Hindawi
Journal of Sensors
Volume 2021, Article ID 8893182, 19 pages
https://doi.org/10.1155/2021/8893182

Research Article

End-to-End Ship Detection in SAR Images for Complex Scenes Based on Deep CNNs

Yao Chen,[1,2] **Tao Duan**,[1] **Changyuan Wang**,[1] **Yuanyuan Zhang**,[1] **and Mo Huang**[1,2]

[1]*Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China*
[2]*University of Chinese Academy of Sciences, Beijing 100029, China*

Correspondence should be addressed to Mo Huang; huangmo@ime.ac.cn

Received 11 September 2020; Revised 23 December 2020; Accepted 12 February 2021; Published 23 March 2021

Academic Editor: Jaime Lloret

Ship detection on synthetic aperture radar (SAR) imagery has many valuable applications for both civil and military fields and has received extraordinary attention in recent years. The traditional detection methods are insensitive to multiscale ships and usually time-consuming, results in low detection accuracy and limitation for real-time processing. To balance the accuracy and speed, an end-to-end ship detection method for complex inshore and offshore scenes based on deep convolutional neural networks (CNNs) is proposed in this paper. First, the SAR images are divided into different grids, and the anchor boxes are predefined based on the responsible grids for dense ship prediction. Then, Darknet-53 with residual units is adopted as a backbone to extract features, and a top-down pyramid structure is added for multiscale feature fusion with concatenation. By this means, abundant hierarchical features containing both spatial and semantic information are extracted. Meanwhile, the strategies such as soft non-maximum suppression (Soft-NMS), mix-up and mosaic data augmentation, multiscale training, and hybrid optimization are used for performance enhancement. Besides, the model is trained from scratch to avoid learning objective bias of pretraining. The proposed one-stage method adopts end-to-end inference by a single network, so the detection speed can be guaranteed due to the concise paradigm. Extensive experiments are performed on the public SAR ship detection dataset (SSDD), and the results show that the method can detect both inshore and offshore ships with higher accuracy than other mainstream methods, yielding the accuracy with an average of 95.52%, and the detection speed is quite fast with about 72 frames per second (FPS). The actual Sentinel-1 and Gaofen-3 data are utilized for verification, and the detection results also show the effectiveness and robustness of the method.

1. Introduction

Ship detection has received unprecedented attention due to the significant application value such as fishery management, ship rescue, maritime traffic control, and battlefield awareness [1]. Synthetic aperture radar (SAR) is an advanced active microwave sensor with the characteristics of high resolution and wide swath [2]. Due to the prominent capacity of all-day and all-weather imaging, it has been extensively used for marine surveillance such as ship detection [3]. However, there still remain some difficulties to resolve for SAR ship detection, such as multiscale detection and complex scenes interference, especially for dense inshore ships. Moreover, it is quite difficult to satisfy the requirements for both high-speed and high-accuracy detection in SAR images. Therefore, this paper focuses on an end-to-end detection method based on deep learning to address these problems.

SAR images are different from optical images, and the microwave imaging mechanism is more complicated. The artificial recognition of SAR image objects is difficult and requires strong domain knowledge, hence, the automatic target recognition (ATR) system of SAR images is necessary [4]. SAR ocean images are heterogeneous and contain the ships, upwelling, breaking waves, and a lot of artifacts such as radio frequency interferences (RFIs) and azimuth ambiguities [5]. The texture and grey information between the ships and the false alarms are indistinguishable. Moreover, the background of the SAR inshore image is more complex due to the impact of multiple interferences from the lands, which also greatly increases the difficulty of detection [3]. A lot of

researches have been devoted to ship detection in SAR images for decades. Traditional methods could be roughly divided into three categories including threshold methods [6], statistics methods [7], and transformation methods [8], they generally use a priori knowledge to extract features manually through a series of candidate regions. Among these traditional methods, the constant false alarm rate (CFAR) and its improved algorithms based on threshold are frequently-used in the SAR field [9]. As for CFAR detectors, an appropriate statistical model is selected to match the probability density function (PDF) of the ocean clutter, and then an adaptive threshold is calculated with a typical probability of false alarm (PFA). The key to CFAR ship detection is the statistical model of the ocean clutter. In general, the most commonly used models in real applications are mainly based on Gauss [10], K [11], Rayleigh [12], and Weibull [13] distributions. The most popular CFAR methods are the cell-averaging CFAR [14], the two-parameter CFAR [15], and the order-statistic CFAR [16], etc. However, the lands, islands, or some artifacts such as RFIs and azimuth ambiguities will affect the final performance of CFAR detectors. These interferences will lead to a mass of false alarms even if the statistical model accurately matches the real sea states [17]. Besides, CFAR detectors are not sensitive to densely arranged ships and the related parameters acquired from the contaminated background will be wrongly estimated and this could lead to severe missing detection [18]. Hence, the application scenarios of CFAR detectors are limited, and the migration capacity is relatively weak.

Deep convolutional neural networks (CNNs) have obtained enormous achievements in object detection by their strong automatic feature extraction and representation ability [19]. A deep-learning-based detector is generally data-driven and requires little manual intervention, efficiently and simply [20]. The mainstream CNN-based object detectors can be divided into two types: one-stage methods and two-stage methods. Two-stage methods divide the detection into two steps including positioning and classification. Region CNN (R-CNN) introduces deep learning methods to the field of object detection and outperforms most of the traditional detection methods [21]. Faster R-CNN [22], Mask R-CNN [23], and R-FCN [24] are the typical two-stage detection algorithms; they have higher accuracy but are relatively time-consuming and tedious. One-stage methods achieve detection tasks directly with category classification and location regression simultaneously. You only look once (YOLOv1) as the first end-to-end algorithm for object detection processes the full input image only once, and this reduces the computational redundancy and improves the detection speed [25]. Single Shot Detector (SSD) [26], RetinaNet [27], YOLOv3 [28], and the latest YOLOv4 [29] transform the classification problem to the regression problem and are the typical one-stage detection algorithms. Due to the end-to-end characteristic, the one-stage methods are relatively fast and easy to train, which means real-time processing and more suitable for mobile deployment.

With the rapid development of SAR sensors in the past few years, the volumes of SAR image data are getting larger and the data are easier to obtain. This leads to the possibility of deep learning methods for SAR ship detection. Nowadays, more and more scholars have introduced deep learning methods into SAR ship detection. Some methods focus on improving the detection accuracy, and most of them use a two-stage detection framework, such as the Faster R-CNN [30, 31]. Based on the original Faster R-CNN, they have made some typical improvements such as adding hard negative mining [30] and dense connection [31]. There are also some methods dedicated to building a more complex network structure to improve the robustness for some tough problems such as dense small ships [32–34]. Zhao et al. propose a cascade coupled convolutional network with attention mechanism to detect SAR ships and show promising results for small ships [32]. A novel dense pyramid network with attention weighting is invented and solves the problem of multiscale ship detection [33]. Mao et al. come up with an efficient ship detector for SAR images based on simplified U-Net which is more accurate with acceptable speed [34]. These novel network structures can achieve higher final detection accuracy but increase the computational complexity to some extent. At the same time, some methods are concerned about more precise location and use the rotatable bounding boxes to detect which could locate the ships more finely [35, 36]. Besides, some novel training techniques such as training from scratch are also adopted in the SAR field, and the final results also outperform other pretrained ship detectors [37]. Recently, more and more methods pay attention to high-speed processing of ship detection, and most of them are based on a one-stage detection framework such as the YOLO series [38–40]. Chang et al. [38] adopt the YOLOv2 for ship detection in SAR images firstly and reduce the inference time to some extent. Zhang et al. [39, 40] put forward the novel methods based on grid CNN and depthwise separable CNN, and the study demonstrates the possibility of real-time ship detection. The above works indicate that the deep learning methods have achieved superior performance in SAR ship detection and have shown great potential. As far as we know, most of the researches either focus on high-accuracy detection or high-speed detection, and only few researches focus on both. However, both of the two indicators are extremely important for ship detection.

According to the characteristics of SAR ocean image and ship distribution, considering the needs of high-accuracy and high-speed, an end-to-end multiscale ship detection method is proposed in our work. The proposed network is mainly composed of a backbone part and a detection part based on the one-stage framework. First, the input images are divided into different grid cells, and each grid cell can predict multiple objects based on the predefined anchor boxes for dense prediction. The optimal sizes of multiscale anchors obtained by K-means clustering are more suitable for SAR ship detection. Next, the whole image is input into DarkNet-53, which is deeper with residual units, to extract both shallow location and deep semantic features. Finally, the detection is completed by the detection network which has a top-down pyramid structure for three different scales with concatenation operations, and the soft non-maximum suppression (Soft-NMS) which is more conducive for dense ships is combined at last. We also introduced the enhancement strategies without increasing network complexity such as mix-up and mosaic data argumentation, multiscale training, and

hybrid optimization. The whole pipeline of the proposed method is a single data-driven network without manual adjustment, which is efficient in computation, strong to detect multiscale objects, and robust to the complex scenes. Moreover, we try to train our one-stage ship detection model from scratch for the first time. These can also provide relevant suggestions for the subsequent improvement of other SAR ship detection algorithms. The experiment results on the open SSDD data set and the Sentinel-1 and Gaofen-3 images demonstrate that the proposed method achieves state-of-the-art performance in both accuracy and speed and outperforms the previous leading algorithms for SAR ship detection. Besides, different from offshore ships, the detection of inshore ships is often more difficult, and it is also a pain point of SAR ship detection [41]. We have specially built relevant inshore and offshore data sets to conduct more accurate verification of detection in different scenes. The results also demonstrate the excellent performance of our method, especially for inshore ships.

The main contributions of our work in this paper are as follows:

(1) An end-to-end multiscale ship detector is established based on deep CNNs which meets the requirements for both high-accuracy and high-speed detection

(2) The effect of each enhancement strategy such as K-means anchors, data augmentation, multiscale training, and hybrid optimization is evaluated and analyzed through a series of ablation experiments

(3) The SAR offshore and inshore ship data sets are established based on the improvements of the original SSDD data set to evaluate the method more precisely

(4) The proposed one-stage ship detection method is trained from scratch without pretraining, and this confirms the effectiveness of training from scratch in the SAR field

(5) The proposed method outperforms other ship detection methods and has high detection accuracy for both inshore and offshore ships based on real-time processing

The organization of this paper is as follows. Section 2 describes the methodology and improvements. In Section 3, a series of experiments and ablation studies are performed to validate the effectiveness and robustness of the method, and the verification of actual scenarios for Sentinel-1 and Gaofen-3 is shown. Finally, the conclusions of this paper are given in Section 4.

2. Methodology

The proposed method will be described in detail in this section. First, the overall architecture and basic principle are introduced. Afterward, the mechanism of every key block will be explained. Besides, the characteristics of the end-to-end training and inference will be emphasized. Other strategies and details such as K-means clustering for the anchor box, batch normalization, Soft-NMS, and training from scratch of the method will be described at last.

2.1. Overall Framework. The object detection methods based on deep learning has shown great potential in remote sensing in recent years [42, 43]. Figure 1 illustrates the end-to-end framework of the proposed method, and one of the most significant features is the grid-based detection. The whole detection pipeline of the method is a single network, so it can be optimized end-to-end directly on the performance. Compared with the other two-stage detector which is divided into region proposal and class prediction [43], the region proposal and class prediction are integrated into one single net. In Figure 1, the feature classification operation as Figure 1(e) and bounding box regression operation as Figure 1(f) are performed simultaneously. This simplifies the network structure and reduces computational redundancy. Hence, the detection speed can get dramatically improved. This ensures the balance of detection accuracy and speed.

Figure 2 is the processing flow of the proposed method. First, to make the input SAR images of different sizes keep the same feature dimension, all the original images with different sizes need to be resized by resampling. Then, the SAR image is divided into $S \times S$ grid of cells. Also, S is one of the hyperparameters of the network that can be dynamically adjusted. For SAR ship detection, we set $S = 9$ as an appropriate tradeoff between accuracy and speed. Next, if the position of a target falls on a particular grid, this grid is used for detection. Each grid cell in the image can predict multiple objects based on the mechanism of the anchor box. The number of anchor boxes is a predefining parameter, and we set 9 different anchor boxes for three different scales. Every grid cell can predict the 3 bounding boxes based on the preset anchor boxes on each scale. This increases the detection capability for dense ships significantly, especially for the inshore scenes.

There are 5 parameters for every bounding box, including four position coordinates and one confidence score. The confidence score of the bounding boxes is calculated using the logistic regression. Among them, coordinates (x, y) represent the top left corner of the box relative to the responsible grids, and coordinates (w, h) denote the width and the height of the final bounding box. The above parameters are all normalized for the convenience of calculation. The confidence scores represent the possibility of detection results and is defined by:

$$\text{Score} = \Pr(\text{ship}) \times \text{IoU}. \tag{1}$$

If there is a ship in the grid cell, $\Pr(\text{ship}) = 1$, otherwise $\Pr(\text{ship}) = 0$. IoU is the intersection over union, and the expression is

$$\text{IoU} = \frac{\text{area}(P_{\text{box}} \cap G_{\text{box}})}{\text{area}(P_{\text{box}} \cup G_{\text{box}})}, \tag{2}$$

where P_{box} is the bounding box of the object, and G_{box} is the ground truth box of the object. Figure 3 is the schematic diagram of the bounding box in prediction.

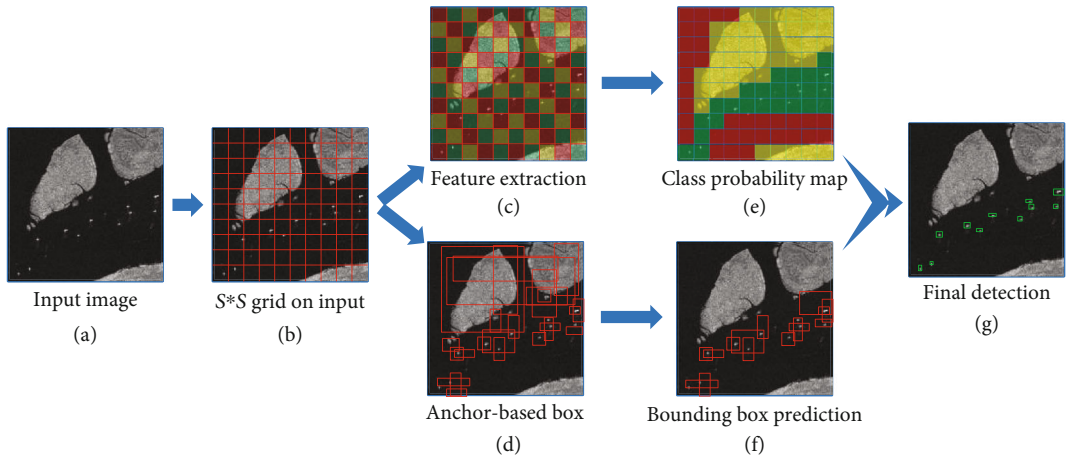


Figure 1: End-to-end framework of the proposed method. (a) Input image after resizing; (b) $S \times S$ grids division; (c) feature extraction; (d) anchor boxes based on the responsible grid; (e) class probability map; (f) bounding box prediction; (g) final detection. All detection procedures are based on one single net.

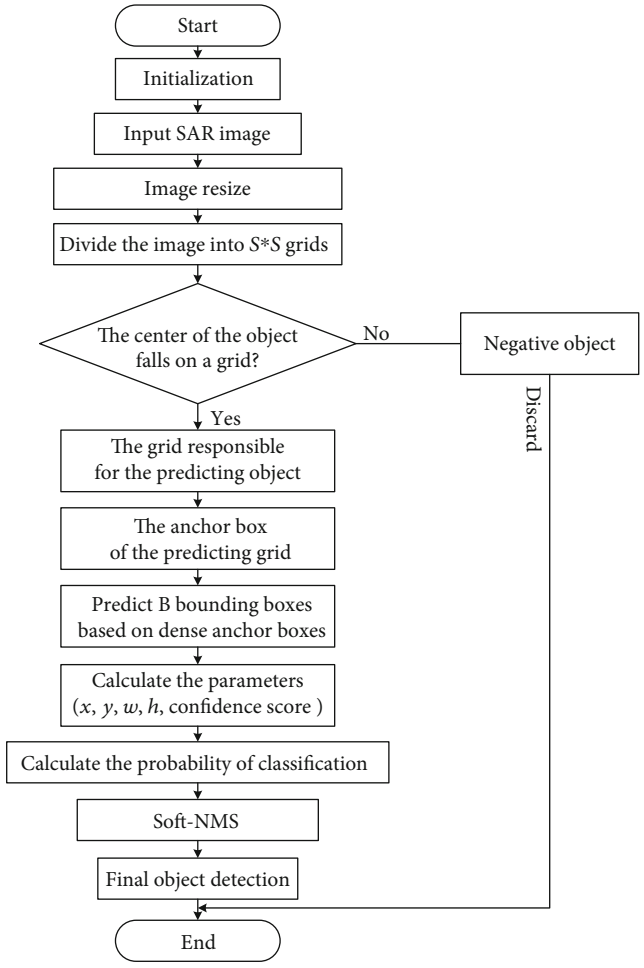


Figure 2: Flow chart of the proposed method. It should be noted that there is a difference between the training processes and the inference processes.

For a specific object, each grid cell has 3 prediction boxes based on the anchor boxes, so there are in total $(9 \times 9 \times 3 \times 5)$ output parameters related to bounding box prediction. The specific calculations of the bounding box corresponding to:

$$b_x = \sigma(t_x) + C_x, \tag{3}$$

$$b_y = \sigma(t_y) + C_y, \tag{4}$$

$$b_w = P_w e^{t_w}, \tag{5}$$

$$b_h = P_h e^{t_h}, \tag{6}$$

where (C_x, C_y) is the center of the grid, $(\sigma(t_x), \sigma(t_y))$ is the coordinate offset of the object, and (P_w, P_h) is the predefined width and height of the anchor box. (t_x, t_y, t_w, t_h) are the network parameters obtained directly through learning, the final bounding box coordinate are (b_x, b_y, b_w, b_h), and all the above parameters are also normalized between 0 and 1.

2.2. Backbone Network. Darknet-53 is improved on the basis of Darknet-19 and has 53 convolutional layers as the name implies [28]. It basically uses a fully convolutional network, replacing the pooling layer with a convolutional stride operation of 2. The whole network does not contain fully connected layers, so the number of parameters is reduced. Under normal circumstances, the number of layers in CNNs is quite relevant to the capability of feature extraction. Darknet-53 adds the effective residual units to avoid gradient dispersion when the network layer is too deep [44]. This improves the network's feature extraction capabilities and also makes the detection results more robust.

The specific structure of Darknet-53 in our model is shown in Figure 4, and it contains six blocks in total as shown in the red dotted box. Darknet-53 contains a lot of successive 3×3 and 1×1 convolutional kernel. Compared to large convolution kernels, smaller convolution kernels can reduce the number of parameters while ensuring the final performance with the same receptive field [20]. The 3×3 convolutional

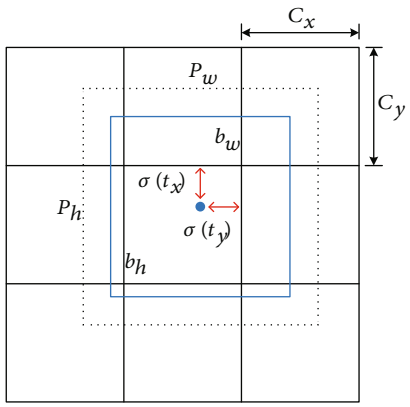


Figure 3: The schematic diagram of the bounding box. Where the black solid boxes are the grids responsible for detection, the black dotted box is the anchor box, and the blue box refers to the bounding box.

layers are used to reduce the size of the input images. The 1×1 convolutional layer is used to compress the features and reduce the dimension. After the operation of convolution layer and residual layer in each block, the feature map will change accordingly, and the smallest feature map on the top layer is compressed to 13×13. The feature maps with different scales contain lots of hierarchical information. If only run the detection on the last layer, this will cause insensitivity for small targets. Therefore, the final detection is performed at 3 different scales after the fusion operations, and this will be explained in detail in the next part.

2.3. Multiscale Detection. Low-level features generally contain the shallow information, middle-level features usually imply the local information, and high-level features normally represent the global information [20]. Pyramid is a valid method to realize multiscale feature fusion and is widely used in the object detection field. Inspired by Feature Pyramid Network (FPN) [45], a top-down pyramid architecture with concatenation operations is adopted to fuse the features in the method. In this way, plentiful features containing both semantic and spatial information are extracted to detect multiscale SAR ships.

There is a total of 106 layers in our methods including feature extraction layers, route layers, detection layers, and other functional layers. Table 1 is the detailed information of multiscale feature maps of the proposed method. There are three scale feature maps in the detection block, which are (13×13), (26×26), and (52×52), respectively, and each scale feature map contains three different size anchor boxes for dense prediction. The detailed structure of the detection network is shown in Figure 5. The feature maps obtained by DarkNet-53 are defined as $\{C0, C1, C2\}$, and the final feature maps after fusion are defined as $\{P0, P1, P2\}$. For feature maps, different receptive fields correspond to different scale targets. The smallest 13×13 feature map with the large receptive field is responsible for large ship detection. The medium 26×26 feature map with the medium receptive field is responsible for medium ship detection. And the largest 52×52 feature map with the small receptive field is responsible for small ship detection. Benefited from those characteristics, the model can preserve powerful features for multiscale object detection, which is very suitable for detecting multiscale ships in SAR images.

2.4. Activation Function and Loss Function. To prevent the phenomenon of gradient vanishing and increase the sparsity of the network, the leaky ReLU [46, 47] is used as the activation function. Due to the ingenious leak mechanism, it can also smooth the network to a certain extent and reach better performance. It can be expressed as:

$$f(x) = \begin{cases} x, x \geq 0, \\ ax, x < 0. \end{cases} \tag{7}$$

The loss function has a huge impact on the final quality of a model. And the loss function of the proposed method integrates coordinate loss, confidence score loss, and classification loss, and the binary cross-entropy loss is used [28]. For an image, the loss function can be abstractly expressed as:

$$\text{Loss}_{\text{total}} = \sum_{i=0}^{S \times S} \text{Coord\_Loss} + \text{Conf\_Loss} + \text{Class\_Loss}. \tag{8}$$

Different types of losses have different effects on the final performance. The specific expressions of coordinate error, confidence score error, and classification error are

$$\begin{aligned} \text{Coord\_Loss} = \lambda_{\text{coord}} \sum_{i=0}^{S \times S} \sum_{j=0}^{B} I_{ij}^{\text{obj}} (2 - w_i^* * h_i^*) [(t_w - t_w^*)^2 + (t_h - t_h^*)^2 \\ + (\sigma(t_x) - \sigma(t_x^*))^2 + \left(\sigma(t_y) - \sigma\left(t_y^*\right)\right)^2], \end{aligned} \tag{9}$$

$$\begin{aligned} \text{Conf\_Loss} = \lambda_{\text{obj}} \sum_{i=0}^{S \times S} \sum_{j=0}^{B} I_{ij}^{\text{obj}} (-1) [c_i^* \log(c_i) + (1 - c_i^*) \log(1 - c_i)] \\ + \lambda_{\text{noobj}} \sum_{i=0}^{S \times S} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} (-1) [c_i^* \log(c_i) + (1 - c_i^*) \log(1 - c_i)], \end{aligned} \tag{10}$$

$$\text{Class\_Loss} = \lambda_{cls} \sum_{i=0}^{S \times S} \sum_{j=0}^{B} I_{ij}^{\text{obj}} (-1) [p_i^* \log(p_i) + (1 - p_i^*) \log(1 - p_i)], \tag{11}$$

where letters marked with * represent the true values of the ground truth box, and letters without * represent the predicted values of the bounding box. λ_{coord}, λ_{obj}, and λ_{cls} are the weight coefficients for different types of losses. $I_i^{\text{obj}} \in \{0, 1\}$ indicates that whether there exists an object in the grid cell i, and $I_j^{\text{obj}} \in \{0, 1\}$ indicates that the jth bounding box of prediction in grid cell i.

2.5. Other Strategies

2.5.1. K-Means Clustering for Anchors. Anchor box mechanism for object detection was proposed to solve the problem

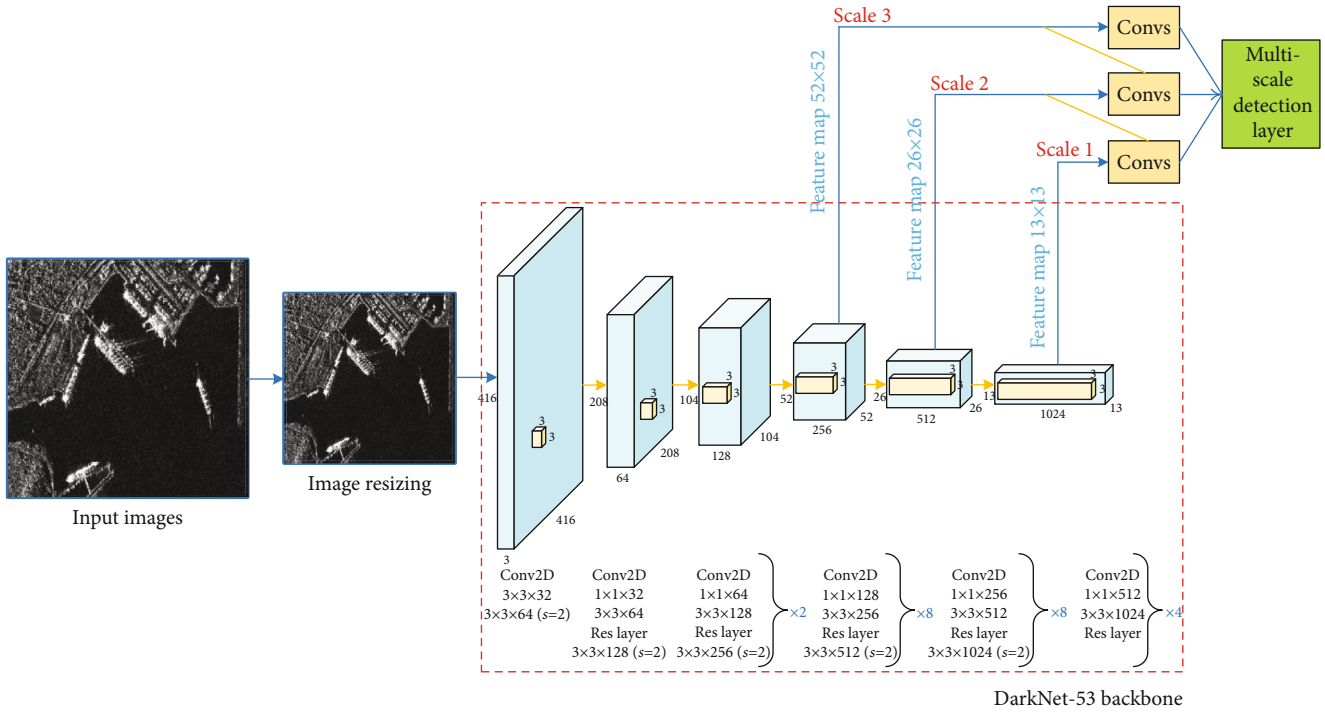


Figure 4: Structure of Darknet-53 backbone. The detailed operations of feature extraction are in the red dotted box and it contains six blocks.

Table 1: Detailed information of multiscale detection.

Feature layer	Size	Anchor box	Number
Feature map-13	13×13	(49,25), (59,118), (108,45)	$13 \times 13 \times 3$
Feature map-26	26×26	(18, 23), (23, 48), (39, 68)	$26 \times 26 \times 3$
Feature map-52	52×52	(8, 12), (11, 22), (13, 30)	$52 \times 52 \times 3$

of multitarget in one predicted box and has been used in many detectors [22]. There are 9 predefined anchor boxes in our method for different scale detection. The k-means clustering is adopted on the overall SSDD data set to automatically find the prior boxes for SAR ship detection. Most ships in SAR images are small and weak targets, which occupy few pixels and have lower contrast. If we use the standard Euclidean distance of the conventional k-means algorithm, the bounding boxes with a larger scale generate more error than the small-scale boxes. This will lead to missed detections of the small and sparse ships, which are very common in SAR images. However, what we want in the final detection are the priors that will lead to high IOU scores. Thus, the distance metric in this study can be expressed as:

$$d(\text{anchor box, cluster centroid}) = 1\text{-IoU}(\text{anchor box, cluster centroid}), \tag{12}$$

where $d(\text{anchor box, cluster centroid})$ is a new distance metric that needs to be minimizing, and $\text{IoU}(\text{anchor box, cluster centroid})$ means the IOU values of different anchor boxes for clustering. The specific size of anchor boxes for three scales are shown in Table 1 and are separately (8, 12), (11, 22), (13, 30), (18, 23), (23, 48), (39, 68), (49, 25), (59, 118), and (108, 45). The optimal cluster centroids obtained by k-means are significantly different than previous hand-picked anchor boxes and have better performance for both precision and recall on SAR ship detection.

2.5.2. Batch Normalization. The deep neural networks are usually hard to train due to the different distribution of each layer. Another strategy of the proposed method is the addition of batch normalization after each convolutional layer, which leads to significant improvements in training convergence [48]. This can reduce the requirement for a low learning rate and speed up the training. Moreover, it also helps regularize the model so the problem of overfitting can be solved, hence, the dropout can be removed to simplify the network. The mathematical expression of batch normalization is

$$Y = \gamma \cdot \frac{X - m(X)}{\sqrt{\sigma^2(X) + \varepsilon}} + \eta, \tag{13}$$

where X is the input vectors, Y is the output vectors, $m()$ means average, γ and η are hyperparameters automatically generated by training, and ε is a constant close to 0 to prevent the situation where the denominator is 0.

2.5.3. Soft Non-maximum Suppression. NMS can be considered as the maximum search in a local area, and it selects the detections with high scores and deletes the close-by neighbors with a lower score. For SAR images, an inshore ship is usually neared closely by other inshore ships in the harbor, and this affects the performance of conventional NMS algorithms. To reduce the redundancy of the final ship

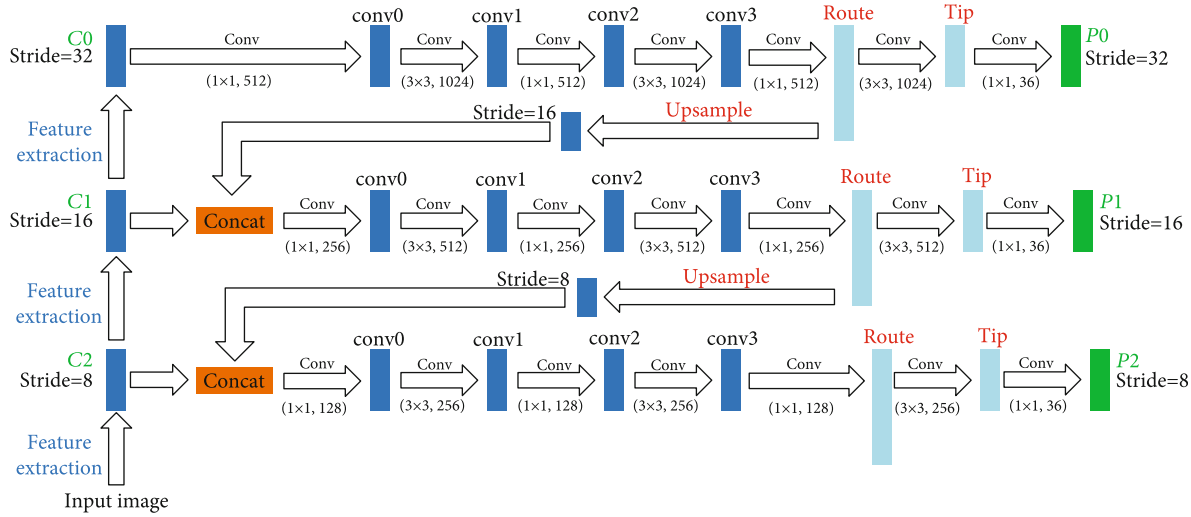


Figure 5: Structure of multiscale detection. The final fused feature maps are $P0$, $P1$, and $P2$ for three different scales and are marked in green. The concatenation operations are marked in orange.

detections, the Soft-NMS [49] with the Gaussian penalty function is adopted in our method based on the classification scores, and this is the last step of detection. It helps to predict final ship detection from a series of redundant candidates and improves the performance of the final detection, especially for inshore scenes. We set the parameter σ of the Gaussian function for Soft-NMS as 0.6, and the Soft-NMS substantially reduces the number of object proposals and will not harm the final detection accuracy.

2.5.4. Train from Scratch. To speed up the convergence, most of the deep learning object detectors are fine-tuned from a pretrained network. Generally, the large-scale optical classification data set called ImageNet is used for pretraining in most cases [50]. However, the loss functions are different between the ImageNet classification tasks and real detection tasks. And the data distributions between classification and detection tasks are absolutely different. These will incur some additional problems to the final detector, such as learning objective bias and insensitivity to locations [51]. Besides, SAR images are single-channel amplitude images, while ImageNet consisted of RGB three-channel images. Usually, the effectiveness of the network will be reduced due to the discrepant image domains. Nowadays, more and more scholars try to train from scratch and have achieved better detection results [37]. To tackle the above critical problems, we try to train our model from scratch for SAR ship detection. Although this will make the training time longer to a certain extent, this is more suitable for detection tasks and achieves better performance in the final detection.

3. Experiments and Results

A series of experiments are conducted to evaluate our method in this section. The open SSDD data set used in our study is introduced first. Besides, we improve the original SSDD data set and establish SSDD-offshore and SSDD-inshore data sets to verify the model's ability to inshore and offshore ships, respectively. The settings of the related experiments and the evaluation criteria are described. The influence of different input image size will be summarized. Then, the experiment results of different data sets including both inshore and offshore ships are given. Moreover, adequate comparisons with other ship detection methods are shown. Finally, the verification on real Sentinel-1 and Gaofen-3 data will be given, and the results also show the superiority of our method.

The experimental platform is a server with Intel(R) Xeon Gold 5118 CPU @2.330 GHz processor and NVIDIA Tesla V100 GPU card with 16 G memory. The programs are based on the PaddlePaddle deep learning framework [52]. The operating system is 64-bit Linux with CUDA v9 and CUDNN v7 computing acceleration.

3.1. Data Set. The SAR ship detection dataset which is called SSDD is established in 2017 to set the baseline of SAR ship detection algorithms [30]. It is publicly released and has been used by many other scholars [31, 33–36, 38–40]. The SSDD data set contains various ships in multitudinous scenes, including different sensors, resolutions, polarizations, scenes, and work modes. It uses the same labeling method as the popular PASCAL VOC data set [53], and the detailed descriptions of SSDD are shown in Table 2.

In the past, the SSDD data set was often randomly divided into the train, validation, and test data sets. However, this made the test environment of the past work different due to the random test sets. Nowadays, more and more scholars have noticed this problem [39, 40]. It is not conducive to testing the performance of the ship detection algorithm under the same baseline. To use SSDD more standardized, we have established the following rules to regulate the use of SSDD with the consent of the creators of SSDD [30]. In this experiment, SSDD is divided into two parts: 80% for the training and 20% for the testing, where the number of images ending in 1 and 9 are the test set, and the rest are the training set. Such a division considers the balance between offshore and inshore ships better. This can provide

Table 2: Detailed descriptions of SSDD.

Payload	Wave bands	Scenes	Resolution	Polarization	Images	Ships
RardarSat-2	C band					
TerraSAR-X	X band	Inshore and offshore	1 m–15 m	HH, VV and HV, VH	1160	2540
Sentinel-1	C band					

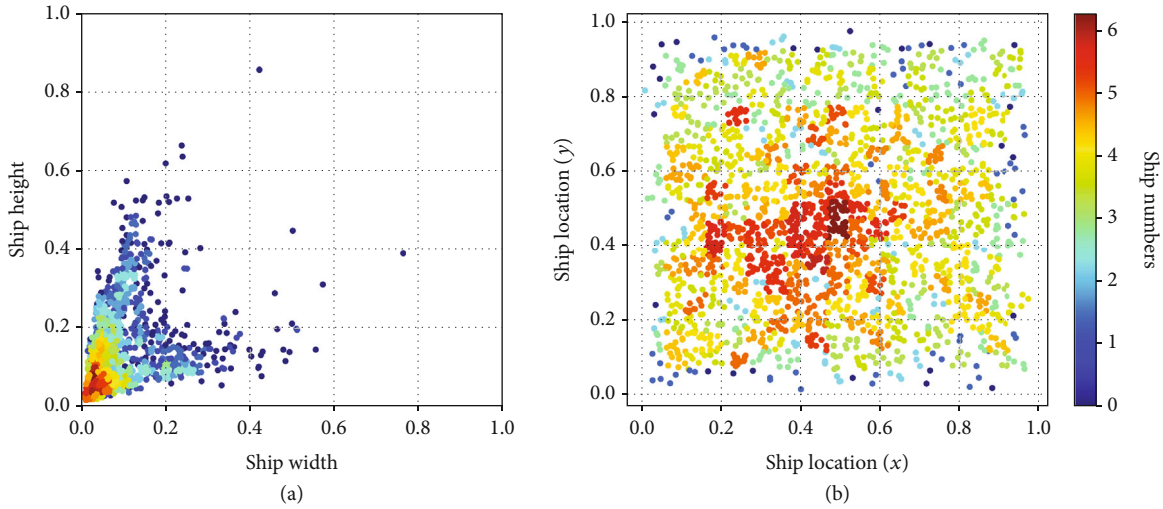


Figure 6: Visualized distribution of multiscale ships in SSDD. (a) Ship size distribution and (b) ship location distribution. The coordinate values are all normalized between 0 and 1. The color change represents the change in the number of ships.

suggestions for subsequent scholars to use the SSDD normatively. In this way, the performance of each algorithm can be evaluated more objectively.

Besides, the statistical analysis is performed on the SSDD data set including the ship distributions of sizes (height and width) and locations (x-axis direction and y-axis direction). The visualized analysis results are shown in Figure 6. The ships have large scale changes, the smallest ship in SSDD is about 4×7 pixels, and the largest ship is about 211×298 pixels. These increase the diversity of the SSDD data set, so it is widely used in SAR ship detection. At the same time, it can be observed in Figure 6(b) that ships are distributed throughout the image, but most of them are concentrated in the center of the image.

The detailed statistical results of different scale ships in the SSDD data set are also given in Table 3. There are 1529 small-scale ships (area < 322 pixels), 935 medium-scale ships (322 < area < 642 pixels), and 76 large-scale ships (area > 642 pixels) in the SSDD. It can be seen that the SSDD data set consists mainly of small-scale targets.

Under normal circumstances, the detection of inshore ships is harder due to land interference. In most inshore cases, the tiny ships are likely to be neglected, and the overlapping ships are hard to classify. To verify the robustness of the method for both offshore and inshore scenes, the SSDD-offshore and SSDD-inshore data sets are established. Based on the original SSDD data set, we carried out a detailed classification according to whether the scenes of each image is inshore or offshore. Through statistics, there are separately 716 inshore ships and 1824 offshore ships in two data sets, and the detailed descriptions are shown in Table 4. For single target detection of ships, the number of the data sets is sufficient to validate our method.

3.2. Implementation Details. In this part, the implementation details of the proposed method such as data preprocessing, training strategies, and optimization tricks will be described, so that the detection results in this study can be reproduced.

3.2.1. Data Augmentation and Normalization. The basic data augmentation methods such as random distortion, random flip, random expansion, and random crop are adopted. At the same time, we also introduce the latest mix-up [54] and mosaic data [29] augmentation techniques. These methods not only increase the original data set but also make the network more robust. Moreover, as the amount of train data increases, the overfitting can be effectively avoided. The schematic diagram of mix-up and mosaic data augmentation is shown in Figure 7. The ground truth boxes also need to be recalculated after the random augmentation. During the training processes, the order of the ground truth boxes in the image has been randomized, and this can reduce the learning of biased features. Besides, the values of the image pixels are also normalized between 0 and 1 to speed up the convergence of the network.

3.2.2. Multiscale Training. To characterize the multiscale features of ships flexibly, we introduce a multiscale training strategy. As introduced in Section 2, the proposed network is all consisted of convolutional layers and does not have fully

Table 3: Statistical results of multiscale ships in SSDD.

Data set	Size of ships (num)			Size of images (pixels)	
	Small	Medium	Large	Height	Width
SSDD	1529	935	76	190~526	214~668

Table 4: Detailed descriptions of SSDD-offshore and SSDD-inshore.

Data sets	Scenes	Images	Ships
SSDD-offshore	Offshore only	956	1824
SSDD-inshore	Inshore only	204	716
SSDD	Off-inshore both	1160	2540

connected layers. As a fully convolutional structure, this makes no restriction on the input size of images. Hence, the multiscale training of different size input becomes possible. During the training processes, we randomly resize the size of input images as the multiples of 32 between 320 and 640 per 10 iterations. This makes our detector more robust to the input size of different resolutions across a variety of scales and enhances the generalization ability of the final model. More importantly, this does not increase the computational complexity of the final network.

3.2.3. Hybrid Optimization. Most optimization algorithms in deep learning are based on gradient descent, and the optimization methods also have a significant influence on the model's final accuracy [55]. In this study, we use a hybrid optimization method that includes the adaptive moment estimation (ADAM) and stochastic gradient descent (SGD) with momentum. These two methods have their own advantages and disadvantages, and they can compensate each other for optimization. The ADAM algorithm is used in the early stages of training for the advantages of efficient calculation and less memory. Then, we fine-tune the loss to convergence by the SGD with momentum algorithm at last [56]. These can optimize the performance of our model more finely and achieve higher accuracy.

3.3. Evaluation Criteria. Some frequently-used evaluation criteria including precision, recall, F1, and mean average precision (mAP) are employed in this study [20]. The definition of precision and recall are as follows:

$$P(\text{Precision}) = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \tag{14}$$

$$R(\text{Recall}) = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \tag{15}$$

where TP and TN indicate the number of true positive and true negative. FP represents the number of false positive, and FN denotes the number of false negative. In general, precision and recall are a pair of mutually influencing values, and they are difficult to evaluate the overall performance. Hence, the $F1$ score and mAP are utilized to evaluate the detector more objectively. Above all, the $F1$ score is expressed as:

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{16}$$

If there is only one category in the data set, such as only ships in SSDD, mAP can be equivalent to average precision (AP), and mAP or AP is defined as follows:

$$\text{mAP} = \int_0^1 P(R)dR. \tag{17}$$

To better test the multiscale detection capability of the proposed method, the Microsoft Common Objects in Context (COCO) evaluation metrics including AP_S, AP_M, and AP_L are also adopted [57]. AP_S is the AP for small ships in which the area is smaller than 32^2, AP_M is the AP for medium ships in which the area is between 32^2 and 96^2, and AP_L is the AP for large ships in which the area is bigger than 96^2. It should be noted that the above-mentioned mAP is the same as AP_{50} in the COCO metrics. The IoU threshold is set as 0.5 for both mAP, AP_S, AP_M, and AP_L in all of our experiments, and the threshold can be adjusted as a balance between missed detections and false alarms according to the practical application.

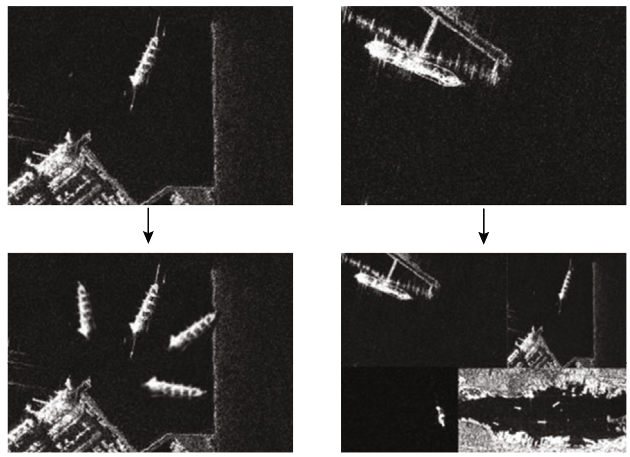
Under normal circumstances, it can be considered that the larger $F1$ or mAP is, the better the detector performs. Frames per second (FPS) indicates that the number of images that can be processed in one second, and we used FPS to represent detection speed. FPS can be computed as:

$$\frac{\text{FPS} = 1s}{\text{Time.}} \tag{18}$$

3.4. The Influence of Different Image Size. Generally, the larger the size of the input image, the greater the amount of information it contains, and the richer the extracted features. This usually improves the detection accuracy of a model. However, when the image size is large, the processing speed will be slow. This is a problem that needs to be balanced, and it is important to choose the appropriate size of the input image for a typical detector. Hence, a set of experiments is carried on to study the influence of the input image size on the proposed detector first.

Due to the fixed structure of our network, the input size must be a multiple of 32. Therefore, we selected a representative set of image sizes based on the multiple of 32, including (320×320, 384×384, 416×416, 448×448, 512×512, 576×576, 608×608, and 640×640). The precision-recall (P-R) curves of the above experiments are given in Figure 8. In general, the area under the curve represents the overall performance of a detector.

For a more detailed comparison, the quantitative values of evaluation indexes of different input sizes are given in Table 5 including the mAP, precision, recall, and $F1$ score. It can be seen that the optimal size for different evaluation index is different. Especially for precision and recall, if one is higher, the other is relatively lower. Therefore, we should take all indexes into consideration. As we can see in Table 5, for the $F1$ score, the size of 608×608 performs best, but the mAP of 608×608 is relatively lower than 416×416. The values of F1 and mAP are relatively close for these two



(a) Mix-up (b) Mosaic

Figure 7: Mix-up (a) and mosaic (b) data augmentation. Mix-up stitches multiple rotating ships into one image and mosaic combines four original images into one image. The ground truth boxes also need to be recalculated.

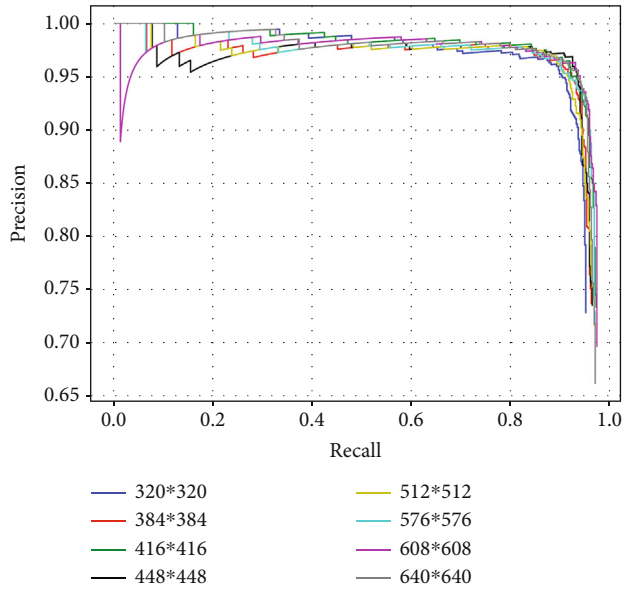


Figure 8: The x-axis denotes the recall and the y-axis denotes the precision. As we can see, precision and recall are a pair of indexes that affect each other.

Table 5: Influence of the different input sizes.

Input size	mAP	Precision	Recall	$F1$
320×320	93.29%	94.74%	91.18%	92.93%
384×384	93.76%	95.35%	92.31%	93.80%
416×416	95.52%	94.54%	94.18%	94.36%
448×448	93.86%	96.86%	92.50%	94.63%
512×512	94.24%	94.95%	91.74%	93.32%
576×576	94.76%	93.52%	94.75%	94.13%
608×608	95.13%	96.31%	93.06%	94.66%
640×640	95.06%	95.41%	93.62%	94.51%

input sizes, this shows the detection capabilities of the model are similar. However, when the size becomes larger such as 608×608, the processing speed also becomes slower. In the case of similar detection capabilities, we should choose a smaller input size for high-speed processing. Moreover, the size of 416×416 is the closest to the original image size of the data set, and the distortion is minimal after resizing the original image. Thus, after both considering the accuracy and speed, we choose the size of 416×416 as the input size in our method.

3.5. Results and Discussions. To validate the performance in different scenes, the test experiments are conducted on SSDD, SSDD-offshore, and SSDD-inshore data sets, respectively. The network is trained for 400 epochs from scratch with batch size = 10 and learning rate = 0.001. Figures 9 and 10 are the typical ship detection results, and we selected some representative images for display including different scales, backgrounds, and ship numbers, respectively. For each sample, the left image indicates the ground truth, and the right is the detection results. The ground truths are marked with green rectangles, and the correct detected ships are marked with blue rectangles. And one rectangle represents a single ship, whether in the detection results or ground truth.

In most cases, offshore ships are relatively easy to detect because of the pure background and fewer disturbances [33]. The test results of typical samples on the SSDD-offshore data set are shown in Figure 9. We picked 8 representative images including a single object and multiple objects under different radar resolution. We can see that there are ships of different scales in the SAR images, Figure 9(d) is the ships of a large scale, and Figure 9(h) is the ships of a small scale. As discussed in Section 2, the feature maps in the top layer that have large receptive fields with more semantic information are conductive to detect the large ships, whereas the feature maps in the down layer that have small receptive fields with more spatial information are fit for small ship detection. A top-down pyramid architecture with concatenation operation is adopted in

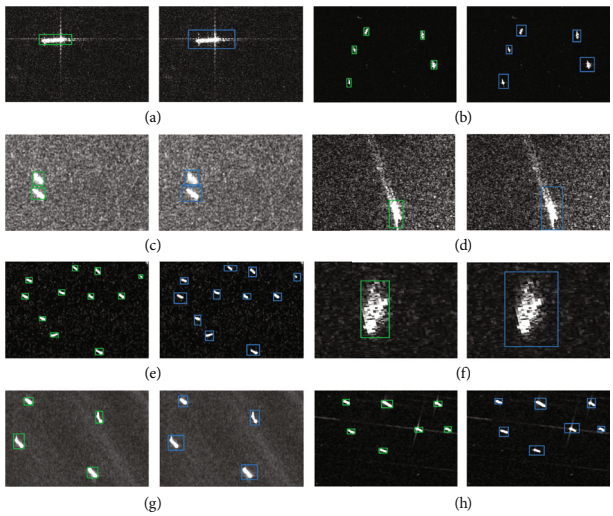


Figure 9: The typical ship detection results for offshore scenes. The green rectangles in (a–h) are ground truths. The blue rectangles in (a–h) indicate detected ships.

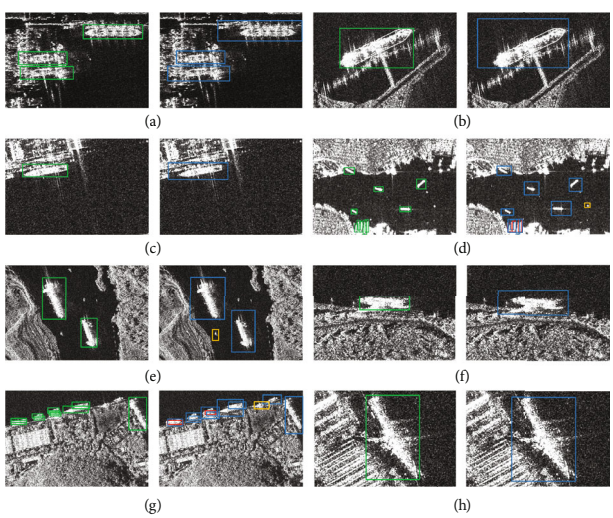


Figure 10: The typical ship detection results for inshore scenes. The green rectangles in (a–h) are ground truths. The blue rectangles in (a–h) indicate detected ships. The orange rectangles in (d), (e), and (g) denote false alarms. The red rectangles in (d) and (g) denote missed detections.

our method, so the detection ability for multiscale ships are stronger with both abundant semantic and location information. Moreover, due to the multiscale training strategy, the detection capabilities for the ships of various scales have been further enhanced without increasing computation complexity. As we can see in Figure 9, all ships are detected correctly across a variety of scales. There were no missed detections or false alarms. It should be noted that Figures 9(c) and 9(d) have strong speckle noise and still have been detected correctly. These results show that our method can achieve better performance in different noise environments with strong antinoise ability.

Due to the interference of heterogeneous ocean background and land clutter, the inshore ships are usually harder to detect and locate [41]. The edge information of docked ships are usually indistinguishable, especially for densely arranged tiny ships. The performance could be easily affected by the connection of ancillary facilities such as bridges, roofs, cables, and other buildings. To evaluate the robustness of our detector under the inshore scenes, we test our method on the SSDD-inshore data set. Figure 10 is the typical detection results for inshore ships. As we can see, Figure 10 contains a variety of inshore ships which includes ships docked at the shore or between the straits. Except for some hard ship samples, the ships across all scales can be detected by our model, and this also shows the powerful multiscale detection capability of our method for complex scenes.

However, there are still some missed detections and false alarms. The orange rectangle appears in Figures 10(d), 10(e), and 10(g), which represents the false alarm objects. These are strong scattering points, which may be unmarked extremely tiny ships or interferers that have high amplitude. This shows that our method still needs improvement. However, compared with missed detections, false alarms can be allowed to some extent because of the lower missing cost. Therefore, the method in this study also has great practicality in a real application. Besides, in Figures 10(d) and 10(g), there exist the missed detections. The missed ships are marked in red rectangles. It can be seen that the missed ships are densely arranged objects, which is inconsistent with the characteristics of most SAR ship targets. These overlapping ships are difficult to distinguish manually even through SAR experts. Most ships are sparse in SAR images and also in the SSDD data set. Therefore, it is difficult for the model to learn the characteristics of densely arranged ships. To better detection of these dense ships, it could be solved by increasing the number of dense ships in the training data set.

Different objects have different length to width ratios, and the prior boxes of the optical objects are not suitable to detect the ships in SAR images. As we can see in the detection results above, most of the objects are well detected in the prediction boxes including multiscale ships in spite of the complexity and diversity of backgrounds. This benefits from the anchors obtained by K-means clustering on SSDD which are more suitable for SAR ship detections. This can also provide a reference for the anchor settings of other SAR ship detection algorithm.

The P-R curves of three different data sets are given in Figure 11. The curves under different scenes are all relatively smooth and show good performance. The detailed ship test results for different data sets are shown in Table 6. As we can see, the mAP of ship detection on SSDD reaches 95.52%, which is reliable in real applications and implies the practicability of the proposed method. Meanwhile, for SSDD-offshore and SSDD-inshore data sets, the mAP can reach 99.50% and 92.80%, respectively. For offshore ships, the method in this study reaches a detection accuracy close to 100%, which is of great significance for detection in vast offshore scenarios. This facilitates global maritime surveillance and has a lot of practical applications such as worldwide military planning and maritime disaster rescue. For inshore ships, the method also reaches a detection accuracy of over 92%, which has been superior to the traditional methods. This will benefit the

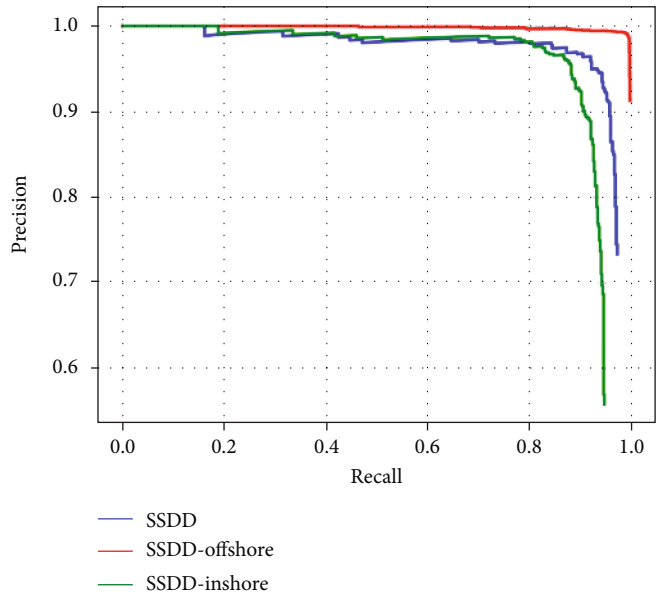


Figure 11: The P-R curves of different data sets.

Table 6: The ship test results of SSDD, SSDD-offshore, and SSDD-inshore (the IoU threshold is set as 0.5).

Data sets	Recall	Precision	mAP (AP_{50})	$F1$	AP_S	AP_M	AP_L
SSDD	94.18%	94.54%	95.52%	94.36%	92.35%	96.26%	97.37%
Offshore	99.23%	99.07%	99.50%	99.15%	97.71%	98.15%	93.02%
Inshore	88.17%	95.49%	92.80%	91.68%	90.26%	94.12%	90.91%

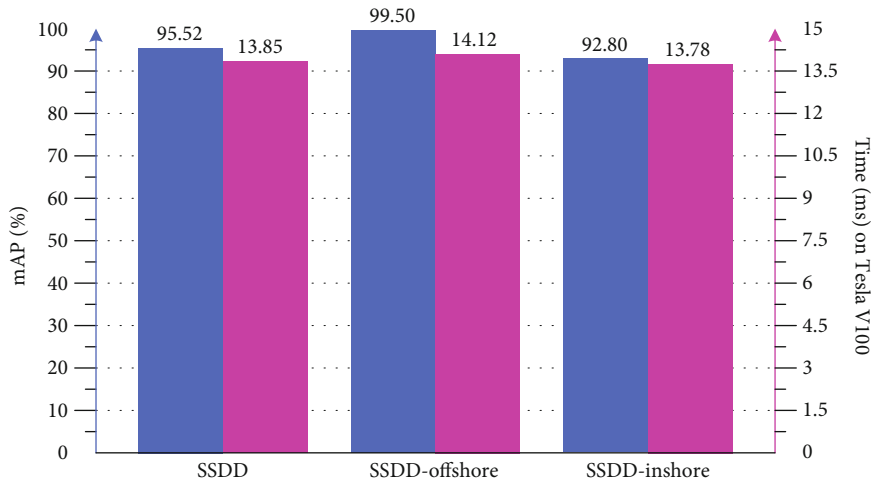


Figure 12: The mAP and detection time of different data sets.

dynamic harbor management, smuggling monitoring, coast safety, and other inshore activities. Also, it can be observed in Table 6 that our method has promising accuracies for both small-scale, medium-scale, and large-scale ships, and the detection accuracy of AP_S, AP_M, and AP_L for both inshore and offshore ships all exceeds 90%. These results indicate that our method has relatively balanced detection capabilities for ships of three different scales.

Due to the efficient paradigm, the advanced unified one-stage methods in this study not only increase the detection accuracy but also speed up the inference time greatly. The test time per image on different data sets is about 14 ms on Tesla V100 GPU, which means the speed is about 72 FPS. This has exceeded the requirements of 30 FPS for real-time processing. The specific mAP (accuracy) and detection time (speed) on different data sets are given in Figure 12. As we

Table 7: Ablation experiments and results.

Improvements	Used?					
Leaky ReLU	√	√	√	√	√	√
Batch normalization	—	√	√	√	√	√
Data augmentation	—	—	√	√	√	√
K-means anchors	—	—	—	√	√	√
Multiscale training	—	—	—	—	√	√
Hybrid optimization	—	—	—	—	—	√
mAP	87.32%	89.03%	91.15%	92.82%	94.43%	*95.52%*

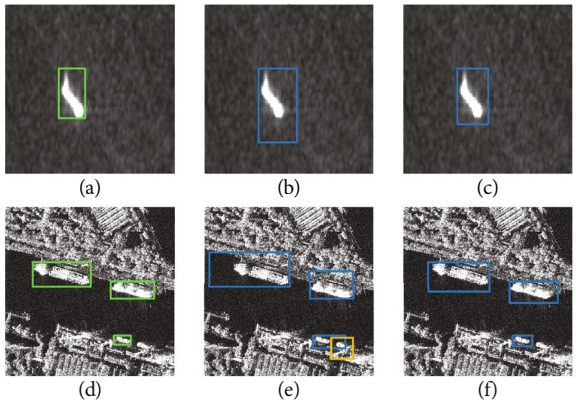


Figure 13: The effect of training from scratch. The second column is the typical detection results by the model with pretraining, the third column is the typical detection results by the model with training from scratch. The green rectangles in (a) and (d) are ground truths. The blue rectangles in (b–e) indicate detected ships. The orange rectangle in (e) denotes a false alarm.

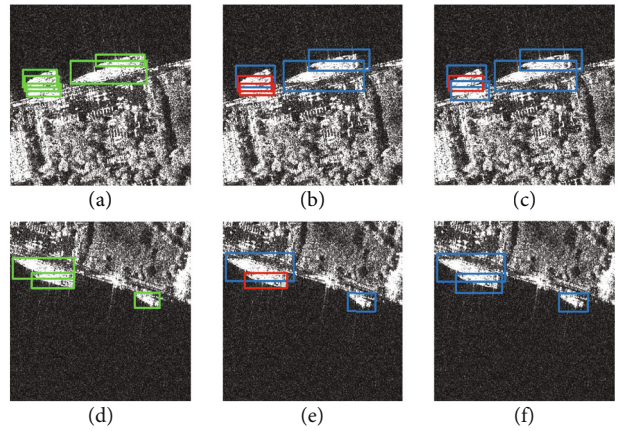


Figure 14: The effect of Soft-NMS. The second column is the typical detection results of the model with conventional NMS, and the third column is the typical detection results of the model with Soft-NMS. The green rectangles in (a) and (d) are ground truths. The blue rectangles in (b–e) indicate detected ships. The red rectangles in (b), (c), and (e) denote missed detections.

Table 8: The comparison of the different training methods.

Training methods	Recall	Precision	mAP	$F1$
Pretraining	92.58%	93.37%	94.11%	93.18%
Training from scratch	94.18%	94.54%	95.52%	94.36%

can see, due to the relatively simple background of the offshore ships, the processing time is relatively faster than inshore ships.

3.6. Ablation Study. To verify the effectiveness of the improvement approaches, multiple sets of ablation experiments on SSDD were performed. The effect of each improvement step of the proposed method such as batch normalization, leaky ReLU, data augmentation, K-means anchors, multiscale training, and hybrid optimization has been analyzed.

As we can see in Table 7, every improvement method leads to a considerable increase in the final mAP. After adding the leaky mechanism to the activation function, the sparsity of the model has increased, and the final mAP achieves 87.32%. Batch normalization not only speeds up the convergence but also makes the mAP improve by 1.71%. Data augmentation can increase the generalization ability of the model and suppress overfitting, while also improving the accuracy by 2.12%. Compared to the original anchors for optical object detection, the new anchor boxes obtained by K-means improve the mAP by 1.67%. Multiscale training strategy not only improves the multiscale detection capability of the model but also improves the mAP by 1.61%. Finally, through hybrid optimization, the mAP is also improved by 1.09% and has an extremely high mAP of 95.52% on SSDD. These improvement methods all play a role in SAR ship detection, and this can provide suggestions for the improvements in the related detection algorithms.

3.7. The Effect of Training from Scratch. To verify the effectiveness of training from scratch in the SAR field, a comparative experiment is conducted. When all the hyperparameters are set as the same, two training methods including pretraining and training from scratch are used to train our model. Figure 13 is some typical detection results (including both inshore and offshore scenes) for the models with two different training methods. It can be observed that compared with the bounding boxes of the pretrained model in Figures 13(b) and 13(e), the bounding boxes of the model trained from scratch in Figures 13(c) and 13(f) are closer to the ground truths. Thus, for SAR ship detection, training from scratch reduces the learning bias (1000 classes in ImageNet data set versus 2 classes in SAR ship data set) from the ImageNet

Table 9: The comparison of the different NMS methods.

NMS methods	Recall	Precision	mAP	$F1$
The proposed method + NMS	93.51%	93.42%	94.48%	93.29%
The proposed method + Soft-NMS	94.18%	94.54%	95.52%	94.36%

pretraining and reduces the influence of domain mismatch (single-channel versus three-channel) between SAR amplitude images and natural RGB images.

Table 8 is the detailed evaluation indexes for the models with different training methods. Compared with pretraining, the accuracy of the model trained from scratch is increased by 1.41%, and it has better performance in SAR ship detection tasks. The experimental results confirm the effectiveness of training from scratch in the SAR field and provide a possible approach to train the SAR detection model for other deep learning methods.

3.8. The Effect of Soft-NMS. To verify the effectiveness of Soft-NMS in the SAR field, a comparative experiment is also conducted. Figure 14 is some typical detection results (inshore scenes) for the two kinds of NMS methods. It can be observed from Figures 14(c) and 14(f) that the Soft-NMS has better detection performance for the dense inshore ships. The missing detections of the dense ships are less than the conventional NMS as in Figures 14(b) and 14(e).

Table 9 is the detailed evaluation indexes for different NMS methods. With the implementation of Soft-NMS, the model performs better, which achieves 1.04% performance gains in terms of mAP. The experimental results demonstrate that our method with Soft-NMS improves ship detection performance and obtains more accurate prediction results in spatial accuracy. This illustrates the superiority of the Soft-NMS method in the detection of SAR ships, and it also provides a way for other methods to improve the detection capabilities, especially for dense inshore ships.

3.9. Comparisons with the State-of-the-Arts. Some mainstream ship detection methods are compared with the proposed method including the CNN-based methods and the CFAR-based methods. The CNN-based detection methods include a variety of two-stage and one-stage algorithms [43]. And the CFAR-based methods include several advanced global algorithms based on the gamma or $G0$ clutter distribution [9–18]. It must be stated that all these methods are conducted on the same SSDD data set.

To verify the overall performance of the CNN-based methods quantitatively, mAP is adopted and the results of the comparisons are shown in Figure 15. As given in Figure 15, the mAP of the proposed method achieves 95.52%, which is 25.42% higher than Faster R-CNN, and nearly 17% higher than SSD and the improved Faster R-CNN [30]. Compared to YOLOv1, YOLOv2, and YOLOv3 [40], the mAP is also improved by 14%, 5%, and 0.2%, respectively. At the same time, some novel and superior methods are also used for comparisons, such as the network based on adaptive recalibration mechanism (MSARN) [36], dense attention pyramid networks (DAPN) [33], grid-based CNN (G-CNN) [39], and depthwise separable CNN (DS-CNN) [40]. The results show that our method has better performance than all the above methods and achieves a big improvement in the final mAP.

The CFAR algorithms and their improved methods are widely used in ship detections [9–18]. We also make a more detailed comparison on the related CFAR-based methods in both inshore and offshore scenes [33]. The results including precision, recall, and $F1$ score are displayed in Table 10. As can be observed in Table 10, the proposed method is superior to other methods in all evaluation indexes. Especially for inshore ship detection, the proposed method improves the accuracy by 45% over the most advanced two-stage CFAR. This shows our model is robust to the complex background and solves the problem that the traditional CFAR method is insensitive to inshore ships.

Compared with some other detection methods implemented on SSDD as shown in Figure 16, such as Reference [30], the detection speed is about 3 FPS, Reference [36] is about 35 FPS, and Reference [39] is about 48 FPS. Our methods have achieved a faster speed of 72 FPS than all the above methods. This shows the great application value on real-time SAR ship detection in the future.

It is worth mentioning that due to the differences in hardware GPU platforms and deep learning frameworks, the speed of the algorithms cannot be compared directly, which is also an issue that needs to be solved urgently. The unified hardware platform and deep learning framework are necessary to test the algorithms, and this will be done in our subsequent work.

3.10. Verification of Actual Scenario for Sentinel-1 and Gaofen-3. To further evaluate the practicality and generalization ability of our detection method, we validate our model on the actual images from both the Sentinel-1 and Gaofen-3 satellites. The Sentinel-1 image is supported by the European Space Agency (ESA) [58], and the Gaofen-3 images are obtained from the AirSARShip-1.0 data set provided by the Aerospace Information Research Institute of Chinese Academy of Sciences [59].

The ground truths of the Sentinel-1 image are marked by the Automatic Identification System (AIS) from the MarineTraffic website [60], which provides the precise positioning of the cooperative ships. Some representative AIS information of two typical ships of the Sentinel-1 image are given as shown in Figure 17. The ground truths of the Gaofen-3 images are marked by SAR experts [59]. These correct annotations increase the reliability of our experimental verifications.

The Sentinel-1 and Gaofen-3 images used in this study contain both offshore ships and inshore ships, which can be used to evaluate our algorithm comprehensively. Due to the wide swath of the original SAR images, we divide the original images into a number of small subimages as 416×416

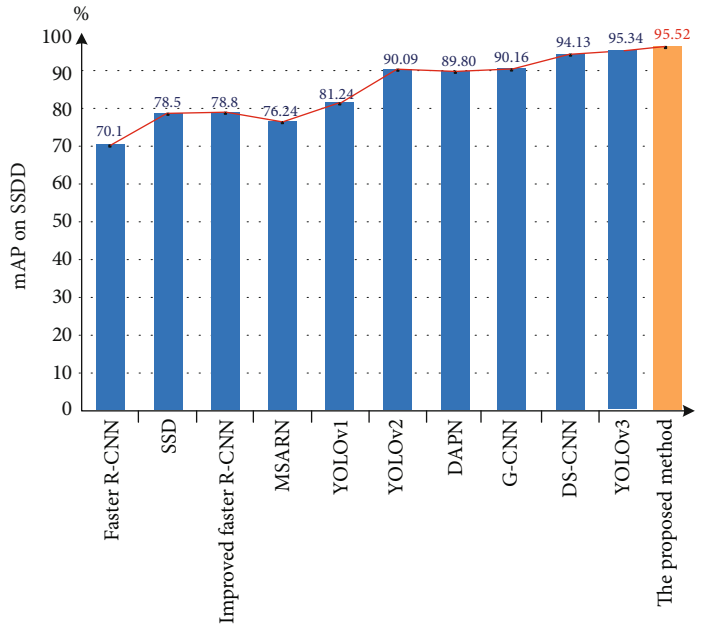


Figure 15: The mAP of different CNN-based methods on SSDD.

Table 10: Comparison with CFAR-based methods on the SSDD data set.

Scenes	Methods	Recall	Precision	$F1$
Inshore	Gamma-distribution CFAR	65.88%	39.23%	49.18%
	G0-distribution CFAR	75.59%	48.71%	59.24%
	Global two-stage CFAR	73.65%	50.53%	59.94%
	The proposed method	88.17%	95.49%	91.68%
Offshore	Gamma-distribution CFAR	90.92%	84.38%	87.53%
	G0-distribution CFAR	98.08%	92.73%	95.33%
	Global two-stage CFAR	96.15%	94.23%	95.18%
	The proposed method	99.23%	99.07%	99.15%

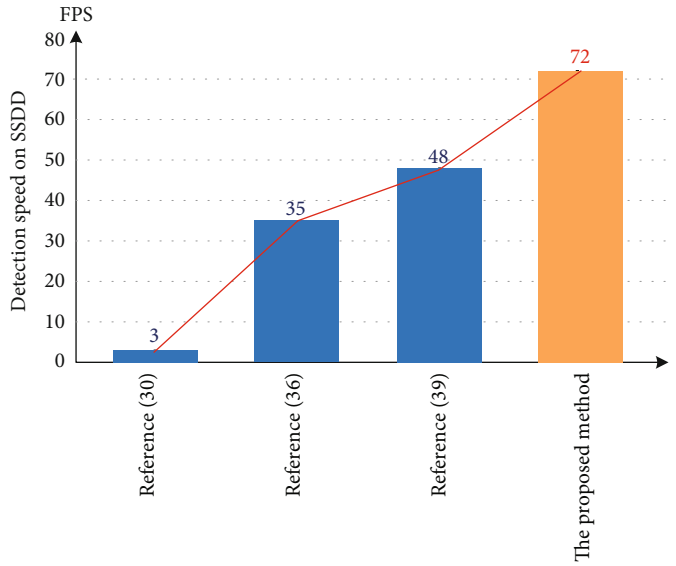


Figure 16: The comparison of speed on SSDD in different references.

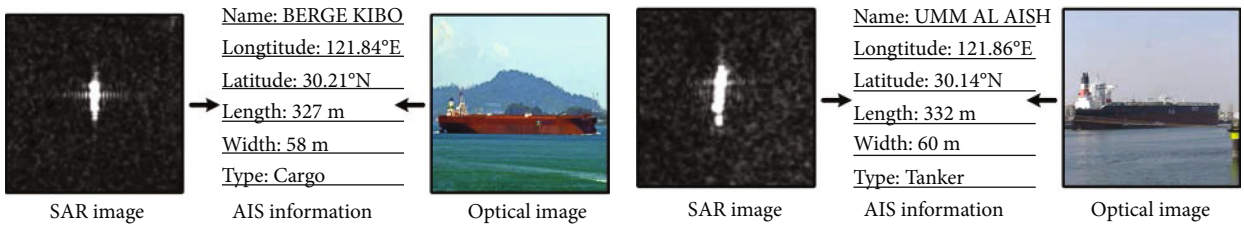


Figure 17: Some representative AIS information for two typical ships of the Sentinel-1 image. (a) A typical cargo and (b) a typical tanker.

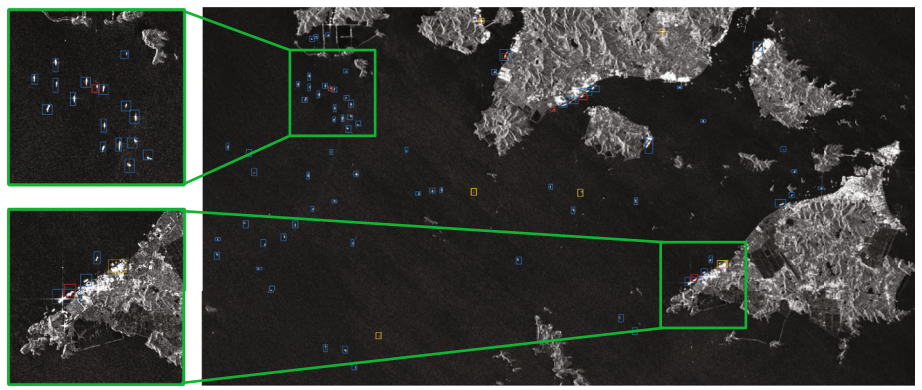


Figure 18: Ship detection results for Sentinel-1 (acquired 26 Feb 2016, C band, interferometric wide swath mode, 37~44°, 10 m × 10 m, original image ID: S1A_IW_GRDH_1SSV_20160226T095416_20160226T095441_004793_005F4B_6B09). The blue rectangles indicate detected ships. The orange rectangles denote false alarms. The red rectangles denote missed detections.

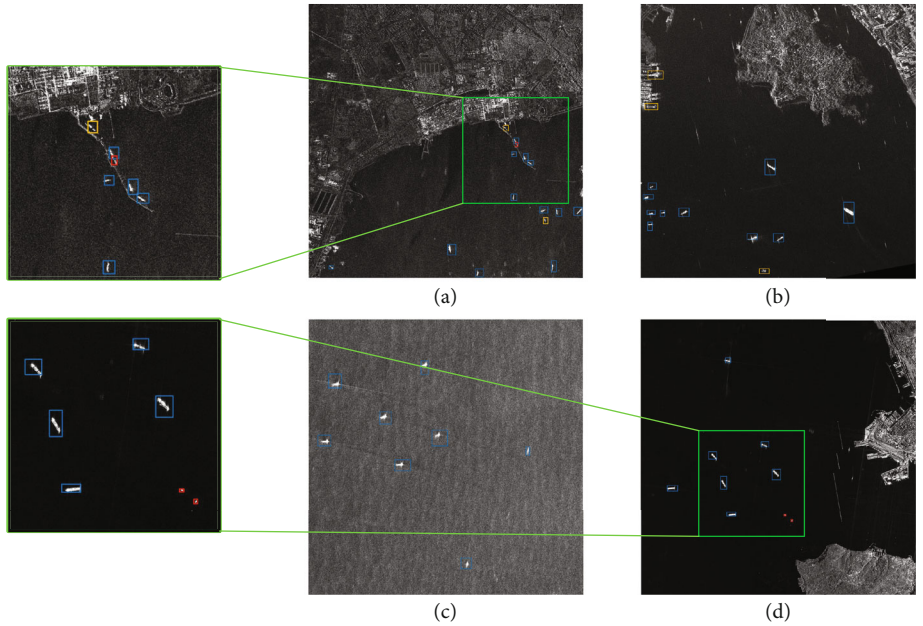


Figure 19: Ship detection results for Gaofen-3 (four selected images from AIR-SAR-SHIP-1.0 [59], C band, stripmap, and Spotlight mode). The blue rectangles indicate detected ships. The orange rectangles denote false alarms. The red rectangles denote missed detections.

manually for processing separately. Figures 18 and 19 are the final detection results for the Sentinel-1 and Gaofen-3 images, respectively. It can be observed that most ships are correctly detected and are marked in the blue rectangle. There are some false alarms for very tiny objects, and they are marked in the yellow rectangle. The missed detections are marked in red rectangle, and these are almost all dense inshore ships. This shows that the detection capability of very dense inshore ships in actual scenarios is insufficient and needs to be improved.

Table 11: The detailed ship test results for Sentinel-1 and Gaofen-3.

Payload	Sentinel-1	Gaofen-3
Ground truth	72	40
TP	67	37
FN	5	3
FP	6	5
Recall	93.06%	92.50%
Precision	91.78%	88.10%

Table 11 is the detailed evaluation indexes of the experimental verification results for both Sentinel-1 and Gaofen-3 images. For the Sentinel-1 image, there are 72 ships in the image, and 67 of them are correctly detected. For the Gaofen-3 images, there are 40 ships in the images, and 37 of them are correctly detected. It can be seen that the detection precision and recall for Sentinel-1 and Gaofen-3 have reached more than 90% and 88%, respectively. These demonstrate the robustness and superior migration ability of our method.

4. Conclusions

In this paper, a concise and efficient multiscale ship detection method based on deep CNNs is proposed to balance the accuracy and speed. The proposed network uses DarkNet-53 as a backbone for feature extraction and adopts a top-down pyramid structure for multiscale feature fusion. Meanwhile, strategies such as Soft-NMS, mix-up, and mosaic data augmentation, multiscale training, and hybrid optimization are combined. The whole pipeline of the one-stage method is a single network that can be optimized end-to-end, so the detection speed is fast. The related experiments on SSDD are carried out, and the mAP of the method reaches 95.52% with about 72 FPS. We also establish SSDD-offshore and SSDD-inshore data sets, respectively, for more precise evaluation. For offshore ships, the mAP of the proposed method reaches 99.50%, and for offshore ships, it reaches 92.80%. The results indicate that the method can detect multiscale ships in different scenes accurately, for both inshore and offshore ships. Besides, through a series of comparisons with the CNN-based and CFAR-based methods, the proposed method outperforms other state-of-the-art ship detectors. Furthermore, we have verified our method on real data of the Sentinel-1 and Gaofen-3 images and still achieve good performance. This demonstrates the robustness and practicality of the proposed method. The amount of SAR image data is increasing nowadays, however, the labeling data is relatively scarce. Our future work will focus on semisupervised learning methods and anchor-free detection methods in SAR ship detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

We thank the editors who handled the manuscript and the anonymous reviewers for their comments towards improving this manuscript. We thank Dr. Li et al. for providing SSDD data set. We thank European Space Agency (ESA) for providing Sentinel-1 images and MarineTraffic website for providing Automatic Identification System (AIS) messages. This work was supported by the Director's Foundation of Institute of Microelectronics, Chinese Academy of Sciences under Grant no. E0518101.

References

[1] S. Brusch, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, and B. van Schie, "Ship surveillance with TerraSAR-X," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 3, pp. 1092–1103, 2011.

[2] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6–43, 2013.

[3] C. Jackson and J. Apel, *Synthetic Aperture Radar Marine User's Manual*, U.S. Department of Commerce, Washington, DC, USA, 2004.

[4] Y. Xie, W. Dai, Z. Hu, Y. Liu, C. Li, and X. Pu, "A novel convolutional neural network architecture for SAR target recognition," *Journal of Sensors*, vol. 2019, Article ID 1246548, 9 pages, 2019.

[5] C. Santamaria and H. Greidanus, "Ambiguity discrimination for ship detection using Sentinel-1 repeat acquisition operations," in *Proc. 2015 IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, pp. 2477–2480, Milan, 2015.

[6] M. Yang and C. Guo, "Ship detection in SAR images based on lognormal ρ-metric," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1372–1376, 2018.

[7] M. Tello, C. Lopez-Martinez, and J. J. Mallorqui, "A novel algorithm for ship detection in SAR imagery based on the wavelet transform," *IEEE Geoscience and Remote Sensing Letters*, vol. 2, no. 2, pp. 201–205, 2005.

[8] A. C. Copeland, G. Ravichandran, and M. M. Trivedi, "Localized radon transform-based detection of ship wakes in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 1, pp. 35–45, 1995.

[9] K. Ji, X. Xing, H. Zou, and J. Sun, "A novel variable index and excision CFAR based ship detection method on SAR imagery," *Journal of Sensors*, vol. 2015, Article ID 437083, 10 pages, 2015.

[10] Y. Jin and B. Friedlander, "A CFAR adaptive subspace detector for second-order Gaussian signals," *IEEE Transactions on Signal Processing*, vol. 53, no. 3, pp. 871–884, 2005.

[11] D. R. Iskander and A. M. Zoubir, "Estimation of the parameters of the K-distribution using higher order and fractional moments [radar clutter]," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, pp. 1453–1457, 1999.

[12] Y. Xu, C. Hou, S. Yan, J. Li, and C. Hao, "Fuzzy statistical normalization CFAR detector for non-rayleigh data," *IEEE*

Transactions on Aerospace and Electronic Systems, vol. 51, no. 1, pp. 383–396, 2015.

[13] T. Liu, J. Zhang, G. Gao, J. Yang, and A. Marino, "CFAR ship detection in polarimetric synthetic aperture radar images based on whitening filter," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 58–81, 2020.

[14] M. Weiss, "Analysis of some modified cell-averaging CFAR processors in multiple-target situations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-18, no. 1, pp. 102–114, 1982.

[15] G. Gao, L. Liu, L. Zhao, G. Shi, and G. Kuang, "An adaptive and fast CFAR algorithm based on automatic censoring for target detection in high-resolution SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 6, pp. 1685–1697, 2009.

[16] M. Shor and N. Levanon, "Performances of order statistics CFAR," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 27, no. 2, pp. 214–224, 1991.

[17] X. Leng, K. Ji, S. Zhou, and X. Xing, "Fast shape parameter estimation of the complex generalized Gaussian distribution in SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 11, pp. 1933–1937, 2020.

[18] X. Leng, K. Ji, S. Zhou, and X. Xing, "Ship detection based on complex signal kurtosis in single-channel SAR imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6447–6461, 2019.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, MA, USA, 2016.

[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. (CVPR)*, pp. 580–587, Columbus, OH, USA, 2014.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[24] 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.*, pp. 379–387, Barcelona, Spain, 2016, https://arxiv.org/abs/1605.06409.

[25] 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)*, pp. 779–788, Las Vegas, NV, USA, 2016.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, C.-Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pp. 21–37, Amsterdam, The Netherlands, 2016.

[27] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[28] J. Redmon and A. Farhadi, "YOLOv3: an incremental improvement," 2018, https://arxiv.org/abs/1804.02767.

[29] A. Bochkovskiy, C. Wang, and H. Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020, https://arxiv.org/pdf/2004.10934.

[30] J. Li, C. Qu, and J. Shao, "Ship detection in SAR images based on an improved faster R-CNN," in *Proc. BIGSARDATA*, pp. 1–6, Beijing, China, 2017.

[31] J. Jiao, Y. Zhang, H. Sun et al., "A densely connected end-to-end neural network for multiscale and multiscene SAR ship detection," *IEEE Access*, vol. 6, pp. 20881–20892, 2018.

[32] J. Zhao, Z. Zhang, W. Yu, and T.-K. Truong, "A cascade coupled convolutional neural network guided visual attention method for ship detection from SAR images," *IEEE Access*, vol. 6, pp. 50693–50708, 2018.

[33] Z. Cui, Q. Li, Z. Cao, and N. Liu, "Dense attention pyramid networks for multi-scale ship detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8983–8997, 2019.

[34] 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.

[35] Q. An, Z. Pan, L. Liu, and H. You, "DRBox-v2: an improved detector with rotatable boxes for target detection in SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 8333–8349, 2019.

[36] C. Chen, C. He, C. Hu, H. Pei, and L. Jiao, "MSARN: a deep neural network based on an adaptive recalibration mechanism for multiscale and arbitrary-oriented SAR ship detection," *IEEE Access*, vol. 7, pp. 159262–159283, 2019.

[37] Z. Deng, H. Sun, S. Zhou, and J. Zhao, "Learning deep ship detector in SAR images from scratch," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 4021–4039, 2019.

[38] 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 Sensing*, vol. 11, no. 7, p. 786, 2019.

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

[40] T. Zhang, X. Zhang, J. Shi, and S. Wei, "Depthwise separable convolution neural network for high-speed SAR ship detection," *Remote Sensing*, vol. 11, no. 21, p. 2483, 2019.

[41] L. Zhai, Y. Li, and Y. Su, "Inshore ship detection via saliency and context information in high-resolution SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1870–1874, 2016.

[42] X. X. Zhu, D. Tuia, L. Mou et al., "Deep learning in remote sensing: a comprehensive review and list of resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[43] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: a survey," 2019, https://arxiv.org/abs/1905.05055.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016.

[45] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 936–944, Honolulu, HI, USA, 2017.

[46] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1–9, Boston, MA, USA, 2015.

[47] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," 2018, https://arxiv.org/abs/1505.00853.

[48] Z. Chen, L. Deng, G. Li et al., "Effective and efficient batch normalization using a few uncorrelated data for statistics estimation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 348–362, 2020.

[49] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS–improving object detection with one line of code," in *Proc. IEEE Conf. Comput. Vis. (ICCV)*, pp. 5562–5570, Venice, Italy, 2017.

[50] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, The MIT Press, 2012.

[51] K. He, R. Girshick, and P. Dollár, "Rethinking ImageNet pretraining," 2018, https://arxiv.org/abs/1811.08883.

[52] PaddlePaddle, "Parallel distributed deep learning: machine learning framework from industrial practice," https://www.paddlepaddle.org/.

[53] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[54] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," 2017, https://arxiv.org/abs/1710.09412.

[55] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[56] N. Keskar and R. Socher, "Improving generalization performance by switching from Adam to SGD," 2017, https://arxiv.org/abs/1712.07628.

[57] T. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proc. of the 13th Europ. Conf.*, pp. 740–755, Zurich, Switzerland, 2014.

[58] ESA, "Copernicus open access hub," 2020, https://scihub.copernicus.eu/.

[59] X. Sun, Z. Wang, Y. Sun, W. Diao, Y. Zhang, and K. Fu, "AirSARSHIP–1.0: high resolution SAR ship detection dataset," *Journal of Radars*, vol. 8, pp. 852–862, 2019.

[60] Marine Traffic, "Historical AIS data," 2020, https://www.marinetraffic.com/zh/p/ais-historical-data.