite image dataset, and the experimental results show the superiority of our network. Besides, the ShipDetectionNet is evaluated on different scenarios, demonstrating its generalizability and effectiveness.

The rest of this article is organized as follows. Section II introduces the related work. The overall architecture of the proposed ShipDetectionNet is introduced in Section III-A. The details of the experiments are presented in Section IV. Finally, Section V concludes this article.

II. Related Work

A. CNN-Based Object Detection Methods

At present, object detection methods based on the CNN are mainly divided into two categories: two-stage detection algorithms and one-stage detection algorithms. In this article, we will briefly review the recent research on these target detectors.

The two-stage target detectors, as the name suggests, require two steps when training the whole network. R-CNN proposed by Girshick *et al.* [21] is the opening work of two-stage target detection algorithms, which mainly include three modules. The first module uses selective search to generate object region proposals. Then, AlexNet [10] is applied to extract a 4096-D feature vector from each region proposal. And the last step is using the class-specific linear support vector machine to refine each region. However, the training of R-CNN commonly costs expensive time and space, and the speed of the detection is slow. To solve these flaws, Girshick [22] proposed a new method named Fast R-CNN, which is relatively faster to train and test. This method introduced the region of interest pooling layer so that the network has no size limit on the input of images. What is more striking is Faster R-CNN [23], which is a further improvement of Fast R-CNN. The reason for its excellent performance is that it generates region proposals based on anchor by adding a region proposal network (RPN).

On the contrary, the one-stage method treats the target detection as a regression problem. YOLO series are classical algorithms for one-stage object detection, and YOLOv1–YOLOv4 [14]–[17] have been supported by the literature. The basic idea of the YOLO algorithm is to divide the feature map into $s \times s$ grid cells, and then, each cell is responsible for detecting the targets falling into it to predict the bounding box, confidence, and classification. In addition, Liu *et al.* [24] proposed a single-shot multibox detector (SSD), which directly generates multiscale feature maps through the CNN to detect. Lin *et al.* [25] proposed another detection network named RetinaNet, which solves the problem of extreme imbalance between foreground and background categories by designing the Focal Loss. Different from the above methods relying on anchor boxes, Tian *et al.* [26] proposed a per-pixel prediction fashion, which is a fully convolutional one-stage object detector. Owing to the elimination of anchor boxes, complex computation is reduced.

B. Ship Detection Frameworks

Ship detection plays an important role in maintaining marine safety. Scholars attempt to design different networks for various tasks in the field of ship remote sensing. For instance, Yang *et al.* [27] developed a model, which consists of an RPN and a deep forest ensemble to overcome the complex scene and complex shipshape. Li *et al.* [28] designed a new network called HSF-Net, which adds a hierarchical selective filtering layer to effectively detect ships of different scales. Besides, there are some ship detection models based on the improved YOLO algorithm. Changhua *et al.* [29] proposed a YOLOv3-based ship detection model for small ships. A residual connection and the network structure of the feature pyramid are redesigned to detect the information of small targets. In addition, an equilibrium factor is introduced into the loss function to optimize the weight of small objectives. The experimental results showed that the detection accuracy of this method is improved by 6.3% compared with the original YOLOv3. Liu *et al.* [30] presented another YOLOv3-based CNN for ship detection in different climates. In this model, anchor boxes and bounding boxes are redesigned, as

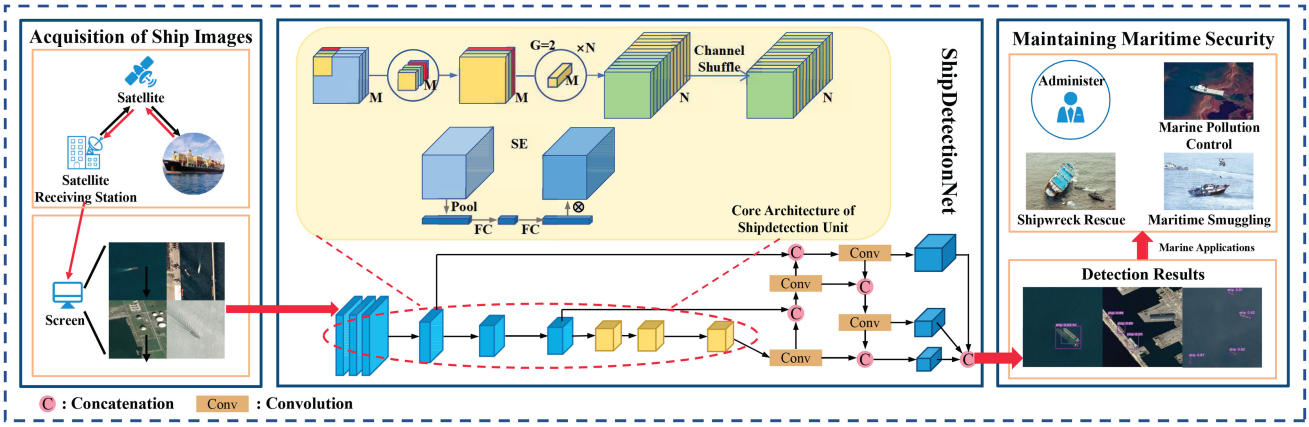


Fig. 1. Flowchart of the proposed ShipDetectionNet for ship detection is based on remote sensing images. The input images are collected using remote sensing techniques. Then, they are received by the ShipDetectionNet, and the detection results are beneficial for marine pollution control, shipwreck rescue, and maritime accident investigation.

well as soft nonmaximum suppression and mixed loss function are introduced to improve the learning and feature expression ability. From this literature, the experimental results illustrate that the proposed eYOLO method outperforms other competing methods.

However, in practical application, there are still challenges in the contradiction between the detection accuracy and the computational complexity of the model. To solve this challenge, many scholars have made extensive efforts in simplifying the model. For example, Wang *et al.* [31] proposed a lightweight convolutional neural network (L-CNN) to detect ship infrared images to tackle the limited spaceborne hardware resources. The L-CNN determines the location of the target through the connected domain, which avoids the traditional regression operation and greatly reduces the computational complexity. Yu *et al.* [32] proposed a lightweight ship detection framework for optical remote sensing images with cloud occlusion. This method exploited the sparse MobileNetV2, which is a pruning structure on MobileNetV2 [33] to obtain candidate subgraphs. Another ship detection algorithm for optical remote sensing images was proposed by Chen *et al.* [34]. This network is an improved YOLOv3 based on an attention mechanism, in which the authors designed a lightweight dilated attention module. In this article, to better balance the storage space of spaceborne hardware resources, we will design a novel ship detection network based on YOLOv5. On the premise of ensuring the detection accuracy, our network dramatically reduces the number of parameters and calculations, effectively compressing the size of the model.

III. Proposed Lightweight CNN Ship Detection Framework

A. Network Structure

The flowchart of the proposed model for ship detection based on remote sensing images is illustrated in Fig. 1. The input images are gathered using remote sensing techniques, such as satellites and unmanned aerial vehicles. The detection results will be used in the maritime surveillance system to push for effective marine accident investigation, accurate ship course navigation,

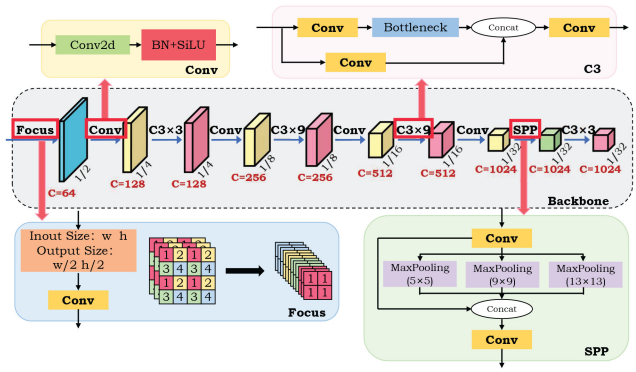


Fig. 2. Backbone of YOLOv5. In summary, the existence of C3, Focus, and SPP structures brings a large number of parameters and calculations, which has a certain redundancy for single-target detection in this article.

stable maritime monitoring, etc. Considering the contradiction between the detection accuracy and computational complexity of the model, a new lightweight model, ShipDetectionNet, is designed based on YOLOv5 in this article. The ShipDetectionNet (the network structure is illustrated in Fig. 5) mainly consists of two proposed convolution units, which are based on DS-Conv, and the pointwise G-Conv is used to replace the point convolution in the DS-Conv. In the meantime, SE blocks [35] are integrated into the proposed convolution units to enhance the model accuracy. Specifically, we thoroughly analyze the limitations of YOLOv5 in Section III-B, corresponding solutions for overcoming the limitations and the construction of our network in Section III-C, and the loss function of ShipDetectionNet in Section III-D. Based on these strategies, the ShipDetectionNet can timely detect ships in practical marine scenes with more robust, accurate, and reliable results.

B. Limitations of YOLOv5

Fig. 2 shows the backbone of the original YOLOv5 model, where $C3 \times 9$ represents nine C3 blocks with 30 such convolution blocks in total, as shown in Fig. 2, which makes the model reach over a hundred layers. Such a cumbersome network comes

with a bunk of redundancy and complexity. The backbone is mainly composed of Focus, Conv, C3, and SPP blocks. The influence of these modules on ship detection is analyzed as follows.

First, C3 is a stack of Conv modules, and Conv is composed of Conv2d, which is a S-Conv. In S-Conv, each kernel convolutes all the channels of the input feature maps, and the number of convolution kernels is equal to the number of output channels. P denotes the number of parameters, Q means the calculation cost, and the equations are as follows:

$$P_{\text{S-Conv}} = k \cdot k \cdot N_{\text{input}} \cdot N_{\text{output}} \tag{1}$$

$$Q_{\text{S-Conv}} = L \cdot L \cdot k \cdot k \cdot N_{\text{input}} \cdot N_{\text{output}} \tag{2}$$

where L is the size of the input image, k is the size of the convolution kernel, and N_{input} and N_{output} represent the number of input channels and the number of output channels, respectively. In general practical tasks, the number of N_{input} and N_{output} generally can reach hundreds or even thousands. Multiplication between them will make P and Q even large. Meanwhile, YOLOv5 has up to 24 layers of C3 in its backbone, which will produce a large number of parameters and calculations.

Second, Focus is used to slice the image before entering the backbone. The specific operation is to get a value every two pixels on a feature map and joint it into a new feature map. The Focus operation centralizes the information of the spatial dimension to the channel dimension so that the input channel is expanded by four times. For example, a $640 \times 640 \times 3$ image is input into the network, and the convolution kernel of the Focus module is 3×3. The number of output channels is four times that of input channels, i.e., 32. The comparison of parameters (presented by P) and computations (presented by Q) between the Focus module and S-Conv is as follows:

$$P_{\text{S-Conv}} = 3 \times 3 \times 3 \times 32 = 864$$

$$P_{\text{Focus}} = 3 \times 3 \times 12 \times 32 = 3456$$

$$Q_{\text{S-Conv}} = 3 \times 3 \times 3 \times 32 \times 320 \times 320 = 88473600$$

$$Q_{\text{Focus}} = 3 \times 3 \times 12 \times 32 \times 320 \times 320 = 353894400.$$

It can be seen that the Focus produces more parameters and computational cost than the S-Conv. The Focus proposed by the author of YOLOv5 is to replace the first three convolution layers in YOLOv3. In our network, we do not need to merge several layers like YOLOv3; on the contrary, the Focus will increase the number of parameters and calculation cost, so this structure is removed.

Finally, the SPP structure is at the back of the backbone. The significance is that the network using SPP can receive images of any size. In YOLOv5, SPP makes the fusion of local features and global features. Because YOLO is applied to complex multitarget detection, SPP is conducive to the case of the large difference in target size. However, the task of this article is a single target detection, and the size difference between targets is relatively small. SPP does not greatly improve detection accuracy in our task. To reduce the complexity of the network, this module is, therefore, removed.

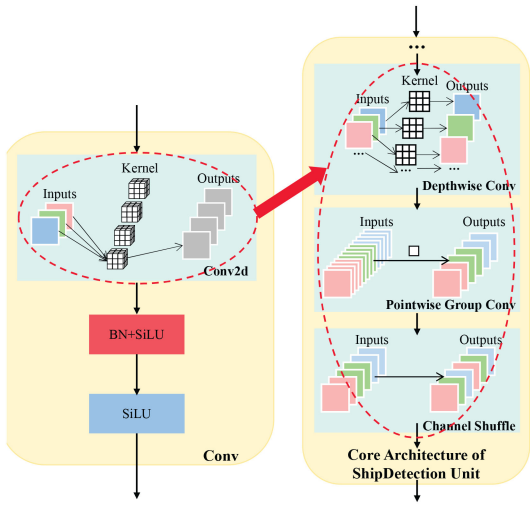


Fig. 3. Improved convolution units. The core architecture of ShipDetection Unit is implemented by using DS-Conv to replace Conv and further applying the group PW-Conv to replace the point convolution in DS-Conv.

In summary, the existence of C3, Focus, and SPP blocks brings a large number of parameters and calculations, which comes with huge redundancy for single-target detection.

C. Construction of ShipDetectionNet Architecture

The ShipDetectionNet proposed in this article is based on YOLOv5. We designed our convolution units to construct the ShipDetectionNet for ship detection, and these convolution units are named ShipDetection Unit. Through the analysis in the previous section, the Focus and SPP structures in YOLOv5 will cause some complexity. To make the detection network lighter, Focus and SPP are removed. Besides, YOLOv5 is stacked by C3 composed of Conv, which is S-Conv. Therefore, it can make the network produce a large number of parameters and computation costs. The improved convolution units are implemented by using DS-Conv to replace Conv and further applying the group pointwise convolution (PW-Conv) to replace the point convolution in DS-Conv, as shown in Fig. 3. The proposed method for complexity reduction and high detection performance of the network is as follows.

1) Replace S-Conv With DS-Conv: DS-Conv, which is a form of factorized convolutions that factorize an S-Conv into a depthwise convolution (DW-Conv) and a PW-Conv. A single kernel is applied to each input channel in the DW-Conv. Then, the PW-Conv applies a 1×1 convolution to combine the outputs of the DW-Conv. This factorized convolution has the effect of drastically reducing the model size and computational cost [36]–[38].

The number of parameters (presented by P) and computational cost (presented by Q) of the DS-Conv is

$$P_{\text{DS-Conv}} = k \cdot k \cdot N_{\text{input}} + N_{\text{input}} \cdot N_{\text{output}} \tag{3}$$

$$Q_{\text{DS-Conv}} = L \cdot L \cdot k \cdot k \cdot N_{\text{input}} + L \cdot L \cdot N_{\text{input}} \cdot N_{\text{output}}. \tag{4}$$

The ratio r_1 of (3) to (1) is

$$r_1 = \frac{k \cdot k \cdot N_{\text{input}} + N_{\text{input}} \cdot N_{\text{output}}}{k \cdot k \cdot N_{\text{input}} \cdot N_{\text{output}}} = \frac{1}{N_{\text{output}}} + \frac{1}{k^2}. \tag{5}$$

Similarly, the ratio r_2 of (4) to (2) is

$$r_2 = \frac{L \cdot L \cdot k \cdot k \cdot N_{\text{input}} + L \cdot L \cdot N_{\text{input}} \cdot N_{\text{output}}}{L \cdot L \cdot k \cdot k \cdot N_{\text{input}} \cdot N_{\text{output}}} = \frac{1}{N_{\text{output}}} + \frac{1}{k^2} \tag{6}$$

where $N_{\text{output}} \gg 1$ and $k > 1$. Therefore, $r_1 < 1$ and $r_2 < 1$ show that the DS-Conv has fewer parameters and calculations than the S-Conv. Applying it to the ShipDetectionNet can compress the model size and reduce the complexity of the network to better adapt to the ship detection task of a single target.

However, the PW-Conv, the second part of the DS-Conv, is also the S-Conv, which will produce more parameters and calculations. If it is improved, the complexity of the model will be further reduced.

2) Replace PW-Conv in DS-Conv With Pointwise G-Conv: The application of G-Conv can be traced back to AlexNet [10], which was used to distribute the model over two GPUs.

Assuming that g is the number of groups, the number of parameters of G-Conv is

$$P_{\text{G-Conv}} = k \cdot k \cdot \frac{N_{\text{input}}}{g} \cdot \frac{N_{\text{output}}}{g} \cdot g. \tag{7}$$

The ratio of (7) and (1) is

$$\text{ratio}_3 = \frac{1}{g} \tag{8}$$

where $g > 0$; then, $\text{ratio}_3 < 1$. It is proved that the G-Conv can obtain the same number of feature maps with fewer parameters, and the number of parameters of the G-Conv is $1/g$ of the S-Conv. Therefore, the G-Conv also makes a great contribution to the compression of the model.

In the DS-Conv, the DW-Conv is a special G-Conv, and its number of groups is equal to the number of input channels. And the PW-Conv in the DS-Conv is S-Conv. If we consider the application of G-Conv, that is, group the PW-Conv, we can reduce the number of parameters. However, if multiple G-Convs stack together, there is a side effect: It is obvious that outputs from a certain group only relate to the inputs within the group [39]. It prevents information flow between groups and weakens information expression. Therefore, the channel shuffle (CS) is followed after the pointwise G-Conv, simply to recombine the channels between different groups.

The comparison between the original DS-Conv and the improved DS-Conv is shown in Fig. 4. The combination of pointwise G-Conv and CS to replace the PW-Conv in the DS-Conv can significantly reduce the consumption of computing cost.

3) Use SE Block to Improve Detection Accuracy: Considering the reality, we reduce the model size to adapt to the deployment of offshore platforms. However, the complexity reduction of the network may lead to the loss of detection accuracy. To ensure detection performance, we add SE blocks to some convolution units. As shown in Fig. 5, the SE blocks are added between the DW-Conv and the PW-Conv in each ShipDetection Unit(b). The reason why the SE blocks are here will be presented in Section IV.

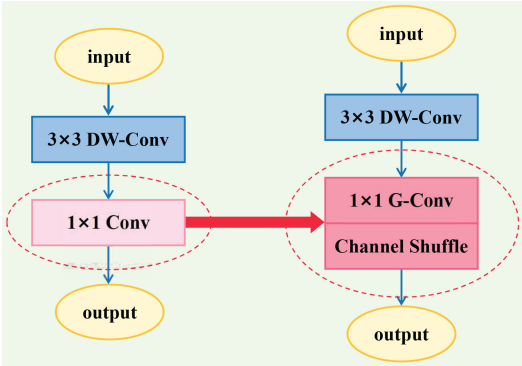


Fig. 4. Comparison between the original DS-Conv and the improved DS-Conv.

The neural network continuously extracts information features in forwarding propagation and backing propagation, but not all features play a crucial role. Through the SE mechanism, the network can search for more powerful representations, which capture the most significant image properties in a given task, thereby improving performance [35].

Suppose that the dimension of the input feature map is $H \times W \times C$, where H, W, and C represent the height, width, and the number of channels, respectively, and U is the tensor of this matrix. First, the global average pooling is performed, that is, $squeeze$. The formula is

$$z_c = F_{\text{sq}}(U_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} U_c(i, j). \tag{9}$$

It means that the cth element of z is generated by shrinking U through its spatial dimensions $H \times W$. To take advantage of the information gathered during the $squeeze$ operation, the second operation, $excitation$, is performed

$$s = F_{\text{ex}}(z, W) = \sigma(g(z, W)) = \sigma(W_2 \delta(W_1 z)) \tag{10}$$

where δ and σ refer to the ReLU function and Sigmoid function, respectively, z represents the result obtained by (9), and W is the weight. The dimension of W_1 is $\frac{C}{r} \times C$ and the dimension of W_2 is $C \times \frac{C}{r}$. r represents the reduction ratio, which aims to reduce the number of channels to reduce the calculation cost. To obtain the final output, U will be rescaled with the s acquired above

$$U' = F_{\text{scale}}(U_c, s_c) = U_c \times s_c \tag{11}$$

where $F_{\text{scale}}(U_c, s_c)$ refers to channelwise multiplication between the scalar s_c and the feature map U_c. It remarks the channel so that it can focus on the key information. The detailed principle of the SE block can be found in [35].

4) Constructing ShipDetection Unit: Our convolution network units are constructed based on the three methods proposed

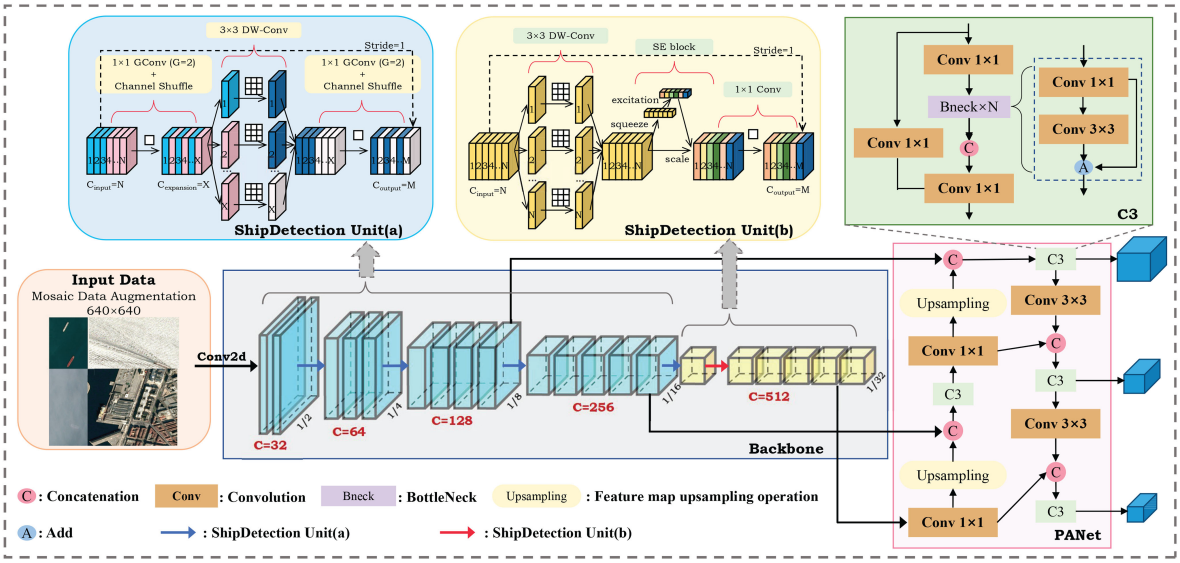


Fig. 5. Structure of the ShipDetectionNet. In the ShipDetectionNet, except that the first convolution layer is S-Conv, all the other convolution layers are ShipDetection Units. The numbers 1, 1/2, 1/4, 1/8, and 1/16 in the backbone represent the reduction ratio of the feature maps after downsampling. After the backbone is in the detection layer, we use the PANet as the detection layer. The ShipDetectionNet has 20 layers.

above to reduce the complexion of the model and ensure detection accuracy.

The design of the ShipDetection Unit draws lessons from the bottleneck in ResNet [40]. A bottleneck consists of a PW-Conv for dimension reduction, a 3×3 convolution, and a PW-Conv for dimension upgrading. Our network units use the composition of the bottleneck for reference, the PW-Conv was replaced with the pointwise G-Conv and CS, and the 3×3 convolutions were replaced with the DW-Conv. The number of output channels will be small because the DW-Conv uses a single kernel for each input channel. If we first use dimensionality reduction PW-Conv such as bottleneck to compress and then go through DW-Conv, fewer features will be extracted. Therefore, we adopt an inverse method, that is, first increase the dimension, then convolute, and finally reduce the dimension [33], [40]–[42].

The network convolution unit, namely ShipDetection Unit, is mainly divided into two categories based on the bottleneck. The classification basis is whether the number of input channels is the same as that of intermediate channels. When the number of input channels and middle layer channels is different, the pointwise G-Conv ($\mathrm{CONV_{PG}}$) and CS are carried out first. This combination can significantly reduce the number of network parameters [39]. Then, the DW-Conv ($\mathrm{CONV_{DW}}$) is performed, and finally, the pointwise G-Conv ($\mathrm{CONV_{PG}}$) and CS are completed. The convolution operation in ShipDetection Unit(a) can be summarized as follows:

$$\begin{aligned} Y &= \mathrm{CONV_{PG}}(X, W, b) \\ Y &= \mathrm{CS}(Y) \\ Y &= \mathrm{CONV_{DW}}(Y) \\ Y &= \mathrm{CONV_{PG}}(Y) \\ Y &= \mathrm{CS}(Y) \end{aligned} \tag{12}$$

where X is the input features, Y is the intermediate features, and W and b refer to trainable parameters in the ShipDetectionNet. When the number of input channels is the same as the number of intermediate channels, since the number of channels after the DW-Conv is the same as the number of input channels, to reduce the redundancy of the network, the previous pointwise G-Conv is removed, and the DW-Conv ($\mathrm{CONV_{DW}}$) is carried out directly. In addition, since the ShipDetection Unit(b) shown in Fig. 5 is stacked at the back of the network, it will be followed by the prediction layer. This part of the ShipDetection Unit will no longer set the pointwise G-Conv but the PW-Conv ($\mathrm{CONV_{PW}}$) to better improve the representation of features. The convolution operation in ShipDetection Unit(b) can be summarized as follows:

$$\begin{aligned} Y &= \mathrm{CONV_{DW}}(X, W, b) \\ Y &= \mathrm{SE}(Y) \\ Y &= \mathrm{CONV_{PW}}(Y). \end{aligned} \tag{13}$$

In the above two types of ShipDetection Units, the processes of $stride = 1$ and $stride = 2$ are different. When $stride = 2$, because the number of input channels is different from the number of output channels, the shortcut structure is not added, which is the reason why the shortcut connection is a dotted line in Fig. 5.

5) Building ShipDetectionNet Using ShipDetection Units: In the ShipDetectionNet, except that the first convolution layer is S-Conv, all other convolution layers are ShipDetection Units. As shown in Fig. 5, the blue convolution blocks in the first half represent ShipDetection Unit(a), and the yellow convolution blocks in the second half represent ShipDetection Unit(b). Before inputting the image into the network, the data are preprocessed, mainly including translation, flipping, mosaic data augmentation, etc. Mosaic data augmentation splices four images together to form a new 640×640 image. The numbers 1, 1/2, 1/4, 1/8, and

TABLE I
NETWORK ARCHITECTURE (SHIPDETECTIONNET)

Input size	Operator	Exp size	Out	SE	Stride
$224^2 \times 3$	Conv2d	—	32	—	2
$112^2 \times 32$	ShipDetection Unit(a)	32	32	0	1
$112^2 \times 32$	ShipDetection Unit(a)	96	64	0	2
$56^2 \times 64$	ShipDetection Unit(a)	192	64	0	1
$56^2 \times 64$	ShipDetection Unit(a)	192	64	0	1
$56^2 \times 64$	ShipDetection Unit(a)	192	128	0	2
$28^2 \times 128$	ShipDetection Unit(a)	384	128	0	1
$28^2 \times 128$	ShipDetection Unit(a)	384	128	0	1
$28^2 \times 128$	ShipDetection Unit(a)	384	128	0	1
$28^2 \times 128$	ShipDetection Unit(a)	384	256	0	2
$14^2 \times 256$	ShipDetection Unit(a)	384	256	0	1
$14^2 \times 256$	ShipDetection Unit(a)	384	256	0	1
$14^2 \times 256$	ShipDetection Unit(a)	384	256	0	1
$14^2 \times 256$	ShipDetection Unit(a)	384	256	0	1
$14^2 \times 256$	ShipDetection Unit(a)	384	512	0	2
$7^2 \times 512$	ShipDetection Unit(b)	512	512	1	1
$7^2 \times 512$	ShipDetection Unit(b)	512	512	1	1
$7^2 \times 512$	ShipDetection Unit(b)	512	512	1	1
$7^2 \times 512$	ShipDetection Unit(b)	512	512	1	1
$7^2 \times 512$	ShipDetection Unit(b)	512	512	1	1

1/16 in the backbone represent the reduction ratio of the feature maps after downsampling. To obtain the output of three feature layers to detect targets of different sizes, the output branches are set in layer 9, layer 14, and the last layer. After the backbone, we use the PANet [43] as the detection layer. In the process of DW-Conv, to reduce the number of network parameters, the size of the convolution kernel is 3×3. The h_swish is selected as the activation function, which aims to reduce the computational cost of the network on the premise of ensuring accuracy. It is defined as

$$h\_swish(x) = x \cdot \frac{\text{ReLU}\,(x+3)}{6}. \tag{14}$$

We present the overall ShipDetectionNet architecture in Table I, where $Exp\,size$ and Out represent the number of intermediate channels and the number of output channels, respectively.

Since the size of the feature maps decreases with the deepening of the network gradually, it is easier to lose the key feature information of the ship, so the number of channels of the feature maps is set to gradually increase with the depth of the network. The proposed ShipDetectionNet structure is a line structure as a whole, without complex branches, with only 20 layers, which reduces the network's redundancy and is more suitable for ship detection from sensing remote satellite images.

D. Loss Function of the ShipDetectionNet

The ShipDetectionNet is a one-stage target detector, so there is no need to generate region proposals. Its loss function consists of three parts: location loss, category loss, and confidence loss. The overall loss is the sum of the above three.

The ShipDetectionNet uses CIoU loss as its location loss, that is, the loss of bounding box regression. Its definition is as follows:

$$L_{\text{CIoU}} = 1 - \text{IoU} + \frac{\rho^2\,(B_P, B_G)}{c^2} + \alpha\upsilon \tag{15}$$

where IoU is the intersection over union of the bounding box and the ground truth, B_P and B_G represent the center points of the bounding box and the center points of the ground truth, respectively, ρ represents the Euclidean distance between the two points, and c refers to the diagonal distance of the minimum closure region that can contain the bounding box and the ground truth at the same time. υ represents the normalization of the difference between the width-to-height ratio of the bounding box and the ground truth, and α is the balancing factor that balances the loss aroused by IOU and the loss of the width and height scale. υ and α are defined as follows:

$$\upsilon = \frac{4}{\pi}\left(\arctan\frac{w_G}{h_G} - \arctan\frac{w_P}{h_P}\right)^2 \tag{16}$$

$$\alpha = \frac{\upsilon}{1 - \text{IoU}\,(B_P, B_G) + \upsilon} \tag{17}$$

where w_G and h_G mean the width and height of the ground truth, respectively; similarly, w_P and h_p are the width and height of the bounding box, respectively.

The category loss function of the bounding box uses binary cross entropy loss (BCEWithLogitsLoss), which is shown as follows:

$$\hat{y}_i = \text{Sigmoid}\,(\hat{x}_i) = \frac{1}{1 + e^{-\hat{x}_i}} \tag{18}$$

$$L_{\text{class}} = -\sum_{n=1}^{N}\left[y_i \log\,(\hat{y}_i) + (1 - y_i)\log\,(1 - \hat{y}_i)\right] \tag{19}$$

where N represents the total number of categories. In our task, N is 1. $\hat{x}_i$ is the predicted value of the current category, $\hat{y}_i$ is the predicted probability obtained after Sigmoid function, and y_i is the true value (1 or 0).

The confidence loss function of the bounding box also uses binary cross entropy loss (BCEWithLogitsLoss), as described in the following:

$$\hat{C}_i = \text{Sigmoid}\,(\hat{x}_i) = \frac{1}{1 + e^{-\hat{x}_i}} \tag{20}$$

$$L_{\text{obj}} = -\sum_{n=1}^{M}\left[C_i \log\left(\hat{C}_i\right) + (1 - C_i)\log\left(1 - \hat{C}_i\right)\right] \tag{21}$$

where M represents the total number of generated bounding boxes. C_i and $\hat{C}_i$ are the value of both true confidence and predicted confidence, respectively.

Therefore, the loss function of the ShipDetectionNet consists of the above three parts:

$$\text{Loss} = L_{\text{CIoU}} + L_{\text{class}} + L_{\text{obj}}. \tag{22}$$

IV. EXPERIMENTS

A. Experimental Setup

1) Settings: We conduct all the experiments on an NVIDIA Titan V GPUs (12 GB) built-in server and introduce Pytorch to implement all the compared models. Through the training process, the batch size is set to 4. We stopped training after 300 epochs. Rather than searching for the best hyperparameters in

the hyperparameter space, we use the same training parameters as those in the corresponding models.

2) Datasets: The data used in this article are from a Kaggle competition for marine ship detection.[1] The official dataset contains 29 GB of images, including 192 556 in the training set and 15 606 in the testing set. We randomly selected 1600 768×768 pixel images (the size of the original image was 768×768 pixels, and it was preprocessed to 640×640 pixels and entered into the ShipDetectionNet) as the dataset of the proposed method and randomly divided them into a training set, a verification set, and a testing set according to the ratio of 7:2:1. The robustness of the proposed model can be effectively verified because of the diverse direction and size of targets in remote sensing satellite images and the extensive ship scenes.

3) Evaluation Metrics: We use *recall*, *precision*, and the *mean average precision (mAP)* for evaluation, defined as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{23}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{24}$$

$$\text{mAP} = \int_0^1 \text{Precision}\left(\text{Recall}\right) d(\text{Recall}) \tag{25}$$

where TP, FP, and FN refer to true positive, false positive, and false negative, respectively. Precision (recall) is the precision–recall curve.

In addition, to adapt to the special marine environment, the complexity of our model is relatively small, so the model size and network parameters are also used for the evaluation.

B. Comparison With State-of-the-Art Methods

We compare the ShipDetectionNet with the following 11 state-of-the-art methods, which are described as follows.

1) *Faster R-CNN [23]:* This two-stage detection algorithm introduced the RPN to generate proposals directly with high recall. The backbone of the selected model is ResNet50 with five FPN feature layers in the $mmdetection$ framework.
2) *RetinaNet [25]:* This algorithm proposed Focal Loss in the classification branch, which solves the problem of positive and negative sample imbalance in target detection. The backbone of the selected model is ResNet50 with five FPN feature layers in the $mmdetection$ framework.
3) *SSD300 [24]:* SSD is a typical one-stage target detection algorithm, which added the pyramidal feature hierarchy-based detection method to realize multiscale detection. We use the SSD300, that is, the input is 300×300 pixel images.
4) *YOLOv2 [15]:* The YOLO algorithm was first proposed to treat the target detection as a regression problem. Compared with YOLOv1, batch normalization is added to make the mAP significantly improved.
5) *YOLOv3 [16]:* To achieve a better training effect, the author trained Darknet-53 as the backbone and designed three scales to fuse the feature maps of different sizes.
6) *YOLOv3-tiny [16]:* YOLOv3-tiny removed some feature layers based on YOLOv3 and retained only two independent prediction branches, suitable for tasks requiring high-speed detection.
7) *YOLOv4 [17]:* This algorithm integrated the best optimization strategies in the field of CNN in recent years. It is easy to distort the image when resizing the image, so the $letterbox$ is used to fill the image to maintain the ratio of length and width.
8) *YOLOv5s:* YOLOv5s is the network with the smallest depth and the smallest width in the YOLOv5 series.
9) *Mask R-CNN [44]:* This algorithm adds a branch of output object mask on Faster R-CNN to make it competent for instance segmentation. In short, Mask R-CNN can complete three tasks: classification, regression, and segmentation.
10) *VFNet [45]:* The authors of this article proposed a novel detection score, IoU-aware classification score (IACS), to achieve higher detection accuracy in dense object detectors. A new loss function Varifocal Loss and a new star-shaped bounding box feature representation are designed to predict and estimate the IACS, respectively.
11) *YOLOF [46]:* This article reconsidered the FPN and pointed out that the success of FPN lies in its divide-and-conquer strategy rather than multiscale feature fusion. Compared with the traditional FPN, this network uses only one-level feature for detection, which greatly improves the detection efficiency.

TABLE II
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT METHODS

Methods	$R\,(\%)$	$mAP\,(\%)$	$Para$	$Size\,(MB)$
Faster R-CNN	83.4	81.51	41.12M	314
Mask R-CNN	72.4	79.06	43.75M	335
RetinaNet	84.7	76.56	36.10M	276
SSD300	83.7	69.24	23.75M	181
YOLOv4	72.4	81.63	63.94M	244
YOLOv3	76.7	75.55	61.50M	117
YOLOv3-tiny	73.6	72.69	8.67M	16.6
YOLOv2	68.7	62.93	50.66M	193
YOLOv5s	83.1	82.64	7.05M	13.7
VFNet	78.5	79.64	32.48M	249
YOLOF	66.2	57.58	42.06M	322
ShipDetectionNet	83.4	**83.03**	**6.05M**	**11.9**

The results shown in Table II (where R and Para refer to recall and the number of parameters, respectively) indicate that our model beats other models in terms of mAP, the number of parameters, and the model size. YOLOv5s is ranked second, and YOLOv2 is the last in mAP. Compared with the latest YOLOv5s, our model has a comparable performance in mAP, while the number of parameters and the model size are reduced by 14.18% and 13.14%, respectively. In addition, YOLOv4 and Faster R-CNN are also close to our model in mAP, but their parameters are 10.57 and 6.80 times than ours, respectively, and their model sizes are 20.50 and 26.39 times than ours,

[1][Online]. Available: https://www.kaggle.com/c/airbus-ship-detection

respectively. Obviously, it is still a challenge for them to balance the detection accuracy and spatial resources. The Faster R-CNN is a single-task model that can only detect targets, and the Mask R-CNN is improved to form a multitask model that can also complete segmentation tasks. Mask R-CNN was trained on polygons, so the branch that completed the segmentation also participated in the training. On large general object datasets such as COCO, the authors of Mask R-CNN show considerable experimental results in this article. However, for the scenario of this article, the task is single and mainly focuses on ship detection without considering other objects, which may not give full play to the advantages of Mask R-CNN. From another point, the network parameters of YOLOv3-tiny are close to that of the ShipDetectionNet, but the performance of YOLOv3-tiny has decreased 12.5% for mAP. Therefore, to ensure accuracy, the ShipDetectionNet with a low number of parameters better balances the detection performance and the model complexity.

It can be seen that the R and mAP trends of RetinaNet and YOLOv4 are opposite; the reasons are as follows: the contribution of RetinaNet lies in the design of the Focal Loss function, which adds weights to balance the number of positive and negative samples, making the loss function pay more attention to the samples that are difficult to distinguish. This can be of great advantage in complex multicategory scenarios. However, our task is relatively simple, and there is only one foreground, which makes the classification of foreground and background not so hard. If Focal Loss continues to be used to increase the weight of positive samples, the focus of RetinaNet training will be excessive on the positive sample, which will result in producing a large number of bounding boxes. The impact of this is that positive samples are well detected, resulting in a high recall. However, some negative samples are also classified as positive samples, leading to a low precision, which pulls down the mAP value. In YOLOv4, the backbone is deep. With continuous downsampling, it is easy to lose pixels of small targets, so small targets cannot be detected well, resulting in low recall. However, YOLOv4 integrated a series of tricks to improve precision, generating a higher mAP. The above analysis also shows that these two networks are not suitable for our dataset.

It is worth noting that YOLOF, as a new target detector, achieve poor results. The possible reason may be that YOLOF only employs a single-layer C5 feature (with a downsample rate of 32) to detect, while the receptive field of the C5 feature can only cover a limited scale range. Although the dilated encoder makes up for the larger receptive field range missing in the C5 feature, it still lacks the receptive field required for small target detection. And there are many small ships in our dataset, so the performance of YOLOF is greatly weakened in our dataset.

C. Comparison With Lightweight Models

We also compare the ShipDetectionNet with five lightweight target detection methods, which are described as follows.

1) *ShuffleNetv2 [47]:* For the shortcomings of ShuffleNetv1 and to improve the speed, ShuffleNetv2 proposed four criteria to reduce the memory access cost, avoid network fragmentation, and reduce elementwise operations.

TABLE III
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT LIGHTWEIGHT METHODS

Methods	$R\,(\%)$	$P\,(\%)$	$mAP\,(\%)$	$Para$
ShuffleNetv2	69.02	88.24	68.19	0.44M
MobileNetv2	74.54	93.10	73.80	5.95M
MobileNetv3-small	77.61	93.36	77.32	3.54M
MobileNetv3-large	80.06	92.55	79.68	5.20M
SARShipNet-20	70.86	90.23	69.58	3.22M
ShipDetectionNet	**83.44**	90.97	**83.03**	6.05M

2) *MobileNetv2 [33]:* MobileNetV2 designed the linear bottlenecks and inverted residuals based on DS-Conv, improving the network performance.
3) *MobileNetv3-large [42]:* On the bias of MobileNetv2, this algorithm added the attention model and improved the activation function using h_swish, instead of ReLU6.
4) *MobileNetv3-small [42]:* It is a reduced version of the MobileNetv3-large, which contains fewer convolutional blocks and fewer filters.
5) *SARShipNet-20 [7]:* This is a lightweight ship detection network for SAR images. Because the code in [7] is not open source, we reproduced this network. To ensure the detection performance, we did not cut the image to 80×80 according to the requirements in [7], but cut it to 640×640 according to the size of our dataset.

To ensure the consistency of experimental variables, we embedded the above networks into the framework of YOLOv5 to enable them to complete the target detection task.

The results shown in Table III (where R, P, and Para refer to recall, precision, and parameter quantity, respectively) illustrate that our model achieves the best in mAP. Although the parameters of ShuffleNetv2 and MobileNetv2 are lower than those of our model, their accuracy is 14.84 and 9.23 lower in mAP than ours, respectively, which is obviously not suitable for the high-accuracy ship detection task. Similarly, MobileNetv3-small and MobileNetv3-large are lighter than ShipDetectionNet, but their accuracy is 77.32% mAP and 79.68% mAP, respectively, which is 5.71 and 3.35 lower than ShipDetectionNet, respectively. The performance of SARShipNet-20 on our dataset is not good. The reason may be that it is difficult to extract the complex feature information of optical images. These networks sacrifice the accuracy of detection in exchange for the lighter weight of the network model. Our purpose is not only to reduce the detection model size but also to evaluate the tradeoffs between accuracy, the number of operations, and the number of parameters. Considering comprehensively, although the lightweight target detection model in Table III is lighter, the detection accuracy is not high enough. The ShipDetectionNet balances the accuracy with the complexity of the model and is more competent for the ship detection field.

D. Comparison of Methods for Special Scenes

To show that the proposed model is suitable for ship detection in various special scenes, the testing set is divided into four types: open sea, nearshore, small ships, and cloud barrier. We put the

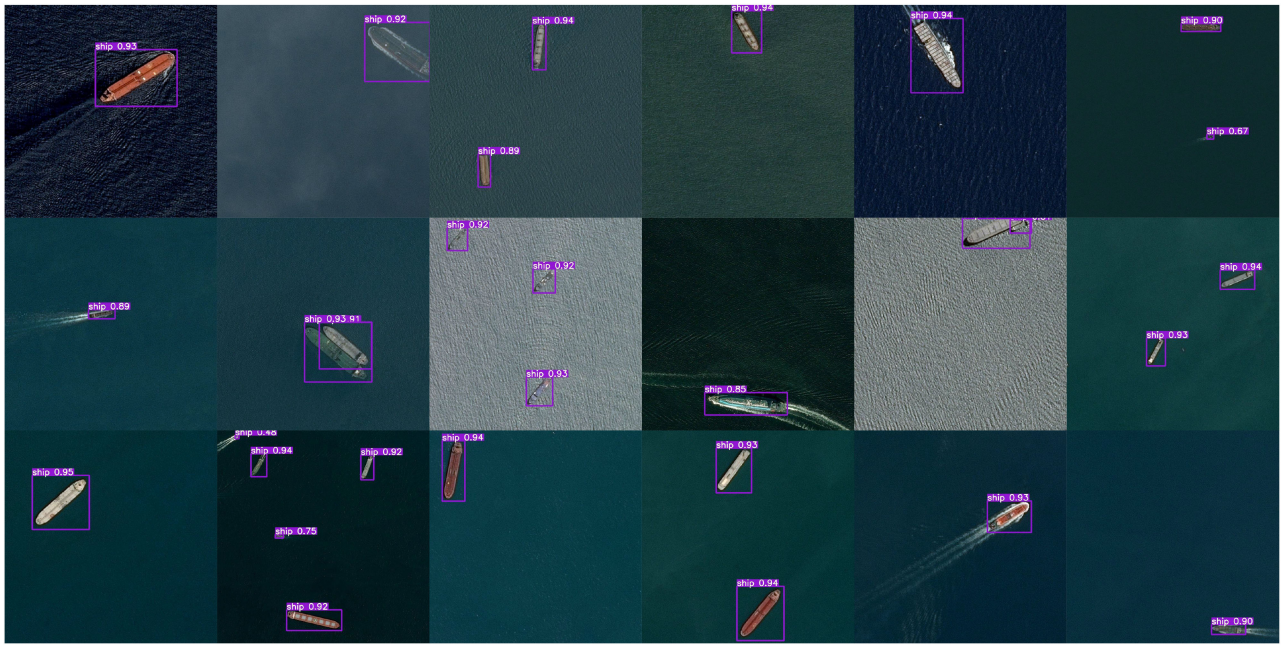


Fig. 6. Test results of the ShipDetectionNet in the scenario of open sea.

Fig. 7. Test results of the ShipDetectionNet in the scenario of nearshore.

above four types of datasets into ShipDetectionNet for testing, respectively, and the results are analyzed as follows.

1) Open Sea: The images of the open sea account for the largest proportion in the whole dataset, including both large ships and small ships. Since there is no interference from the surroundings (surrounded by the sea), the ship is easy to detect, so the detection confidence is also high. In Fig. 6, the detection confidence of most ships is above 90%, and the highest detection confidence is 95%. It is important to note that the ShipDetectionNet can successfully distinguish two ships when they are close together, which is challenging to do in some other detectors.

2) Nearshore: Corresponding to the ships in the open sea is the ships that are nearshore. Owing to the interference of sundries, the targets are difficult to identify. For example, it is easy to mistakenly detect nearshore objects like ships. However, it can be seen from Fig. 7 that our model can well identify the ships, and the highest detection confidence is 95%. In a complex environment, two adjacent ships can also be separated by the ShipDetectionNet. It shows that our model is not affected by the surroundings and has robust detection ability.

3) Small Ships: Owing to the taking distance, taking angles, and other reasons, the targets are quite small in the obtained images. The continuous downsampling of the CNN will make

Fig. 8. Test results of the ShipDetectionNet in the scenario of small ships.

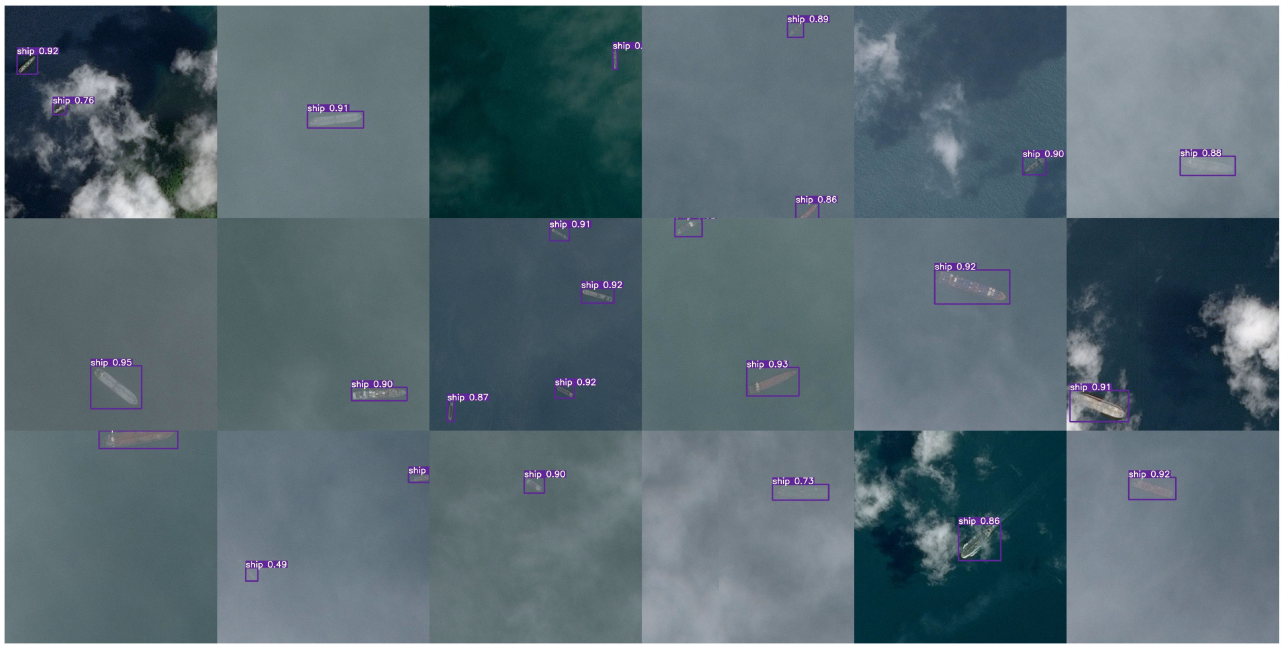


Fig. 9. Test results of the ShipDetectionNet in the scenario of cloud barrier.

the position information of the feature map rough or even lose it, which is easy to cause inaccurate positioning or detection missed of small-scale ships. Therefore, there are some difficulties and challenges in the recognition of small ships. Fig. 8 shows the detection results. It can be seen that the ship is rather small, and in some images, it is not easy to distinguish the target and the background with the naked eye as if the ship and the background are fused. However, our model can even make the confidence of some ships more than 90%, which is enough to improve the effectiveness of the ShipDetectionNet.

4) Cloud Barrier: In practical applications, marine ship images are often affected by bad weather. Owing to the dataset, we only tested the detection effect of the ShipDetectionNet in a cloudy environment. With clouds stretching over the ocean, fuzzy ship images shot by the satellite present low contrast and poor fine details and arouse challenges over accurate ship detection. Fig. 9 shows the detection results. It can be seen that good detection results can be obtained for both larger and smaller ships. Even ships that are difficult to see with the naked eye are successfully identified by the ShipDetectionNet. It is enough to illustrate the adaptability of the proposed method to severe weather conditions.

Finally, we found that in the image containing multiple ships, the detection performance of small ships is worse than that of

TABLE IV
EVALUATION INDEX OF SHIP DETECTION RESULTS OF THE SHIPDETECTIONNET

Models	R	mAP	$Para$	G	$T\,(ms)$
ShipDetectionNet1	81.90	81.29	6.78M	16.0	14.36
ShipDetectionNet2	82.52	81.93	5.39M	14.8	10.42
ShipDetectionNet3	82.82	82.26	6.12M	15.4	13.40
ShipDetectionNet4	**83.44**	**83.03**	6.05M	15.3	10.66

TABLE V
COMPARISON OF DETECTION PERFORMANCE OF DIFFERENT LIGHTWEIGHT METHODS IN FAIR1M

Methods	$Recall\,(\%)$	$Precision\,(\%)$	$mAP\,(\%)$
ShuffleNetv2	50.16	84.20	47.79
MobileNetv3-small	66.23	87.93	64.30
MobileNetv3-large	72.24	87.25	70.73
YOLOv5s	79.22	86.37	77.30
ShipDetectionNet	**80.03**	83.14	**77.56**

large ships. On the one hand, it is caused by the characteristics of the network itself. To increase the receptive field in the network, the feature map is required to be continuously shrunk, so the information of the small area is naturally difficult to be transmitted to the rear of the detector. On the other hand, in our dataset, the number of small ships is significantly less than that of large ships, which brings some difficulties for the network to adapt to the target. In the future work, we will also continue to explore ways to improve the detection accuracy of small ships, to make the overall detection ability of the network more comprehensive.

E. Ablation Experiment

We propose a method to reduce the number of parameters and calculations of the network, but reducing the complexity of the model is bound to affect the accuracy of detection. The ShipDetectionNet finds the best tradeoff between accuracy, the number of operations, and the number of parameters by adding SE blocks. We will evaluate where SE blocks are added in this part.

To explain the method of this article conveniently, we divide the whole network into three parts. The first part is the S-Conv of layer 1, which does not participate in the discussion; the second part is 2–15 layers, which is composed of ShipDetection Unit(a); and the third part consists of 16–20 layers and is composed of ShipDetection Unit(b).

To evaluate the impact of the location of SE blocks on the experimental results, we set up four cases: 1) ShipDetectionNet1, where SE blocks are added to the second part and the third part; 2) ShipDetectionNet2, where SE blocks are removed from the model structure; 3) ShipDetectionNet3, where the SE block is only added to the second part; and 4) ShipDetectionNet4, where the SE block is only added to the third part, as shown in Table IV.

In Table IV, R represents for recall, Para and G are parameters and GFLOPs, respectively, and T refers to the inference time of each image. From Table IV, we can see that adding SE blocks reasonably will improve detection accuracy, but it has a slight negative impact on the detection speed. Although the second and third parts of ShipDetectionNet1 all add SE blocks, the mAP value is the lowest, indicating that adding too much attention mechanism increases the redundancy of the network to play the opposite role. ShipDetectionNet4 with SE blocks added only in the third part can generate satisfactory results with the mAP of 83.03%, and the number of parameters, GFLOPs, and inference time decreased compared with ShipDetectionNet3 with SE blocks added only in the second part. Therefore, ShipDetectionNet4 is chosen as the ShipDetectionNet proposed in this article.

The evaluation is carried out on the testing set of satellite remote sensing ship images to verify the generalization performance of the ShipDetectionNet. From Table IV, the recall for ShipDetectionNet4 is 83.44%, the precision is 90.97%, and the mAP value is 83.03%. In addition, the average inference time of each image is 10.66 ms, which means only 1.7 s is needed to complete the detection of 160 images in the testing set. Interestingly, some ashore ships, small ships, and ships in cloudy weather in complex scenes can also be successfully detected on this dataset, which indicates that the ShipDetectionNet is robust.

F. Tests on the FAIR1M Dataset

We trained several networks on a new dataset FAIR1M [48] and tested the models. FAIR1M is a large dataset for fine-grained target detection and recognition in remote sensing images. The image contains rich geographic information and panchromatic and several multispectral bands. Each image is of the size in the range from 1000×1000 to 10 000×10 000 pixels and contains objects exhibiting a wide variety of scales, orientations, and shapes. This dataset contains five categories and 37 subcategories. However, according to our own tasks, 1000 images containing ships were selected to constitute our dataset, which was randomly divided into the training set, verification set, and test set in a ratio of 7:2:1. Since the annotation of this dataset is oriented bounding box, we converted the data annotation into horizontal bounding box and canceled the classification of ships to adapt to our own tasks. To ensure the consistency of experimental variables, we embedded the following lightweight networks into the framework of YOLOv5, enabling them to complete the object detection task. The batch size of each network was set to 4, and 300 epochs were trained. The default values of the hyperparameters of the original network were kept. The experimental results are as follows.

It can be seen that our network achieves the optimal results in both recall and mAP in Table V. It does not perform well in precision, but there is room for improvement. Both the large and small versions of MobileNetv3 achieve high-precision results, which may be due to its network architecture search (NAS) strategy. Find the optimal parameter collocation through NAS to build the optimal network. The inspiration for us is that although the realization of NAS requires the support of certain experimental equipment, we can try more combinations of various experimental parameters in future experiments to find the optimal network architecture. The low recall values of MobileNet and ShuffleNet lead to their low mAP values, because

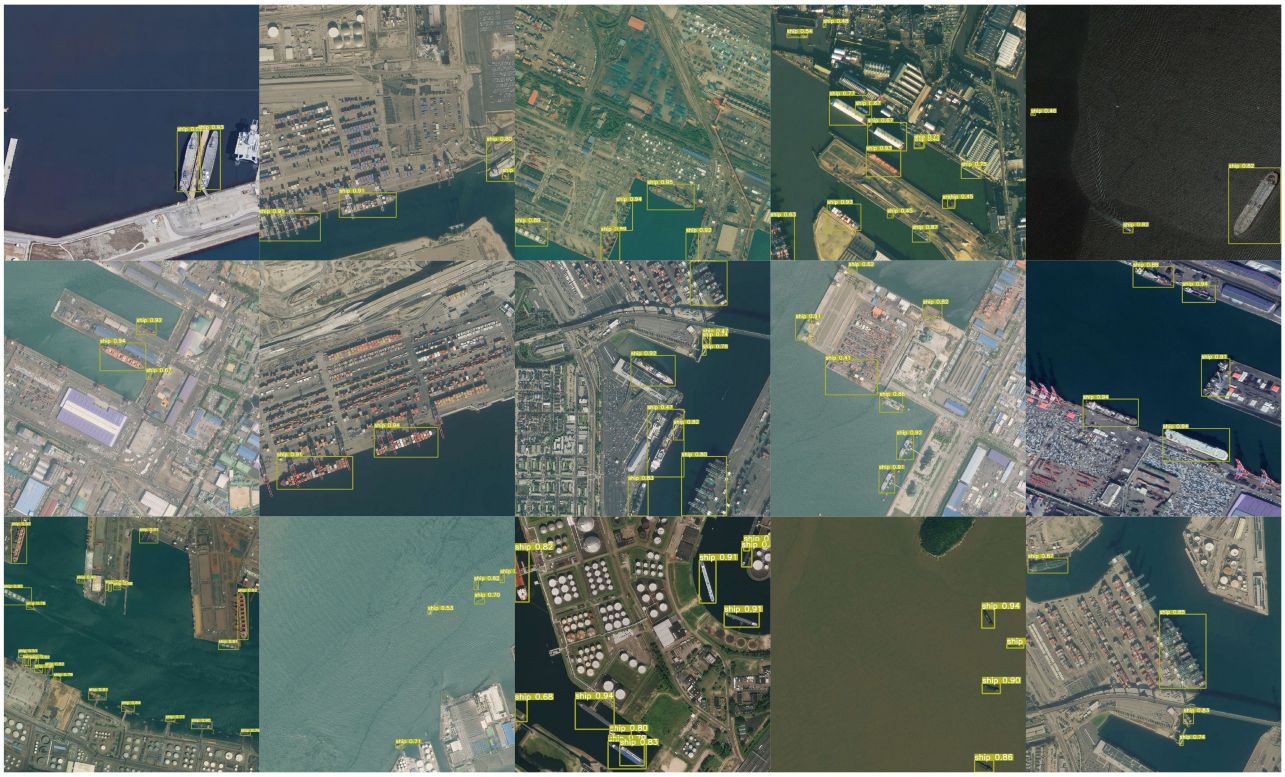


Fig. 10. Test results of the ShipDetectionNet in the FAIR1M dataset.

the number of their parameters is too small, and the network is not sufficient to extract and learn features. Considering the tradeoff between performance and efficiency, our network is more suitable for the ship detection task. In future work, we will continue to study fine-grained ship detection, detecting different types of ships to better adapt to practical applications.

Fig. 10 shows the detection effect of our network on the FAIR1M dataset, and it can be seen that the ShipDetectionNet can still achieve a pretty detection effect in this relatively complex scene. In addition, we can observe that the detection effect of small ships is not as good as that of large ships. This is still caused by continuous downsampling. Excessive downsampling rate makes the pixels occupied by small objects gradually decrease or even disappear. Although multiscale fusion is used in our network, it may not be enough according to the detection results. One of our future ideas is to add a multiscale fusion branch to the shallow layer of the network to preserve as many pixels of the small target as possible. Meanwhile, we also considered using some data enhancement strategies to increase the proportion of small targets.

V. Conclusion

In this article, we propose an enhanced network based on YOLOv5, called ShipDetectionNet, to balance the detection performance and the model complexity. We take the DS-Conv to replace S-Conv and further use the pointwise G-Conv to replace the point convolution in the DS-Conv, which further reduces the number of parameters and computational cost of the network. Meanwhile, to ensure the detection accuracy of the network, we also add SE blocks. Based on these, we build two types of network convolution units, ShipDetection Unit(a) and ShipDetection Unit(b), and use them to structure the ShipDetectionNet. Compared with 11 state-of-the-art baselines and five lightweight models, the results show the superiority of the proposed method. And the ShipDetectionNet is evaluated on different scenarios, demonstrating its generalizability and effectiveness.

To make the ship detection more effective, reliable, and robust, we will consider the influence of climate on the detection images, such as haze, rain, and low lighting. Owing to the small amount of data in the harsh environment, the expansion of the dataset can be considered, such as artificially creating some fog scenes. After continuous attempts, we believe that our model will be more superior.

References

[1] Y. Mao, Y. Yang, Z. Ma, M. Li, H. Su, and J. Zhang, "Efficient low-cost ship detection for SAR imagery based on simplified U-Net," *IEEE Access*, vol. 8, pp. 69742–69753, 2020.

[2] D. J. Crisp, "The state-of-the-art in ship detection in synthetic aperture radar imagery," *Defence Sci. Technol. Organisation Salisbury*, Edinburgh, SA, Australia, Tech. Rep. DSTO-RR-0272, 2004.

[3] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sens. Environ.*, vol. 207, pp. 1–26, 2018.

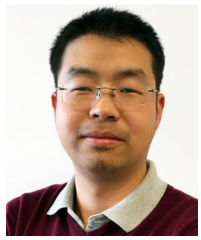
[4] A. Jiaqiu *et al.*, "A novel ship wake CFAR detection algorithm based on SCR enhancement and normalized Hough transform," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 4, pp. 681–685, Jul. 2011.

[5] G. Gao, "A Parzen-window-kernel-based CFAR algorithm for ship detection in SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 557–561, May 2011.

[6] X. Junyi, J. Kefeng, L. Lin, and D. Chun, "Ship target detection from optical satellite remote sensing image based on GLRT," *Remote Sens. Technol. Appl.*, vol. 27, no. 4, pp. 616–622, 2012.
[7] Z. Xiaoling, Z. Tianwen, S. Jun, and W. Shunjun, "High-speed and high-accurate SAR ship detection based on a depthwise separable convolution neural network," *J. Radars*, vol. 8, no. 6, pp. 841–851, 2019.
[8] C. Wang, F. Bi, L. Chen, and J. Chen, "A novel threshold template algorithm for ship detection in high-resolution SAR images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2016, pp. 100–103.
[9] J. Zhu, X. Qiu, Z. Pan, Y. Zhang, and B. Lei, "Projection shape template-based ship target recognition in TerraSAR-X images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 2, pp. 222–226, Feb. 2017.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.
[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.
[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.
[13] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.
[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
[15] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7263–7271.
[16] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[17] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.
[18] X. Xu, X. Zhang, and T. Zhang, "Lite-YOLOv5: A lightweight deep learning detector for on-board ship detection in large-scene sentinel-1 SAR images," *Remote Sens.*, vol. 14, no. 4, 2022, Art. no. 1018.
[19] S. Liu *et al.*, "Multi-scale ship detection algorithm based on a lightweight neural network for spaceborne SAR images," *Remote Sens.*, vol. 14, no. 5, 2022, Art. no. 1149.
[20] P. Xu *et al.*, "On-board real-time ship detection in HISEA-1 SAR images based on CFAR and lightweight deep learning," *Remote Sens.*, vol. 13, no. 10, 2021, Art. no. 1995.
[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
[22] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, vol. 28, pp. 91–99.
[24] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
[25] 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.
[26] 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. 9627–9636.
[27] F. Yang, Q. Xu, B. Li, and Y. Ji, "Ship detection from thermal remote sensing imagery through region-based deep forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 449–453, Mar. 2018.
[28] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, "HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 12, pp. 7147–7161, Dec. 2018.
[29] H. Changhua, C. Chen, and H. Chuan, "Ship small target detection based on deep convolution neural network in SAR image," *J. Chin. Inertial Technol.*, vol. 27, no. 3, pp. 397–405, 2019.
[30] R. W. Liu, W. Yuan, X. Chen, and Y. Lu, "An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system," *Ocean Eng.*, vol. 235, 2021, Art. no. 109435.
[31] N. Wang, B. Li, X. Wei, Y. Wang, and H. Yan, "Ship detection in spaceborne infrared image based on lightweight CNN and multisource feature cascade decision," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4324–4339, May 2021.
[32] J. Yu, X. Peng, S. Li, Y. Lu, and W. Ma, "A lightweight ship detection method in optical remote sensing image under cloud interference," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, 2021, pp. 1–6.
[33] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[34] L. Chen, W. Shi, and D. Deng, "Improved YOLOv3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images," *Remote Sens.*, vol. 13, no. 4, 2021, Art. no. 660.
[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7132–7141.
[36] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[37] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1251–1258.
[38] C. Jia *et al.*, "Semantic segmentation with light field imaging and convolutional neural networks," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021, Art. no. 5017214.
[39] 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.
[40] 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.
[41] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5987–5995.
[42] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[43] 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.
[44] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2961–2969.
[45] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "VarifocalNet: An IoU-aware dense object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8514–8523.
[46] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13039–13048.
[47] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet v2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.
[48] X. Sun *et al.*, "FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery," *ISPRS J. Photogrammetry Remote Sens.*, vol. 184, pp. 116–130, 2022.

Yifan Yin received the B.S. degree in computer science in 2019 from the School of Computer Science and Technology, Tianjin University of Technology, Tianjin, China, where she currently working toward the M.S. degree.

Her current research interests include artificial intelligence, deep learning, computer vision, and object detection.



Xu Cheng (Member, IEEE) received the Ph.D. degree in engineering from the Department of Ocean Operations and Civil Engineering, Intelligent Systems Laboratory, Norwegian University of Science and Technology, Ålesund, Norway, in 2020.

In 2022, he joined the Smart Innovation Norway, Halden, Norway, where he is currently a Permanent Researcher. He has authored or coauthored more than 30 papers. His research interests include data analysis and artificial intelligence in maritime operations, time-series analysis, and predictive maintenance of wind turbines.

Fan Shi (Member, IEEE) received the Ph.D. degree in optics from Nankai University, Tianjin, China, in 2012.

He is currently an Associate Professor with the Tianjin University of Technology, Tianjin. From June 2018 to August 2019, he was a Research Scholar with West Virginia University, Morgantown, WV, USA. His research interests include machine vision, pattern recognition, and optics.

Meng Zhao received the B.S. degree in automation in 2010 and the M.S. and Ph.D. degrees in control science and engineering in 2016, all from Tianjin University, Tianjin, China.

Since 2016, she has been an Associate Professor with the School of Computer Science and Engineering, Tianjin University of China, Tianjin. From May 2019 to April 2020, she was a Postdoctoral Researcher supported by the European Research Consortium for Informatics and Mathematics "Alain Bensoussan Fellowship Programme." Her research interests include medical image processing, medical/biomedical engineering, and machine learning/deep learning in medical informatics.

Guoyuan Li (Senior Member, IEEE) received the Ph.D. degree in computer science from the Department of Informatics, Institute of Technical Aspects of Multimodal Systems, University of Hamburg, Hamburg, Germany, in 2013.

In 2014, he joined the Intelligent Systems Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Ålesund, Norway, where he is currently a Professor of Ship Intelligence. He has authored or coauthored more than 80 articles. His research interests include modeling and simulation of ship motion, autonomous navigation, intelligent control, optimization algorithms, and locomotion control of bioinspired robots.

Shengyong Chen (Senior Member, IEEE) received the Ph.D. degree in robot vision from the City University of Hong Kong, Hong Kong, in 2003.

He is currently a Professor with the Tianjin University of Technology, Tianjin, China. He received a fellowship from the Alexander von Humboldt Foundation of Germany and worked with the University of Hamburg, Hamburg, Germany, from 2006 to 2007. He has authored or coauthored more than 200 scientific papers in international journals, with 80 in IEEE transactions. He is an inventor of more than 100 patents. He organized more than 20 international conferences. His research interests include computer vision, robotics, and machine intelligence.

Dr. Chen received the National Outstanding Youth Foundation Award of China in 2013. He is an Associate Editor for several international journals, including IEEE TRANSACTIONS ON CYBERNETICS. He is a Fellow of the Institute of Engineering and Technology and a Distinguished Member of the China Computer Federation.