Multi-Size Convolution and Learning Deep Network for SAR Ship Detection From Scratch

LONG HAN1, DA RAN2, WEI YE2, WEIDONG YANG3, AND XU WU4

1Graduate School, Space Engineering University, Beijing 101416, China
2School of Space Information, Space Engineering University, Beijing 101416, China
3Jiuquan Satellite Launch Center, Jiuquan 735400, China
4Unit 32359 of PLA, Xining 810000, China

Corresponding author: Long Han (hello_long9372@163.com)

This work was supported by the Science and Technology on Complex Electronic System Simulation Laboratory under Grant DXZF-JC-ZZ-2017-007.

ABSTRACT Synthetic aperture radar (SAR) ship detection is a popular branch of SAR interpretation. A growing number of scholars are devoting themselves to applying convolutional neural network (CNN) to SAR ship detection. Currently, most CNN-based SAR ship detectors are variants of object detectors in optical images; however, the essential differences between SAR and optical images restrict their performance. To this end, by focusing on the attribute of SAR image’s ‘point’ property which is determined by its imaging mechanism, we design a novel SAR ship detector from scratch. The innovatively designed PCB-MSK (parallel convolutional block of multi-size kernels) consists of two groups of convolutions, each group is composed of four convolutional layers corresponding to kernel sizes of 3, 5, 7, and 9; the stride is 1 for one group and 2 for another. In the designed convolutional module with features reused (CMFR), the output and input feature maps of the previous block are concatenated for current layer to reduce information loss during forward propagation and to strengthen the supervision for shallow layers during parameter optimization. For each source prediction layer, the binary classification is first conducted to alleviate the positive/negative imbalance; deconvolution and feature fusion are utilized to enhance the feature representation. Then, we perform fine detection. Experiments on RDISD-SAR and SSDD, in which RDISD-SAR is meticulously constructed by us based on two open-access datasets, show that our method achieves a state-of-the-art accuracy and competitive speed, the average precision (AP) reaches 88.70% and 90.57% for RDISD-SAR and SSDD, respectively. These APs are 10.43% and 4.23% higher than DSOD, and 6.64% and 1.70% higher than ScratchDet. The detection speed is 58.2 FPS on a GTX 1080Ti GPU, the number of parameters is 18.19M and the amount of computations is 21.33G. In addition, experimental results show that the robustness of our detector is very excellent.

INDEX TERMS Ship detection, CNN, SAR dataset, scratch.

I. INTRODUCTION Synthetic aperture radar (SAR), which has an all-day imaging and reconnaissance capability, is an indispensable and important monitoring tool in the field of remote sensing. This technology plays an irreplaceable role in many civil and military applications, such as marine fishery management, ocean environment protection, handling abnormal sea situations, maritime navigation monitoring and control, and key target reconnaissance and surveillance in water. The strong application demand has greatly promoted the development and progress of related technologies, and SAR ship detection in remote sensing is now receiving increasing attention [1]–[11]. Although many achievements have been published in the past few decades, there are still many challenges to address, such as ship detection in complex scenarios (e.g., inshore, offshore, and inland river environments) and detection of ships of various sizes, densely arranged ships, and defocused SAR ships.
In recent years, especially the last 5 years, the CNN has shown a powerful ability for object detection and image feature representation. Region-based approaches, such as the very famous Faster RCNN series [12]–[15], and regression-based methods, such as the YOLO series [16]–[19] and SSD [20], have achieved state-of-the-art performance in both accuracy and efficiency in object detection. Due to their excellent detection performance, these data-driven algorithms and their variants are being introduced to object detection in remote sensing images, such as optical remote sensing images and SAR images, to improve their accuracy, efficiency, and robustness. However, the most popular and best performing CNN-based detectors always contain millions of learnable parameters, which means that training such a detector demands large numbers of bounding box-level annotated images. To achieve good performance without large numbers of annotated images, most existing SAR ship detectors are developed based on transfer learning; they make use of pretrained backbones to initialize the parameters of the networks and then fine-tune the detectors with SAR datasets. There are three benefits to transfer learning-based SAR ship detectors: 1) fast convergence, 2) accurate and stable detection performance, and 3) the small demand for annotated SAR images. Besides, there are some common features extracted by some layers between SAR images and optical images, the transfer-learning based methods have some advantages in this regard.

Despite these benefits, there are also three problems that have negative effects on further performance improvements [21]: 1) limited space for designing a backbone to initialize a network via a pretrained model; their structures should be almost the same. Moreover, pretrained models are always achievements of ImageNet-based classification tasks, but they are too heavy to train from scratch with very limited annotated SAR images. Thus, there is little flexibility to deeply optimize the network architecture. 2) The learning bias whereby the differences between image classification and object detection in loss functions and categories may lead to different gradient search and parameters optimization spaces. 3) Domain mismatches such that a SAR image greatly differs from images from ImageNet in terms of the vision, image attributes, and statistical distribution. Furthermore, ships in SAR images are also very different from the objects in optical images in terms of their size, background, surroundings, resolutions, and properties.

A promising solution to the above three problems is to train a SAR ship detector from scratch. However, it is difficult to conduct such research without numerous accurately annotated datasets of SAR images for ship detection. Fortunately, J. Li et al. [22], Y. Wang et al. [23], and S. Xian et al. [25] released three SAR datasets for ship detection in November 2017, April 2019, and December 2019, respectively, namely, SSDD, SAR-Ship-Dataset [24], and AIR-SARShip-1.0. These open-access datasets greatly assist in the development of CNN-based SAR ship detection and lay the foundation for this study. (Because AIR-SARShip-1.0 consists of 31 images of large scenes and 487 annotated ships, which are not enough ships to train the CNN, we conduct our experiments using the first two datasets only.)

Most existing CNN-based SAR ship detectors are designed based on powerful backbones from general object detection. Although this procedure is feasible and performs well, in view of the essential difference between SAR images and natural optical images in their principle of imaging, it is necessary to study the unique properties of SAR image that are significantly different from those of general images to further improve the detection performance. Based on our literature investigation, there are relatively few studies on this subject. Ma et al. [26] transferred the original single-channel SAR image to a three-channel image via a 2-D Fourier transformation-based low-pass filter to fit the traditional CNN’s input shape. Wang et al. [27] proposed a hierarchical CNN to detect ships from coarse to fine to overcome the unique ghost phenomenon in SAR images. By the sparse optimization and reconstruction performed in [28], the sidelobe in SAR images is suppressed and the image quality is improved to achieve a better detection performance.

Taking the essential differences between SAR images and optical images into consideration, this paper proposes a novel, fast, precise, and robust CNN-based SAR ship detection approach that focuses on the unique “point” attribute of a SAR image that can be trained from scratch. The detector mainly consists of four modules: a shared stem, a parallel convolutional block of multi-size kernels (PCB-MSK), a convolutional module with features reused (CMFR), and a feature enhancement and fusion block (FEFB). The overall detection frame is shown in Fig. 1. In addition, we crafted a large, rich, and diverse dataset named the reference dataset of SAR images for ship detection (RDISD-SAR) based on SSDD and SAR-Ship-Dataset. Experiments on RDISD-SAR and the original SSDD demonstrate that the proposed approach reaches the state-of-the-art ship detection performance compared with other trained-from-scratch detectors and achieves greater accuracy than competitive mainstream pretraining-based detectors. Moreover, the proposed method has varying superiority over the existing detectors in terms of the number of parameters and the amount of computations. The main contributions of this paper are as follows:

1. In view of the “point” attribute of a SAR image, we innovatively design a PCB-MSK module to fit the special property. It consists of two groups of convolutions, and each group is composed of four convolutional layers corresponding to kernel sizes of 3, 5, 7, and 9. The stride is 1 for one group and 2 for another. Besides, the efficient CMFR is designed to extract multi-scale feature maps of SAR image together with the PCB-MSK.

2. We develop a novel SAR ship detection algorithm that can be easily trained from scratch. Our algorithm achieves state-of-the-art detection performance, and the APs of our detector are 88.70% and 90.57% on RDISD-SAR and SSDD, respectively, the detection speed is 58.2 FPS.
3. We reconstruct a well-crafted dataset for SAR ship detection, i.e., RDISD-SAR, based on two open-access datasets. Experiments show that RDISD-SAR is more ideal than SSDD in terms of the volume, variety, and subset settings.

The rest of this paper is arranged as follows: related works are presented in Section II, the proposed approach is described in detail in Section III, experiments and analyses are described in Section IV, and Section V and VI present the discussion and conclusion, respectively. Section VII is our future work.

II. RELATED WORK

Here, we briefly introduce the related CNN-based SAR ship detection methods and representative object detection approaches. Over the last 5 years, CNN-based SAR ship detection has received impressive improvements. The related achievements can be roughly divided into two groups, namely, region-based methods and regression-based methods.

A. REGION-BASED SAR SHIP DETECTION METHODS

The detection process of these methods consists of two stages. A set of sparse candidate proposals are generated in the first stage; then, after classifying these proposals into different categories, the representatives of this group are Faster RCNN, RFCN, Cascade RCNN [29], and their variants. Under the basic detection framework, scholars have proposed many improvements to achieve a better SAR ship detection performance.

Kang et al. [30] fully integrated and used the context information of multilayer feature maps, which not only improves the detection performance for small objects but also reduces false and missed detections. Li et al. [22] transferred Faster R-CNN to SAR ship detection and employed a feature fusion strategy. These authors also developed more efficient training tricks; above all, they created and opened a SAR ship detection dataset (SSDD), which was the first open-access dataset for SAR ship detection both at home and abroad. Li et al. [31] applied the pretrained deep residual network ResNet-50 to SAR ship detection. Cui et al. [32] introduced
the attention mechanism and dense connections to a feature pyramid network (FPN [33]) and embedded a dense attention pyramid network (DAPN). Thus, these authors fused a feature map with a convolutional block attention module (CBMA) that consists of channel and spatial attention, and they also utilized dense connections to reduce the information loss. Jiao et al. [34] designed a multiscale CNN-based ship detector with dense connections based on the Faster R-CNN framework and the deeper ResNet-101 and added a feature fusion module between layers with different receptive fields to enhance representations of small, inshore, and offshore ships. Gui et al. [35] embedded a feature fusion layer between the second and the fourth residual modules. In addition, these authors designed a light detection subnet based on separable convolution and position-sensitive RoI (PSRoI) and took the focal loss [36] as the loss function to help train the detector.

Most of the above approaches were developed by directly utilizing the pretrained backbones and predesigned network structure and embedding a feature enhancement module or introducing better training tricks and algorithms to strengthen the quality of the feature representation to improve the detection performance. Although some progress has been made, the problems we mentioned in Section I remain prominent.

B. REGRESSION-BASED SAR SHIP DETECTION

These approaches directly predict the locations and categories of objects from feature maps of different scales. The use of feature pyramids and a multiscale prediction structure leads to varying improvements in both accuracy and speed compared to the previous methods. Based on the detection framework, researchers have conducted many studies on SAR ship detection.

Chang et al. [37] applied YOLOv2 to SAR ship detection after streamlining the original model. These authors reduced the number of output channels at each layer of the backbone, which significantly accelerated the computing efficiency without impairing the detection performance. With MobileNet [38] as the backbone and inspired by YOLO’s idea, [39] directly divided the input image into $s \times s$ $(s=7)$ grids to improve the speed; they detected large, middle, and small ships from the 13th, 26th, and 52nd detection scale, respectively. The introduction of separable convolution led to fewer parameters and did not damage the feature extraction. Specifically for small ships, Yang et al. [40] fused feature maps of shallow layers of SSD via dilated convolution and $1 \times 1$ convolution to enhance a ship’s features according to the context information. The pretrained VGG-16 and DarkNet-19 were transferred to the research. Chen et al. [41] took a group convolution that came from Inception [42]–[44] into ResNet-101 and combined it with the attention mechanism to enhance the representation of small ships in complex scenarios. During training, the loss function based on generalized intersection over union (GIoU) [46] was used to guide the parameters update. In the test phase, soft-NMS [47], which can reduce the missing detection for overlapped objects, was used to suppress the redundant bounding boxes that belonged to the same ship. Su et al. [48] proposed a transfer learning and feature fusion-based method in which the pretrained ResNet-50 was used as its backbone and the multibranch parallel convolution and feature fusion were utilized to enhance the features of the layers with sizes of $38 \times 38$, $10 \times 10$, and $5 \times 5$. Furthermore, Wang et al. [49] applied RetinaNet to SAR ship detection in complex scenarios.

The abovementioned studies also follow the pretrained backbone and predesigned network architecture and adopt a variety of feature fusion and feature enhancement strategies to improve the feature representation of ships in an SAR image. In addition, effective training algorithms and tricks are employed to improve the detection performance. However, some researchers have devoted themselves to designing trained-from-scratch object and ship detectors to solve the problems we described in Section I.

Zhang et al. [50] and Zhang et al. [51] introduced depthwise separable convolution into SAR ship detection and designed two high-speed detectors. Zhang et al. [52] designed a lightweight SAR ship detector with only 20 convolution layers based on depthwise separable convolution. Mao et al. [53] developed an efficient and low-cost anchor-free SAR ship detector based on the simplified U-Net. Zhang et al. [54] designed a lightweight and feature-optimized network from scratch. These authors followed the SSD framework in which feature reuse and concatenation were adopted to fuse feature maps from the half-channel-VGG16. Moreover, the attention-mechanism-based optimization strategy was embedded in the network, the number of parameters of the method is only one-quarter of SSD while the AP is 3% higher than it. Deng et al. [55] constructed a SAR ship detection algorithm from scratch, and it consisted of a backbone, ship proposal network (SPN), and ship discrimination network (SDN). The backbone was similar to the DenseNet of DSOD. In addition, the SPN filtered its input feature maps by kernels with sizes of $1 \times 1$, $3 \times 3$, $5 \times 5$, and $7 \times 7$ to generate proposals; then, the SDN conducted fine detection in which PSRoI and average pooling were adopted. In the training phase, the focal loss was employed as the loss function to address the imbalance between positive and negative examples. Overall, there are very few SAR ship detectors that can be trained from scratch, and there is still a gap in the detection performance between these approaches and transfer learning-based ones.

As seen from this brief review, both region-based methods and regression-based methods (including pretraining-based and trained-from-scratch methods) mainly focus on optimizing the network, training algorithms and tricks. The unique and essential attributes of a SAR image are seldom taken into consideration. Therefore, we conducted this research and designed an accurate, fast, and robust CNN-based detector from scratch.

III. THE PROPOSED DETECTOR

A. MOTIVATIONS

The great success of VGGNet [56] on the famous ImageNet proves that using a small convolutional kernel can
result in more accurate detection than a large kernel. Furthermore, the number of parameters and computations can be greatly reduced, which leads to less time cost. For example, three kernels that have sizes of $3 \times 3$ include 27 weight parameters, while a $9 \times 9$ kernel includes 81 parameters. However, the performances of three cascaded small kernels are similar to or better than that of a $9 \times 9$ kernel when extracting features. Since the referenced work was published, the structure of massive cascaded $3 \times 3$ kernels has been widely used in many excellent networks to reduce the parameters.

However, SAR images and optical images are essentially different in imaging theory, where the difference is reflected in terms of their statistical and visual characteristics. A significant discrepancy is that the pixels belonging to the same object in an optical image are continuous locally, that is, these local pixels have special attributes of a “plane” or a “line” (a “plane” property in a homogeneous region and a “line” property in edges and outlines). In contrast, local pixels in SAR images often do not have such continuity, as the pixels in SAR image are indexes of the intensity of the microwave reflected back to the radar, and a pixel corresponds to a scattering point of an object. In other words, pixels in SAR images have the attribute of “points”, as shown in Fig. 2 and Fig. 3, the attribute of SAR image mainly results from the fact that under existing hardware conditions, the frequency of SAR’s working wave band is much lower than the frequency of light, which results in an object being a set of discrete scattering points for SAR. Because the small size of a $3 \times 3$ convolutional kernel is very delicate in filtering pixels locally, when dealing with images with such “mutation” pixels in a homogeneous region, the mutation pixels greatly affect the output of currently layer. Furthermore, this problem negatively affects feature extraction, as shown in Fig. 4(a). In contrast, large kernels can sense images in a larger range and are less affected by “mutation” pixels, such as those shown in Fig. 4(b)(c)(d), than small kernels. Taking the special attribute of a SAR image into consideration and motivated by Inception series [42]–[44] and [45] in which each inception module is consists of convolutional layers with kernel size of $1 \times 1$, $1 \times 3$, $7 \times 1$, et al., a parallel structure of convolutional layers with $3 \times 3$, $5 \times 5$, $7 \times 7$, and $9 \times 9$ kernels is designed in the shallow layers to reduce the negative effect that “mutation” pixels have on feature extraction and thereby improve the detection accuracy.

It is known that large convolutional kernels mean a sharply increasing number of parameters. However, in our previous studies, we found that there are some all-zero matrices in feature maps from hidden layers of different resolutions, which means that these feature maps make no contribution to the object detection results. Hu et al. [57] revealed that although this type of redundancy of a CNN cannot be avoided, it can be reduced. Moreover, for a single-polarized SAR image, its pixels only indicate the echo signal strength of the scattering point; the image only contains intensity information and does not contain any other information, e.g., the color information of an optical image. Taking these two points into consideration, we take appropriate measures to reduce the number of output channels and thereby decrease the number of parameters. Moreover, we employ the feature reuse strategy to improve the efficiency. This strategy cannot greatly increase the number of parameters, but it makes full use of shallow feature maps.
FIGURE 4. Schematic illustration of the effects that “mutation” pixels have on the output. In each of the four figures, we have an input feature map, a convolutional kernel, and an output feature map. From left to right, the black lines with arrows at the end represent the direction of the convolutional kernel sliding. Inside the red box, we have the input feature map, while the dashed lines are the padding pixels.

B. THE PROPOSED BACKBONE

The backbone is one of the most important parts of a CNN-based object detection algorithm, as it is responsible for extracting feature maps that are beneficial for object detection from the original input image via cascaded convolutional layers and some other layers. For transfer learning-based detectors, to stably fine-tune the model with the SAR dataset and make it converge quickly, the model of the pretrained weight that has been trained with ImageNet is often employed to initialize the CNN, e.g., ZFNet, VGG-16, and ResNet use this procedure. Although these pretrained models help SAR ship detectors perform well, there are still many issues that need to be studied, as we indicated in the introduction. Thus, designing a backbone that can be trained from scratch is the right way to address these problems.

Studies performed using DSOD and ScratchDet [58] reveal that one should follow these principles to design a trained-from-scratch backbone: proposal-free, stem, deep supervision, dense prediction, and BatchNorm. Based on these five criteria and motivated by the DS/64-192-48-1 of DSOD and Root-res34 in ScratchDet, which we combine with our analysis of a SAR image’s “point” attribute in the previous section, we design a novel backbone for SAR ship detection.

The newly designed backbone is named MSK-3579-1/2, where “MSK” is the acronym for multi-size kernel, “3579” indicates the sizes of the four convolutional kernels in order, and “1/2” refers to strides of 1 and 2, respectively. In total, MSK-3579-1/2 is composed of three modules: the first module is the shared stem, and we add a feature concatenation layer and an extra convolutional layer to the traditional stem which consists of three cascaded convolutional layers, as shown in Fig. 5 and Fig. 6. The shared stem module is followed by a “wide” PCB-MSK block that shares the output of the previous layer and contains 8 convolutional layers in which the strides of four of them are set to 1, while a stride of 2 is assigned to another 4 layers. The kernel sizes of each of the 4 layers are $3 \times 3$, $5 \times 5$, $7 \times 7$, and $9 \times 9$, and for convenience, we use the expression ConvXX_1 to indicate them; the remaining 4 layers are expressed as ConvXX_2 ($X = 3, 5, 7, 9$). After a $2 \times 2$ MaxPooling layer, the output of ConvXX_1 is concatenated with the output of ConvXX_2 by channel. The third part is a convolutional module with features reused (CMFR) and consists of a stack of cascaded convolutional layers, i.e., Conv2/3/4/5/6 with output sizes $80 \times 80$, $40 \times 40$, $20 \times 20$, $10 \times 10$, and $5 \times 5$, respectively. By concatenating the feature maps of the current layer and the previous layer by channel, the efficiency of the feature utilization is improved; furthermore, this procedure is beneficial to improving the detection accuracy.

In Table 1, we provide the key details for each layer of the proposed MSK-3579-1/2, and the output size, MACC (multiply-accumulate for a convolution layer; its formula is $\text{MACC} = K_w \times K_h \times C_{in} \times H_{out} \times W_{out} \times C_{out}$, where $K_w, K_h$ are the width and height of the convolutional kernel, respectively; $C_{in}$ refers to the number of channels of the input feature maps; and $H_{out}, W_{out},$ and $C_{out}$ are the height, width, and number of channels of the output feature maps, respectively; the MACC is a metric of the number of computations), and number of parameters (#Params) are also listed.

![FIGURE 5. Traditional stem.](image)

![FIGURE 6. Proposed stem.](image)
TABLE 1. Details of the proposed backbone MSK-3579-1/2.

| Block Name | Layer Name | Operation | Layer params | Output Shape | MACC | #Params |
|------------|------------|-----------|--------------|--------------|------|---------|
|            |            |           | (k, s, p)    | (M=10^6, G=10^5) | (k=10^5, M=10^6) |
|            | Conv1_1    | Conv      | [3, 2, 1]    | 64 x 160 x 160 | 44.24M | 1.73k   |
|            | Convl_2    | Conv      | [3, 1, 1]    | 64 x 160 x 160 | 943.72M | 36.86k  |
|            | Convl_3    | Conv      | [3, 1, 1]    | 64 x 160 x 160 | 943.72M | 36.86k  |
|            | Concat1    | Concatenate, (Conv1_1, Conv1_2, Conv1_3) | 64 x 160 x 160 | ~ | ~ |
|            | Convl_4    | Conv      | [3, 1, 1]    | 64 x 160 x 160 | 1.89G   | 73.73k  |
| PCB-MSK    | Pool1      | MaxPooling | k=2, s=2    | 64 x 80 x 80 | 58.98M | 9.22k   |
|            | Convl32    | Conv      | [3, 2, 1]    | 16 x 80 x 80 | 163.84M | 25.60k  |
|            | Convl52    | Conv      | [5, 2, 2]    | 16 x 80 x 80 | 321.13M | 50.18k  |
|            | Convl72    | Conv      | [7, 2, 3]    | 16 x 80 x 80 | 530.84M | 82.94k  |
|            | Convl99    | Conv      | [9, 2, 4]    | 16 x 80 x 80 | 530.84M | 82.94k  |
|            | Concat2    | Concatenate, (Conv33_2, Conv55_2, Conv77_2, Conv99_2) | 64 x 80 x 80 | ~ | ~ |
|            | Concat3    | Concatenate, (Pool1, Concat2) | 128 x 80 x 80 | ~ | ~ |
|            | Convl3_in_branch1 | Conv | [3, 2, 1]    | 128 x 40 x 40 | 943.72M | 589.82k |
|            | Convl2     | Conv      | [3, 1, 1]    | 128 x 80 x 80 | 3.77G   | 589.82k |
|            | Convl2     | Conv      | [3, 1, 1]    | 128 x 80 x 80 | 943.72M | 147.46k |
|            | Pool2      | MaxPooling | k=2, s=2    | 128 x 40 x 40 | 58.98M | 9.22k   |
|            | Convl4_in_branch1 | Conv | [3, 2, 1]    | 128 x 20 x 20 | 58.98M | 147.46k |
|            | Convl3_in | Concatenate, (Conv3_in_branch1, Pool2) | 256 x 40 x 40 | ~ | ~ |
|            | Convl3     | Conv      | [3, 1, 1]    | 256 x 40 x 40 | 943.72M | 589.82k |
|            | Convl2     | Conv      | [3, 1, 1]    | 256 x 40 x 40 | 943.72M | 589.82k |
|            | Pool3      | MaxPooling | k=2, s=2    | 256 x 20 x 20 | 589.82k |
|            | Convl4_in_branch2 | Conv | [1, 1, 0]    | 256 x 20 x 20 | 13.11M | 32.77k  |
|            | Convl5_in_branch1 | Conv | [3, 2, 1]    | 256 x 10 x 10 | 589.82M | 589.82k |
|            | Convl4_in | Concatenate, (Conv4_in_branch1, Conv4_in_branch2) | 256 x 20 x 20 | ~ | ~ |
| CMFR       | Convl1     | Conv      | [3, 1, 1]    | 256 x 20 x 20 | 235.93M | 589.82k |
|            | Convl2     | Conv      | [3, 1, 1]    | 256 x 20 x 20 | 235.93M | 589.82k |
|            | Convl3     | Conv      | [3, 1, 1]    | 256 x 20 x 20 | 235.93M | 589.82k |
|            | Concat4    | Concatenate, (Conv4_1, Conv4_3) | 512 x 20 x 20 | ~ | ~ |
|            | Pool4      | MaxPooling | k=2, s=2    | 512 x 10 x 10 | ~ | ~ |
|            | Convl5_in_branch2 | Conv | [1, 1, 0]    | 512 x 10 x 10 | 13.11M | 131.07k |
|            | Convl5_in | Concatenate, (Conv5_in_branch1, Conv5_in_branch2) | 512 x 10 x 10 | ~ | ~ |
|            | Convl5     | Conv      | [3, 1, 1]    | 256 x 10 x 10 | 117.96M | 1.18M   |
|            | Convl5     | Conv      | [3, 1, 1]    | 256 x 10 x 10 | 58.98M  | 589.82k |
|            | Convl5     | Conv      | [3, 1, 1]    | 256 x 10 x 10 | 58.98M  | 589.82k |
|            | Concat5    | Concatenate, (Conv5_1, Conv5_3) | 512 x 10 x 10 | ~ | ~ |
|            | Convl6     | Conv      | [3, 1, 1]    | 256 x 10 x 10 | 13.11M  | 131.07k |
|            | Convl6     | Conv      | [3, 2, 1]    | 512 x 5 x 5  | 29.49M  | 1.18M   |
| FEBF       | Convl3_tmp1 | Conv | [3, 1, 1]    | 256 x 40 x 40 | 943.72M | 589.82k |
|            | Convl3_tmp2 | Conv | [3, 1, 1]    | 256 x 40 x 40 | 943.72M | 589.82k |
|            | Convl4_tmp1 | Conv | [3, 1, 1]    | 256 x 20 x 20 | 235.93M | 589.82k |
|            | Convl4_tmp2 | Conv | [3, 1, 1]    | 256 x 20 x 20 | 235.93M | 589.82k |
|            | Convl5_tmp1 | Conv | [3, 1, 1]    | 256 x 10 x 10 | 58.98M  | 589.82k |
|            | Convl5_tmp2 | Conv | [3, 1, 1]    | 256 x 10 x 10 | 58.98M  | 589.82k |
|            | Convl6_tmp1 | Conv | [3, 1, 1]    | 256 x 5 x 5   | 29.49M  | 1.18M   |
|            | Convl6_tmp2 | Conv | [3, 1, 1]    | 256 x 5 x 5   | 14.75M  | 589.82k |
|            | Deconv3    | Deconv    | [2, 0, 2]    | 256 x 40 x 40 | 104.86M | 262.14k |
|            | Deconv3    | Deconv    | [3, 1, 1]    | 256 x 40 x 40 | 943.72M | 589.82k |
|            | Deconv4    | Deconv    | [2, 0, 2]    | 256 x 20 x 20 | 26.21M  | 262.14k |
TABLE 1. (Continued.) Details of the proposed backbone MSK-3579-1/2.

| Layer   | Output   | Width (px) | Height (px) | MACC (M) | FLOPs (G) |
|---------|----------|------------|-------------|----------|-----------|
| Conv4   | Conv     | [3, 1, 1]  | 256×20×20   | 235.93M  | 589.82k   |
| Deconv5 | Deconv   | [2, 0, 2]  | 256×10×10   | 6.55M    | 262.14k   |
| Conv5   | Conv     | [3, 1, 1]  | 256×10×10   | 58.98M   | 589.82k   |

Note: Input: 3×320×320. All of the Conv X X parts consist of convolution, BatchNorm, Scale, and ReLU; k refers to the kernel size (both width and height); s is the stride; p represents the padding; MACC is a measure of the computation. Total MACC: 21.33G, Total #Params: 18.19M.

C. ANCHORS REFINEMENT

The one-stage detection algorithms directly generate a large number of anchors of predefined shapes and scales on the feature maps by sliding windows. Then, these algorithms locate the object and classify its category via bounding box regression and class regression for all the anchors. However, the region of interest that contains the objects generally only requires a few areas of the image, as the remaining area is almost always background or uninteresting objects. This fact leads the majority of anchors to contain no objects, i.e., negative examples, while anchors that may overlap with the ground truth, i.e., positive examples, only represent a small part of the anchors. Because numerous negative examples will cause the backward propagation algorithm to pay more attention to them when updating the parameters, these examples adversely affect the final detection performance. Two-stage detectors do not have such an imbalance because the region proposal network (RPN) first selects a sparse set of proposals that may contain some objects and filters out regions that do not contain any objects and then performs fine detection on these proposals. Thus, this paper draws on the idea of RefineDet and adopts a method of detection from coarse to fine to eliminate the imbalance. Furthermore, this paper improves the detection accuracy.

1) BINARY CLASSIFICATION OF ANCHORS

Since ships in SAR images are always sparse generally speaking, the pixels belonging to a ship in a SAR image account for only a very small part of the whole image. The vast majority of the regions of a SAR image belong to the background. This pattern leads to a serious imbalance between negative examples and positive ones that are densely produced by sliding windows. To mitigate this problem, we first classify all the anchors into two categories, i.e., objects or background, as shown in Fig. 7. Because this operation easily filters out many regions of background, the imbalance is greatly mitigated. Consequently, the model pays more attention to the positive examples in the training phase, which will result in improvements of the object location and class regression.

2) FEATURE ENHANCEMENT AND FUSION

Generally, the pyramid structure of a backbone is unable to extract features of objects with various sizes. Especially for small ships in large scenarios, the type of network may easily lead to features of small objects not being significant enough for detection, and they even disappear sometimes. To this end, we design a feature enhancement and fusion block (FEFB) to help remedy this situation, as shown in Fig. 8. We make full use of deconvolution to enlarge the size of feature maps of different layers from deep to shallow in order, and we fuse the enlarged feature maps with the outputs of the corresponding layers of the backbone one by one. Finally, we conduct ship prediction on the fused feature maps of multi-scales.

3) FINE DETECTION

In this module, the sparse set of candidate anchors that are achieved via anchor binary classification are mapped to the output of FEFB to conduct fine location and classification, i.e., bounding box regression and class regression. Due to the large numbers of negative examples having been filtered out and because the features of ships, in general, and small ships, in particular, are very significant, we achieve an excellent detection performance in complex scenarios and for ships of various sizes.

D. THE OVERALL DETECTION FRAMEWORK

On the basis of the above design, we develop a complete SAR ship detection framework, as shown in Fig. 1. This framework mainly consists of six modules, i.e., the shared stem, PCB-MSK, CMFR, FEFB, anchors binary classification, and fine detection. First, any input image is resized to 320×320; then, some data augmentation strategies, such as the strategies used in RefineDet and SSD, are adopted to enrich the inputs, e.g.,
random clipping, horizontal and vertical mirroring, or brightness, contrast, and saturation adjustments with a probability of 0.5. Next, an image is fed into the proposed MSK-3579-1/2 in which the outputs of Conv3_3, Conv4_3, Conv5_3, and Conv6_2 are selected as the detection source layers. The detection head is similar to RefiDet and SSD in that it initially densely generates sets of anchors using sliding windows according to the predefined scales and ratios; then, all of the anchors are fed into the binary classification and FEFB. At this point, we conduct fine detection; finally, the non-maximum suppression (NMS) algorithm is applied to select the best bounding box from a set of detection output.

Note that to ensure the stability of the training, in addition to convolution and ReLU, all of the “Conv” units in our model are composed of BatchNorm and Scale, and the “inplace” property is set to “True” for all the layers. Among these layers, BatchNorm refers to the normalization operation, Scale refers to the translation and scaling operations, and ReLU means the activation function. The corresponding calculation formulas are as follows:

\[
\begin{align*}
    x_{\text{BN}} &= \frac{x_i - \mu_{\text{Batch}}}{\sigma^2_{\text{Batch}}} \\
    x_{\text{Scale}} &= \gamma \cdot x_{\text{BN}} + \beta \\
    \mu_{\text{Batch}} &= \frac{1}{N_{\text{Batch-size}}} \sum_{i=1}^{N_{\text{Batch-size}}} x_i \\
    \sigma^2_{\text{Batch}} &= \frac{1}{N_{\text{Batch-size}}} \sum_{i=1}^{N_{\text{Batch-size}}} (x_i - \mu_{\text{Batch}})^2 \\
    x_{\text{ReLU}} &= \max\{0, x_{\text{in}}\}
\end{align*}
\]

where \(i = 1, \ldots, N_{\text{Batch-size}}\), \(x_i\) is the i-th image of the current batch; \(\mu_{\text{Batch}}\) and \(\sigma^2_{\text{Batch}}\) are the average value and variance, respectively; \(\gamma\) and \(\beta\) are the translation and scaling factors (both of which are learnable parameters and are updated during training), respectively; and \(x_{\text{in}}\) is the value of the pixels in the current feature map.

In Fig. 1, the symbol “F” means fusion. For the shared stem module, the feature fusion mode is an element-wise sum, while the three “F” processes in FEFB are implemented via an element-wise dot product. All of the 5 “C” symbols refer to concatenation of the input feature maps by channel.

The loss function is the traditional multitask loss function RefiDet, which we will not describe in detail here. All of the convolutional kernels are initialized via Xavier, while the biases are initialized with a constant.

IV. EXPERIMENTS AND RESULTS

In this section, we describe many experiments that were performed to verify the effectiveness of the proposed backbone and detection framework. In addition, we describe some ablation experiments that were conducted to compare our detector and some other well-performing SAR ship detection approaches.

Our experiments were carried out on a server configured with an NVIDIA GeForce GTX 1080 Ti, 11 GB of memory, NVIDIA driver version 430.64, CUDA v8.0, and CUDNN v7.1. The CPU was an Intel(R) Xeon(R) CPU E5-1620 v2 @ 3.70 GHz × 8, and the operating system was Ubuntu 16.04. The proposed method was implemented based on the Caffe framework [59] and the basic detection framework is SSD. During training, some key hyperparameters were set as follows: momentum = 0.9, \(\gamma = 0.1\), weight decay = 5e-4, max_iter = 50000, and base_lr = 0.25. Moreover, the learning rate decayed to 1/10, 1/100, and 1/1000 at iterations 30000, 40000, and 45000, respectively, and the optimizer was SGD.

A. DATASETS

SSDD [22] is the first open-access dataset of SAR images for ship detection, and it is widely used by SAR researchers both at home and abroad. However, there is no division for the training subset, test subset, and validating subset. Most of its users, including Dr. Jianwei Li, set the three subsets via random division in a certain proportion, e.g., 7:2:1. There is an obvious drawback to such a division scheme: it is not conducive to comparing the ship detection performances of various related detectors. Dr. Yuanyuan Wang of the University of the Chinese Academy of Science (UCAS) developed another, larger dataset [23] of SAR images for ship detection, i.e., SAR-Ship-Dataset [24]. Unfortunately, this dataset also fails to set the subsets division scheme. Considering the influence of the data distribution on the detection performance and the convenience of comparing the performances of various approaches, we integrate the two datasets together and conduct some comprehensive and scientific processing. Then, we acquire our experimental dataset, which is called RDISD-SAR.

1) DESCRIPTION AND ANALYSIS OF THE SOURCE DATASETS

The images of SSDD come from three sources: the Yantai Port of Radarsat-2, the India Visakhapatnam Port of TerraSAR-X, and Sentinel SAR images of the European Space Agency (ESA). With four types of polarization, namely, HH, HV, VH and VV, and resolution ranges of 1 m-15 m, the SSDD contains a total of 1160 images of 2540 ships in various scenarios (SSDD’s author published 2456 ships; to enrich the dataset in this paper, some SSDD images are replaced with our images). These images vary in size and have an average size of 400 × 400 pixels. Each image is a slice of the original SAR image, and there are 2.19 ships in each image on average. Compared with SSDD, SAR-Ship-Dataset is much larger: there are a total of 43819 images with a fixed size of 256 × 256 pixels in the dataset, the number of ships in the images ranges from 1-32, and the dataset’s authors labeled a total of 59535 ships in it. There is an average of 1.36 ships per image, and the original SAR images come from Sentinel-1 and Gaofen-3.

There are two common drawbacks of the random division scheme: (1) it is unable to guarantee the consistency of the dataset distributions of the training subset, the validation subset, and the test subset, which may cause a well-trained...
detector to develop a "preference" whereby the detector performs excellently on some types of images and objects but performs poorly on detecting other types of images; (2) a detection preference may lead to the accuracy achieved with the test subset to not be representative; hence, it may not reflect the true performance, especially in terms of robustness. Thus, we reconstructed the SSDD and SAR-Ship-Dataset datasets in terms of the inherent properties of ships, general image assessments, and SAR image assessments. We divided all of the images in the two datasets into four classes (class1~class4) from the "simplest" situation to the most "complex" situation. The classification was made by at least 6 experienced SAR experts. The principles followed for the work and some descriptions of each class follow:

- **Class1**: The images in the dataset are quite simple, and the background of each image is a pure sea surface and does not contain land, islands, or any other type of scatterer. The sea surface is almost free of wind and waves. In a few images, there may be a slight wake or other types of slow seawater flow, which visually appear as weak noise (specifically, speckle and sea clutter) and are hard to perceive by human eyes. Each of the images contains one or more ships, and some of them are not ideally imaged, i.e., there are micro-motion and defocusing effects in these SAR images. The number of images in class1 is 19490, and it accounts for 45.6% of all of the images.

- **Class2**: The images in the dataset are relatively simple, the background of each image is also a pure sea surface and does not contain land or islands except for a few man-made scatterers. In all of the images, there are a variety of levels of wind and waves on the sea. Wakes or other types of seawater flows are contained in some images. There is some noise of varying intensity in the images, e.g., speckle, sea clutter, and other sources of noise that are apparent to human eyes. Each image contains one or more ships. In some images, although the man-made scatterers are relatively significant, their intensity is far less than the intensity of the ships. A few ships are not ideally imaged because of their micro-motion, which results in defocusing. The number of images in class2 is 15262, and it accounts for 35.7% of all of the images.

- **Class3**: The images in the dataset are relatively complex. In addition to the sea surface, the background of each image includes land, islands, and other types of man-made scatterings. In the images, some objects are very similar to the ship. It is very difficult to distinguish ships from other objects, and we cannot determine whether a pixel belongs to the object of interest or to the background. Most of the images contain more than one ship, and a few of these ships are imaged poorly. The number of images in class3 is 7024, and it accounts for 16.4% of all of the images.

We divided all of the images of class4 into four levels according to image noise (sea clutter and speckle). In addition, we divided the images of class4 into four levels in light of image complexity. The complete dataset tree is shown in Fig. 9. In particular, we found that there are two serious issues to be aware of in SAR-Ship-Dataset. One issue is that some images in SAR-Ship-Dataset are in the single-channel format; images in this format may result in unexpected errors during training and testing because a three-channel input image is required for a detector; hence, we converted all of those images into the three-channel format. The other issue is that we found that there are a few labels with severely distorted position information for the ships in the XML files of SAR-Ship-Dataset. The labeled coordinates of the ships are highly inconsistent with their real positions. In this paper, we call these images dirty data. There are five main types of such inaccurate annotations: (1) an image is labeled as containing one or more ships, but there are no ships in the image; (2) there is a severe deviation between the labeled position and the real position of a ship; (3) not all of the ships in an image are labeled; (4) the labeled bounding box is too large to surround a ship compactly; and (5) an object that cannot be confirmed to be a ship is labeled as a ship. Representative examples of dirty data are shown in Fig. 10.
L. Han et al.: Multi-Size Convolution and Learning Deep Network for SAR Ship Detection From Scratch

2) SUBSET SETTINGS

On the basis of the image classification, 10000 images are selected as the training, test, and validation subsets, and their components are shown in Fig. 11. We emphasize that inasmuch as the ships in class1 and class 3 are relatively easy to discriminate, they only account for a small portion of RDISD-SAR. Comparatively speaking, it is very difficult to distinguish the ships in class2 and class4. To improve both the ability of the detectors to adapt to various complex scenarios and their robustness, the majority of our dataset comes from class2 and class4, which have the same distributions in the training, test, and validation subsets.

There is only one category of objects, i.e., a ship, in RDISD-SAR, and the number of images reaches 10000. Compared with PASCAL VOC (VOC2012 trainval contains 11540 images, 20 categories, and 27540 objects) and MS COCO (MS COCO2015 contains 165482 images in 91 categories), the training set has 81208 images, and the validation set and test set have 81434 images between them), there are enough images in our dataset for training and testing.

B. EVALUATION METRICS

Generally, the time cost and detection accuracy are two indicators used to evaluate an object detector. Because there is only one type of object (i.e., a ship) labeled in our datasets, we take the average precision (AP) to evaluate the detection performance. The AP is defined by

$$AP = \int_{0}^{1} P(R)dR$$

where P represents the precision and R represents the recall. These terms are defined as follows:

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}$$

where TP (True Positive) is the number of objects that are correctly detected, FP (False Positive) is the number of objects that are incorrectly detected, and FN (False Negative) is the number of objects that should have been detected. From the above definitions, we can draw a conclusion that the larger the AP, i.e., the larger the area under the P-R curve, the better the performance will be. In the calculations, the recall takes discrete values in [0, 1].

The time cost (T, ∼ms) for detecting each image is expressed as how much time it takes to process the image, and we also take the FPS (Frames Per Second) to indicate the detection speed:

$$FPS = \frac{1}{T}$$

C. EXPERIMENTS

1) ABLATION EXPERIMENTS

In this section, we employ the control variates to analyze and verify the effectiveness of our novel designs, including the structure of PCB-MSK, the feature aggregation mode in the shared stem, and the anchors refinement.

a: PARALLEL CONVOLUTIONS IN PCB-MSK

The kernel sizes of this module are 3 × 3, 5 × 5, 7 × 7, and 9 × 9, and we performed three groups of experiments to verify their effectiveness. In group one, the convolutional stride was set to 1, i.e., ConvXX_s1; in group two, the stride was set to 2, i.e., ConvXX_s2; in the last group, convolutions with stride 1 and stride 2 were configured at the same time, i.e., we used ConvXX_s1s2, and their output feature maps were concatenated by channel. The experimental results are listed in Table 2, and it is clear that ConvXX_s1s2 reaches the highest AP. In addition, we also conducted experiments without PCB-MSK. After training under the same settings with (c), the loss value remains large and is no longer reduced. Then, we validated the performance of this configuration on the RDISD-SAR test subset, and we found that the AP was less than 5.0%. Because we repeated the same experiment three times and obtained the same results, we believe that PCB-MSK is effective in the proposed approach. Besides, we employed dilated convolution to achieve the same reception fields with
TABLE 2. Ablation experimental results of the proposed approach with RDID-SAR.

| Shared Stem | PCB-MSK | Refine | P  | #Params | MACC  |
|-------------|---------|--------|----|---------|-------|
|             | C(1,2)  | C(1,2,3) | S1 | S2 | S1S2 | R | AP | (M) | (G) |
| (a)         | ✓       |         |    |      |       | 92.38% | 52.77% | 9.08 | 12.80 |
|             |         |         |    |      |       | 82.38% |       |      |      |
| (b)         |         | ✓       |    |      |       | 92.33% | 53.21% | 9.08 | 9.57  |
|             |         |         |    |      |       | 82.64% |       |      |      |
| (c)         |         | ✓       |    | ✓   | ✓     | 91.98% | 58.46% | 9.40 | 14.47 |
|             |         |         |    |      |       | 83.22% |       |      |      |

multi-size kernels in PCB-MSK, e.g., Conv55_1(k=5, s=1, p=2) has the same reception field with Conv55_1(k=3, s=1, p=2, dilation =2), the experimental results on RDID-SAR are: P=91.28%, R=87.34%, AP=88.34%, Time =17.8ms, #Params=17.93%, and MACC=17.14G, although the detection accuracy is better than RefineDet trained with RDID-SAR, it is slightly worse than the PCB-MSK we proposed in terms of AP and Time.

b: FEATURE AGGREGATION IN SHARED STEM

The traditional and most widely used stem always consists of three cascaded convolutional layers, i.e., Conv1_1, Conv1_2, and Conv1_3. Taking into account the noise characteristics and the complex backgrounds of SAR images, we embedded an extra feature aggregation layer to increase the feature utilization efficiency. The experimental results of three configurations of the shared stem are listed in Table 2. Obviously, the concatenation of Conv1_1, Conv1_2, and Conv1_3 obtains the best detection accuracy of 83.22%, which contributes AP gains of 1.33% and 1.28% compared with C(1,3) and C(2,3), respectively.

c: EFFECTIVENESS OF ANCHORS REFINEMENT

In (a) and (b), the anchors refinement has not been adopted yet, that is, we directly conducted the bounding box regression and class regression from densely produced anchors. The detection head was the same as in SSD. In (c), the network is configured with FEFB and fine detection, the experimental results listed in Table 2 illustrate that the FEFB contributes an AP gain of 4.62% to the proposed method, and the best-configurated version reaches an AP of 88.70%.

2) COMPETITIVE DETECTORS

To verify the performance of the proposed method more comprehensively, we performed a comparison study with related state-of-the-art detectors, Faster RCNN, RFCN, Cascade RCNN, YOLOv3, YOLOv4, RetinaNet, SSD, DSSD, RefineDet, RFNet [60], M2Det [61], DSOD, and ScratchDet were implemented to help evaluate the detection performance. In addition, we trained RefineDet and SSD300 from scratch to achieve a deep comparison between our detector and the classical detectors (in fact, we are trying our best to train all the pre-trained-based detectors above from scratch, but only experiments of SSD and RefineDet are successful so far); all of them were trained from scratch with an extra BatchNorm layer embedded into each convolutional layer.

The APs of the competitive detectors given above and the proposed method on RDID-SAR and SSDD are listed in Table 3. Their time cost for detection, numbers of parameters, and MACCs are also listed.

3) EXPERIMENTS ON SSDD

Because extensive studies on SAR ship detection are almost entirely conducted with SSDD, to compare the detection performance of the proposed method with other related studies, we conducted the same experiments with SSDD in which the training, test, and validation subsets were randomly divided at a ratio of 7:2:1. The APs are also listed in Table 3. Note that we conducted three pairs of experiments for both RFNet and M2Det with SSDD. Although the configurations were exactly the same as in the experiments with RDID-SAR, the APs were less than 1.00% for each of them, so we believe that these experiments failed and we do not list their APs in Table 3.

D. EXPERIMENTAL RESULTS AND ANALYSIS

We performed many SAR ship detection experiments with both RDID-SAR and SSDD. The results are shown in Table 3, Fig. 13, Fig. 14, Fig. 15, Fig. 16, and Fig. 17.

First, we analyze these experimental results from the perspective of AP. It is obvious that the proposed method reaches the best AP of 88.70%, which is 0.41% higher than RefineDet and 3.07% higher than RetinaNet, which is the best-performing region-based detector. Compared with the trained-from-scratch detectors, we observe a remarkably superior accuracy of the proposed detector; in fact, it is 10.43% and 6.64% higher than the accuracies of the classical DSOD and ScratchDet, respectively, even 4.26% higher than SSD300 trained from scratch; besides, the time cost of our method is only 17.2ms per image which is significantly faster than the others. Furthermore, although the pretraining-based RefineDet has the best performance among the remaining detectors, the AP of the trained-from-scratch RefineDet is only 77.48%, which is 11.22% lower than our result. Comparatively, the AP of the trained-from-scratch SSD300 reaches 84.44%, which is 0.46 higher than the
TABLE 3. Performances of related detectors.

| No. | Detectors      | Backbone      | Pretrained | Input size | AP (RDSD-SAR) | SSDD | Time (ms) | #Params (M) | MACC (G) |
|-----|----------------|---------------|------------|------------|---------------|------|-----------|------------|----------|
| E1  | Faster RCNN    | VGG16         | ✓          | 224x224    | 77.10%        | 79.80% | 48.3      | 136.69     | 51.69    |
| E2  | RFCN           | ResNet101     | ✓          | 600x1000   | 76.42%        | 49.21% | 52.0      | 49.74      | 136.62   |
| E3  | Cascade RCNN   | VGG16         | ✓          | 600x1000   | 83.70%        | 41.91% | 63.0      | 350.65     | 192.59   |
| E4  | RetinaNet      | ResNet50      | ✓          | 500x500    | 85.63%        | 84.70% | 76.0      | 36.35      | 52.40    |
| E5  | YOLOv3         | DarkNet53     | ✓          | 416x416    | 86.51%        | 86.57% | 12.0      | 61.52      | 32.76    |
| E6  | YOLOv4         | CSPDarkNet53  | ✓          | 416x416    | 87.23%        | 90.34% | 16.5      | 64.36      | 30.17    |
| E7  | SSD300         | VGG16         | ✓          | 300x300    | 83.98%        | 88.62% | 13.5      | 23.75      | 30.43    |
| E8  | SSD512         | VGG16         | ✓          | 512x512    | 81.16%        | 87.99% | 24.5      | 23.75      | 87.55    |
| E9  | DSSD           | ResNet101     | ✓          | 321x321    | 77.65%        | 88.70% | 67.0      | 158.99     | 46.03    |
| E10 | RefineDet      | VGG16         | ✓          | 320x320    | 88.29%        | 90.59% | 16.5      | 33.91      | 37.49    |
| E11 | RFBNet         | VGG16         | ✓          | 300x300    | 81.83%        | ~      | 18.0      | 33.82      | 34.93    |
| E12 | M2Det          | VGG16         | ✓          | 320x320    | 49.84%        | ~      | 673.0     | 80.21      | 55.76    |
| E13 | SSD300+BN      | VGG16+BN      | ×          | 300x300    | 84.44%        | 86.06% | 21.6      | 23.75      | 30.43    |
| E14 | RefineDet+BN   | VGG16+BN      | ×          | 320x320    | 77.48%        | 90.10% | 28.5      | 33.91      | 37.49    |
| E15 | DSOD           | DS/64-192-48-1 | ×       | 300x300    | 78.27%        | 86.34% | 34.0      | 12.50      | 14.08    |
| E16 | ScratchDet     | Root-ResNet-34 | ×   | 300x300    | 82.06%        | 88.87% | 23.0      | 23.41      | 32.89    |
| E17 | Ours           | MSK-3579-1/2  | ×          | 320x320    | 88.70%        | 90.57% | 17.2      | 18.19      | 21.33    |

FIGURE 12. Representative examples of each class.

FIGURE 13. Comparison of the APs of related detectors.

The pretraining-based SSD300. Although this detector has the best performance among the trained-from-scratch detectors, there is an AP gap of 4.26% between it and our detector. The excellent performance of our method can also be seen from the experimental results with SSDD. In this case, the AP reaches 90.57%, which is only 0.02% lower than the pretraining-based RefineDet’s 90.59%, but is much more precise than any other detectors trained with SSDD.

For an input of $3 \times 320 \times 320$, our detector takes only 17.2 ms to finish detecting, and its detection speed is 58.2 FPS. Among the other detectors, YOLOv3 is the fastest one, as its speed is as high as 83.3 FPS; the pretraining-based SSD300 is also much quicker, with a speed of 74.1 FPS. Compared with the trained-from-scratch detectors, the detection speeds of DSOD, ScratchDet, SSD300, and RefineDet are 29.4 FPS, 43.5 FPS, 46.3 FPS, and 29.5 FPS, respectively. There is no doubt that our detector achieved a very fast detection speed.

The number of parameters of our detector is 18.19M, which is 5.69M more than DSOD’s 12.5M and 5.22M less than ScratchDet’s 23.41M; it is also approximately 53.64%
of RefineDet’s figure and less than the figures for most of the other detectors. The MACC of our detector is only 21.33G, which is less than the figure for any other detector except DSOD’s 14.08G. It is obvious that both the number of parameters and the number of computations of our detector are less than most of the other competitive detectors. For detectors that need to load a pretrained model to initialize the parameters, we successfully conducted all of the experiments with both RDISD-SAR and SSDD, where their base learning rate (base_lr) was set to 5e-4. We also tried some other settings that were greater than 5e-4, but failed (loss=NAN). Excitingly, when base_lr was set to 0.25, the proposed detector was still successful and stable. In fact, we attempted more values of base_lr from 0.001 to 0.25, and all of them were successful. The final AP of every test was roughly 88.70%.

From the results and analysis above, it can be seen that there is significant superiority of the proposed approach in terms of the accuracy, the speed, the number of parameters, and the quantity of computations. To further improve and optimize our model in the future, we briefly analyze the number of parameters and MACCs of the main modules, as shown in Table 4.

It is obvious that the distribution of #Params and MACC are not consistent, that is, more parameters do not precisely correspond to a matching MACC. For example, the number of parameters of the shared stem only accounts for 1.03% of the total parameters, while its MACC comprises 22.36% of the model. In addition, PCB-MSK takes no more than 2.00% of the parameters and one-quarter of the MACC, and Conv2-Conv6 take nearly half of the parameters and one-third of the MACC. The parameters from FEFB are another main source of the total parameters, as they comprise nearly half of the parameters but only one fifth of the MACC.
**F. ROBUSTNESS ANALYSIS**

Robustness is one of the most important items for SAR ship detection, as we always hope that a detector can adapt to various situations. For ships in SAR, the background may be simple, involve various scenarios, be complex, be inshore, be offshore, or have other variations. Furthermore, the ships are multiscale. An excellent detector should be able to correctly predict the locations and categories of most objects. Thus, we conducted more SAR ship detection experiments based on the four classes of images we classified in Section 3.1, i.e., Class1, Class2, Class3, and Class4, to evaluate the performance of our...
Most of the detectors perform very well for Class1, which mainly consists of simple images. The APs are almost as high as 90.00%, and the highest one is YOLOv4’s 95.18% (trained with RDISD-SAR). Hence, it is difficult to accurately distinguish the real performances of various detectors, and these results are not representative. More complex images yield lower detection accuracies. The results with Class4 are the worst for most of the detectors, and the APs are relatively small; the highest one is RefineDet’s 86.78% (trained with RDISD-SAR), and the smallest one is YOLOv3’s 17.91%.

**FIGURE 17.** Ship detection results (trained with SSDD).
TABLE 5. APs of detectors with respect to the different classes of SAR images.

| No. | Detector            | Training set | Test set   | RDISD-SAR | SSDD | Class-1 | Class-2 | Class-3 | Class-4 |
|-----|---------------------|--------------|------------|-----------|------|---------|---------|---------|---------|
| E1-1| Faster RCNN         | RDISD-SAR    | 77.10%     | 68.89%    | 89.65%| 80.47%  | 87.16%  | 72.19%  |
| E1-2| Faster RCNN         | SSDD         | 38.36%     | 81.19%    | 69.14%| 41.05%  | 47.40%  | 23.54%  |
| E2-1| R-FCN               | RDISD-SAR    | 76.42%     | 60.39%    | 93.87%| 90.10%  | 87.04%  | 67.46%  |
| E2-2| R-FCN               | SSDD         | 41.75%     | 50.20%    | 84.51%| 61.76%  | 56.19%  | 29.52%  |
| E3-1| Cascade RCNN        | RDISD-SAR    | 74.27%     | 53.45%    | 92.93%| 81.68%  | 82.21%  | 69.12%  |
| E3-2| Cascade RCNN        | SSDD         | 31.83%     | 44.96%    | 78.73%| 39.52%  | 37.52%  | 22.03%  |
| E4-1| RetinaNet           | RDISD-SAR    | 85.63%     | 85.20%    | 93.76%| 93.34%  | 95.64%  | 85.16%  |
| E4-2| RetinaNet           | SSDD         | 54.42%     | 84.70%    | 82.20%| 67.22%  | 67.54%  | 33.73%  |
| E5-1| YOLOv3              | RDISD-SAR    | 86.51%     | 88.32%    | 91.70%| 93.21%  | 96.37%  | 84.86%  |
| E5-2| YOLOv3              | SSDD         | 33.34%     | 86.57%    | 63.78%| 44.71%  | 29.57%  | 17.91%  |
| E6-1| YOLOv4              | RDISD-SAR    | 87.23%     | 89.07%    | 95.18%| 95.32%  | 93.50%  | 85.26%  |
| E6-2| YOLOv4              | SSDD         | 48.34%     | 90.34%    | 80.35%| 67.56%  | 46.92%  | 26.66%  |
| E7-1| SSD300              | RDISD-SAR    | 83.94%     | 82.09%    | 90.02%| 89.23%  | 88.26%  | 80.84%  |
| E7-2| SSD300              | SSDD         | 82.71%     | 88.37%    | 89.07%| 87.05%  | 68.71%  | 41.68%  |
| E8-1| SSD512              | RDISD-SAR    | 80.72%     | 80.28%    | 89.62%| 88.78%  | 85.78%  | 71.25%  |
| E8-2| SSD512              | SSDD         | 81.23%     | 88.40%    | 88.20%| 84.65%  | 66.64%  | 38.84%  |
| E9-1| DSSD                | RDISD-SAR    | 77.65%     | 79.37%    | 89.71%| 88.11%  | 83.66%  | 65.22%  |
| E9-2| DSSD                | SSDD         | 84.42%     | 88.26%    | 90.02%| 82.793 | 67.26%  | 36.87%  |
| E10-1| RefineDet           | RDISD-SAR    | 88.29%     | 90.24%    | 90.29%| 90.05%  | 89.93%  | 86.78%  |
| E10-2| RefineDet           | SSDD         | 87.62%     | 90.59%    | 87.60%| 86.64%  | 67.94%  | 44.73%  |
| E11-1| RFBNet              | RDISD-SAR    | 81.83%     | 79.01%    | 90.14%| 89.05%  | 89.34%  | 78.77%  |
| E12-1| M2Det               | RDISD-SAR    | 49.80%     | 59.30%    | 82.81%| 74.58%  | 54.14%  | 37.16%  |
| E13-1| SSD300 (Scratch)    | RDISD-SAR    | 84.44%     | 81.05%    | 89.78%| 89.51%  | 89.08%  | 82.04%  |
| E13-2| SSD300 (Scratch)    | SSDD         | 79.70%     | 85.16%    | 88.55%| 86.51%  | 58.91%  | 36.41%  |
| E14-1| RefineDet (Scratch) | RDISD-SAR    | 77.48%     | 81.76%    | 89.51%| 87.58%  | 83.73%  | 59.38%  |
| E14-2| RefineDet (Scratch) | SSDD         | 56.80%     | 90.10%    | 89.38%| 76.24%  | 56.90%  | 37.22%  |
| E15-1| DSOD                | RDISD-SAR    | 78.25%     | 74.90%    | 89.59%| 88.32%  | 84.75%  | 66.24%  |
| E15-2| DSOD                | SSDD         | 79.66%     | 85.65%    | 88.86%| 87.35%  | 68.04%  | 42.41%  |
| E16-1| ScratchDet          | RDISD-SAR    | 81.72%     | 76.46%    | 89.82%| 89.00%  | 87.71%  | 76.56%  |
| E16-2| ScratchDet          | SSDD         | 86.16%     | 88.60%    | 88.85%| 85.56%  | 63.90%  | 36.53%  |
| E17-1| Ours                | RDISD-SAR    | **88.70%** | 89.99%    | 90.09%| 90.12%  | 90.42%  | 86.65%  |
| E17-2| Ours                | SSDD         | 61.07%     | 90.57%    | 88.45%| 75.91%  | 62.25%  | 38.07%  |

V. DISCUSSION

Since SAR images have both the common properties of general images and their own attributes, taking some of SAR’s particularities into consideration when designing a SAR ship detection algorithm is beneficial and achieves a better performance. In this paper, we performed a worthwhile study from a novel perspective, i.e., the “point” attribute of a SAR image. The large number of experimental results demonstrate that the proposed method has a competitive performance in terms of the detection accuracy, the speed, the number of (trained with SSDD), which is 68.87% lower than the corresponding figure. Such huge differences of APs suggest that for various detectors, detecting ships in complex SAR images is more difficult than ship detection in simple SAR images. The AP of our detector reaches 86.63% in Class4, which is higher than the rest of competitive detectors except RefineDet’s 86.78% (trained with RDISD-SAR). Furthermore, it can be seen from Table 5 that the detection performance of the proposed approach trained with RDISD-SAR in Class1 ∼ Class4 remains more accurate than most of the other detectors, which illustrates that the proposed detector is very robust.

More detection results of our detector are shown in Fig. 18 and Fig. 19. Our detector can easily predict ships in the simple scenario with strong noise, very small ships, inshore ships, ships in complex scenarios, and even defocused ships. The predicted bounding box is very close to the ship’s real position; furthermore, the confidence score is sufficiently high. After contrasting Fig. 18 and Fig. 19, we can conclude that detectors trained with RDISD-SAR perform significantly better than detectors trained with SSDD.
parameters, and the quantity of computations compared with most mainstream detectors regardless of whether they are region-based or regression-based. Above all, our detector can be trained from scratch and eliminate the pretrained model’s limitations, which brings great convenience and allows us to design algorithms more flexibly.

As train-from-scratch SAR ship detectors, both ours and Ref. [55] employ multi-size kernels to improve the detection performance, the difference between them is that [55] added extra convolutional layers with kernels of $3 \times 3$, $5 \times 5$, $7 \times 7$ in the deep layer based on the DenseNet of DSOD for multi-branches prediction, however, we designed the PCB-MSK in the shallow layer of the network to deal with the “point” attribute of SAR images. Numbers of experiments we conducted and in [55] show that such design is very effective for SAR ship detection.

Compared with the datasets of optical images, e.g., PASCAL VOC and MS COCO, the image number of SSDD is not small. Because there is no division among the training, test, and validation subsets, researchers usually randomly divide the three subsets in their studies, including SSDD’s authors. This practice causes two problems, as we described above. As a result, we generated an elaborate reference dataset of SAR images for ship detection. The experimental results on both RDISD-SAR and SSDD show that our dataset is more able to evaluate various SAR ship detectors; thus, RDISD-SAR is better than SSDD for SAR ship detection.

VI. CONCLUSION

In this paper, we focus on designing an excellent train-from-scratch SAR ship detector from the perspective of a SAR image’s “point” attribute. First, the feature fusion is embedded in the traditional stem to reduce the feature loss in shallow layers during forward propagation. Then, 8 parallel convolution layers with kernel sizes of $3$, $5$, $7$, and $9$ and strides of 1 and 2 are designed to adapt the “point” attribute. After the outputs of PCB-MSK are concatenated by channel, they are fed into the following module. The cross-layer feature-reuse structure is adopted in five cascaded convolution blocks of our backbone, and the output and input features of the previous layer are concatenated by channel. Then, we input these features into the current layer to increase
the efficiency of the use of information. We use multiscale prediction to detect ships of various sizes from four source layers. For each source layer, a rough binary classification is first performed to filter out some of the negative examples, and this process is able to mitigate the adverse effects of the imbalance. Next, we improve the quality of the feature extraction through deconvolution and feature fusion. Finally, we conduct fine detection to locate the position and regress the class.

We reconstructed two open-access datasets of SAR images after dirty cleaning and a two-level classification. Then, RDISD-SAR was carefully made according to the image diversity, and the quality, training, test, and validation subsets were scientifically set. Experiments on RDISD-SAR and SSDD show that the proposed approach reaches state-of-the-art accuracy and has a competitive detection speed compared with related detectors. Moreover, the APs of our method on RDISD-SAR and SSDD are 88.70% and 90.57%, respectively, which are higher than the corresponding figures of DSOD by 10.43% and 4.23%, respectively, and are 6.64% and 1.70% higher than the APs of ScratchDet. The detection speed of our detector is 58.2 FPS. Moreover, in terms of the number of parameters and the quantity of computations, our detector also shows some superiorities. Numerous ship detection results show that our method is able to accurately locate ships of various sizes in many scenarios and that the robustness of the proposed approach is superb. In addition, the produced RDISD-SAR shows obvious advantages in evaluating various detectors more scientifically than SSDD.

VII. FUTURE WORK
SAR images in the real-world applications are much diverse, although these transfer-learning-based detectors, train-from-scratch detectors, and the proposed detector have performed fairly well in our experiments, the better performance is our relentless pursuit, so we are going to study how to improve the robustness of these algorithms, domain adaptation is a promising solution to achieve this for it can help the detector learn more advanced common features belonging to an object. So, this will be the focus of our next research.

ACKNOWLEDGMENT
The authors would like to express our sincere thanks to Dr. Jianwei Li and Dr. Yuanyuan Wang for their open-access datasets, and they are very grateful to these detectors’ designers for opening their source code. Their RDISD-SAR is open to anyone who is interested in SAR ship detection. The readers are welcome to contact them.

REFERENCES
[1] H. Dai, L. Du, Y. Wang, and Z. Wang, “A modified CFAR algorithm based on object proposals for ship target detection in SAR images,” IEEE Geosci. Remote Sens. Lett., vol. 13, no. 12, pp. 1925–1929, Dec. 2016.
[2] X. Wang and C. Chen, “Ship detection for complex background SAR images based on a multiscale variance weighted image entropy method,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 2, pp. 184–187, Feb. 2017.
[3] C. Wang, F. Bi, W. Zhang, and L. Chen, “An intensity-space domain CFAR method for ship detection in HR SAR images,” IEEE Geosci. Remote Sens. Lett., vol. 14, no. 4, pp. 529–533, Apr. 2017.
[4] T. Li, Z. Liu, R. Xie, and L. Ran, “An improved superpixel-level CFAR detection method for ship targets in high-resolution SAR images,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 1, pp. 184–194, Jan. 2018.
[5] S. Wang, M. Wang, S. Yang, and L. Jiao, “New hierarchical saliency filtering for fast ship detection in high-resolution SAR images,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 1, pp. 351–362, Jan. 2017.
[6] G. Gao and G. Shi, “CFAR ship detection in nonhomogeneous sea clutter using polarimetric SAR data based on the notch filter,” IEEE Trans. Geosci. Remote Sens., vol. 55, no. 8, pp. 4811–4824, Aug. 2017.
[7] X. Yuan, T. Tang, D. Xiang, Y. Li, and Y. Su, “‘Target recognition in SAR imagery based on local gradient ratio pattern,’” Int. J. Remote Sens., vol. 35, no. 3, pp. 857–870, Feb. 2014.
[8] D. Xiang, T. Tang, Y. Ban, and Y. Su, “Man-made target detection from polarimetric SAR data via nonstationarity and asymmetry,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 9, no. 4, pp. 1459–1469, Apr. 2016.
[9] T. Tang, D. Xiang, and H. Xie, “Multiscale salient region detection and salient map generation for synthetic aperture radar image,” J. Appl. Remote Sens., vol. 8, no. 1, Dec. 2014, Art. no. 083501.
[10] D. Xiang, T. Tang, W. Ni, H. Zhang, and W. Lei, “Saliency map generation for SAR images with Bayes theory and heterogeneous clutter model,” Remote Sens., vol. 9, no. 12, pp. 1290, Dec. 2017.
[11] L. Zhang, L. Zhang, and B. Du, “Deep learning for remote sensing data: A technical tutorial on the state of the art,” IEEE Geosci. Remote Sens. Mag., vol. 4, no. 2, pp. 22–40, Jun. 2016.
[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142–158, Jan. 2016.
[13] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Apr. 2015, pp. 1440–1448.
[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[15] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2016, pp. 379–387.
[16] 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. (CVPR), Jun. 2017, pp. 6517–6525.
[17] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767
[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934. [Online]. Available: http://arxiv.org/abs/2004.10934
[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in Proc. Eur. Conf. Comput. Vis. Amsterdam, The Netherlands: Springer, 2016, pp. 21–37.
[20] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, “DSOD: Learning densely supervised object detectors from scratch,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Aug. 2017, pp. 1–10.
[21] 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. (BIGSARDATA), Nov. 2017, pp. 1–6.
[22] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, “A SAR dataset of ship detection for deep learning under complex backgrounds,” Remote Sens., vol. 11, no. 7, p. 765, Mar. 2019.
[23] Z. Cui, X. Wang, N. Liu, Z. Cao, and J. Yang, “Ship detection in large-scale SAR images via spatial shuffle-group enhance attention,” IEEE Trans. Geosci. Remote Sens., early access, Jun. 2, 2020, doi: 10.1109/TGRS.2020.2997200.
[24] S. Xian, W. Zhtru, S. Yuanruin, D. Wenhuin, Z. Yue, and F. Kun, “AIR-SARShip-1.0: High-resolution SAR ship detection dataset,” J. Radars, vol. 8, no. 6, pp. 852–862, 2019.
[25] M. Ma, J. Chen, W. Liu, and W. Yang, “Ship classification and detection based on CNN using GF-3 SAR images,” Remote Sens., vol. 10, no. 12, p. 2043, Dec. 2018.
[26] J. Wang, T. Zheng, P. Lei, and X. Bai, “A hierarchical convolution neural network (CNN)-based ship target detection method in spaceborne SAR imagery,” Remote Sens., vol. 11, no. 6, p. 620, Mar. 2019.
[28] F. Gao, W. Shi, J. Wang, E. Yang, and H. Zhou, “Enhanced feature extraction for ship detection from multi-resolution and multi-scene synthetic aperture radar (SAR) images,” Remote Sens., vol. 11, no. 22, p. 2694, Nov. 2019.

[29] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2018, pp. 6154–6162.

[30] M. Kang, K. Ji, X. Leng, and Z. Lin, “Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection,” Remote Sens., vol. 9, no. 8, p. 860, Aug. 2017.

[31] Y. Li, Z. Ding, C. Zhang, Y. Wang, and J. Chen, “SAR ship detection based on resnet and transfer learning,” in Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Jul. 2019, pp. 1188–1191.

[32] 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.

[33] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 936–944.

[34] J. Jiao, Y. Zhang, H. Sun, X. Yang, X. Gao, W. Hong, K. Fu, and X. Sun, “A densely connected End-to-End neural network for multiscale and multiscale SAR ship detection,” IEEE Access, vol. 6, pp. 20881–20892, 2018.

[35] Y. Gui, X. Li, and L. Xue, “A multilayer fusion light-head detector for SAR ship detection,” Sensors, vol. 19, no. 5, p. 1124, Mar. 2019.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020.

[37] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C.-Y. Hsiao, and W.-H. Lee, “Ship detection based on YOLOv2 for SAR imagery,” Remote Sens., vol. 11, no. 7, p. 786, Aug. 2019.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861, [Online]. Available: http://arxiv.org/abs/1704.04861

[39] T. Zhang and X. Zhang, “High-speed ship detection in SAR images based on a grid convolutional neural network,” Remote Sens., vol. 11, no. 10, p. 1206, May 2019.

[40] L. Yang, J. Su, and X. Li, “Ship detection in SAR images based on deep convolutional neural network,” Syst. Eng. Electron., vol. 41, no. 9, pp. 1990–1997, 2019.

[41] C. Chen, C. He, C. Hu, H. Pei, and L. Jiao, “A deep neural network based on an attention mechanism for SAR ship detection in multiscale and complex scenes scenarios,” IEEE Access, vol. 7, pp. 104884–104863, 2019.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[43] 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. (CVPR), Jun. 2016, pp. 2816–2824.

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, inception-ResNet and the impact of residual connections on learning,” in Proc. 31st AAAI Conf. Artif. Intell., Feb. 2017, pp. 4278–4284.

[45] T. Zhang, X. Zhang, J. Shi, and S. Wei, “HyperLi-net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery,” ISPRS J. Photogramm. Remote Sens., vol. 167, pp. 123–153, Sep. 2020.

[46] H. Y. Cui, T. Kato, K. Tsuji, T. Gawk, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 658–666.

[47] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, “Soft-NMS—Improving object detection with one line of code,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 5562–5570.

[48] J. Sun, L. Yang, H. Huang, and G. Jin, “Improved SSD algorithm for small-scale SAR ship detection,” Syst. Eng. Electron., vol. 42, no. 5, pp. 1026–1034, May 2020.

[49] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei, “Automatic ship detection based on RetinaNet using multi-resolution Goafon-3 imagery,” Remote Sens., vol. 11, no. 5, p. 531, Mar. 2019.

[50] T. Zhang, X. Zhang, J. Shi, and S. Wei, “Depthwise separable convolution neural network for high-speed SAR ship detection,” Remote Sens., vol. 11, no. 21, p. 2483, Oct. 2019.

[51] X. Zhang, T. Zhang, J. Shi, and S. Wei, “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.

[52] T. Zhang and X. Zhang, “ShipDeNet-20: An only 20 convolution layers and <1-MB lightweight SAR ship detection,” IEEE Geosci. Remote Sens. Lett., early access, May 29, 2020, doi: 10.1109/LGRS.2020.2993899.

[53] Y. Miao, 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.

[54] X. Zhang, H. Wang, C. Xu, Y. Lv, C. Fu, H. Xiao, and Y. He, “A lightweight feature optimizing network for ship detection in SAR image,” IEEE Access, vol. 7, pp. 141662–141678, 2019.

[55] Z. Deng, H. Sun, S. Zhou, and J. Zhao, “Learning deep ship detector in SAR images from scratch,” IEEE Trans. Geosci. Remote Sens., vol. 57, no. 6, pp. 4021–4039, Jun. 2019.

[56] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. 3rd Int. Conf. Learn. Represent. (ICLR), May 2015, pp. 1–14.

[57] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, “Network trimming: A data-driven neuron pruning approach towards efficient deep architectures,” 2016, arXiv:1607.03250. [Online]. Available: http://arxiv.org/abs/1607.03250

[58] R. Zhu, S. Zhang, X. Wang, L. Wen, H. Shi, L. Bo, and T. Mei, “ScratchDet: Training single-shot object detectors from scratch,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 2263–2272.

[59] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in Proc. ACM Int. Conf. Multimedia, 2014, pp. 675–678.

[60] S. Liu, D. Huang, and Y. Wang, “Receptive field block net for accurate and fast object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), Sep. 2018, pp. 385–400.

[61] Q. Zhang, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, “M2Det: A single-shot object detector based on multi-level feature pyramid network,” in Proc. 33rd AAAI Conf. Artif. Intell., Honolulu, HI, USA, Jan. 2019, pp. 9259–9266.
WEIDONG YANG received the B.S., M.S., and Ph.D. degrees from Space Engineering University, Beijing, China, in 2005, 2008, and 2019, respectively.
He is currently an Engineer with Jiuquan Satellite Launch Center. His research interests include radar target detection, tracking, and radar signal processing.

XU WU received the B.S. degree from Northwestern Polytechnical University, in 2015, and the M.S. degree from Space Engineering University, in 2018.
He is currently an Assistant Engineer with the Unit 32359 of PLA. His research interests include SAR image processing and interpretation.