remote sensing

MDPI

Article

Ship Detection in SAR Images Based on Multi-Scale Feature Extraction and Adaptive Feature Fusion

Kexue Zhou [1], Min Zhang [1,*], Hai Wang [1] and Jinlin Tan [1,2]

[1] School of Aerospace Science & Technology, Xidian University, Xi'an 710126, China; kxzhou@stu.xidian.edu.cn (K.Z.); wanghai@mail.xidian.edu.cn (H.W.); linsheng@stu.xidian.edu.cn (J.T.)
[2] Shaanxi Academy of Aerospace Technology Application Co., Ltd., Xi'an 710199, China
* Correspondence: minzhang@xidian.edu.cn

Abstract: Deep learning has attracted increasing attention across a number of disciplines in recent years. In the field of remote sensing, ship detection based on deep learning for synthetic aperture radar (SAR) imagery is replacing traditional methods as a mainstream research method. The multiple scales of ship objects make the detection of ship targets a challenging task in SAR images. This paper proposes a new methodology for better detection of multi-scale ship objects in SAR images, which is based on YOLOv5 with a small model size (YOLOv5s), namely the multi-scale ship detection network (MSSDNet). We construct two modules in MSSDNet: the CSPMRes2 (Cross Stage Partial network with Modified Res2Net) module for improving feature representation capability and the FC-FPN (Feature Pyramid Network with Fusion Coefficients) module for fusing feature maps adaptively. Firstly, the CSPMRes2 module introduces modified Res2Net (MRes2) with a coordinate attention module (CAM) for multi-scale features extraction in scale dimension, then the CSPMRes2 module will be used as a basic module in the depth dimension of the MSSDNet backbone. Thus, our backbone of MSSDNet has the capabilities of features extraction in both depth and scale dimensions. In the FC-FPN module, we set a learnable fusion coefficient for each feature map participating in fusion, which helps the FC-FPN module choose the best features to fuse for multi-scale objects detection tasks. After the feature fusion, we pass the output through the CSPMRes2 module for better feature representation. The performance evaluation for this study is conducted using an RTX2080Ti GPU, and two different datasets: SSDD and SARShip are used. These experiments on SSDD and SARShip datasets confirm that MSSDNet leads to superior multi-scale ship detection compared with the state-of-the-art methods. Moreover, in comparisons of network model size and inference time, our MSSDNet also has huge advantages with related methods.

Keywords: adaptive feature fusion; synthetic aperture radar (SAR); multi-scale ships detection; YOLOv5

Citation: Zhou, K.; Zhang, M.; Wang, H.; Tan, J. Ship Detection in SAR Images Based on Multi-Scale Feature Extraction and Adaptive Feature Fusion. *Remote Sens.* **2022**, *14*, 755. https://doi.org/10.3390/rs14030755

Academic Editors: M. Pilar Jarabo Amores and David de la Mata Moya

Received: 21 December 2021
Accepted: 3 February 2022
Published: 6 February 2022

1. Introduction

In remote sensing, Synthetic Aperture Radar (SAR), which is an active microwave imaging radar that can observe the surface of the earth day and night [1,2], plays a significant role in marine traffic monitoring. In recent years, many countries have developed their own spaceborne technology, such as Germany's TerraSAR-X, China's Gaofen-3 and Canada's RADARSAT-2. Such efforts make object detection of SAR images an increasingly attractive topic.

Deep learning-based object detection for natural images has witnessed a growing number of publications [3–10], in many of which dividing object detection into one-stage and two-stage is a common way. The one-stage object detection algorithms treat object detection as a regression problem and obtain bounding box coordinates and class probabilities from image pixels. The typical algorithms are the You Only Look Once (YOLO) series [11–14], Single Shot MultiBox Detector (SSD) [6], and RetinaNet [9], etc. The two-stage object

detection algorithms firstly generate region proposals as the potential bounding boxes and construct a classifier to classify these bounding boxes. After that, the bounding boxes will be refined through post-processing, and finally, duplicate detections will be eliminated. The typical two-stage algorithms are Fast RCNN [7], Faster RCNN [8] and Mask RCNN [10], etc. In general, the two-stage object detection algorithms are more accurate than the one-stage ones, but the one-stage methods are faster and simpler to train.

Inspired by deep learning's great power in object detection, researchers have introduced deep learning into image processing in remote sensing [15–18]. Image processing in SAR is one of the most important fields in remote sensing. Ship detection with multi-scale features [19–22] has gotten more and more attention in recent years. Liu et al. [23] constructed a ship proposal generator to solve the multi-scale problem of ships in SAR images, which can get the highest recall and quality of proposals. The serious missed detection problem of small-scale ships in SAR images had a terrible influence on the performance of object detection, Kang et al. [22] solved this problem by constructing a context-based convolutional neural network with multi-layer fusion, in which a high-resolution region proposal network (RPN) was used to generate high-quality region proposals, and an object detection network with contextual features can obtain useful contextual information. Fu et al. [24] balanced semantically the multiple features across different levels by proposing an attention-guided balanced pyramid, which can focus on small ships in complex scenes efficiently. Cui et al. [25] adopted an attention mechanism to focus on multi-scale ships, in which a dense attention pyramid network was proposed, namely DAPN. The convolutional block attention module in DAPN used channel attention and spatial attention to extract resolution and semantic information and highlight salient features. Additionally, in the study of [26–30], different methods were proposed to detect multi-scale ships in SAR images and had achieved satisfying detection results. Although the ship detection algorithms mentioned above had a significant improvement in detection performance, their multi-scale features fusion only fused the feature maps directly. In this way, the fused feature layers are restricted by each other, which is not appropriate for ships of different sizes. Releasing the constraint of feature fusion directly is beneficial to improve the detection performance of multi-scale objects.

This paper describes the design and implementation of a multi-scale ship detection network to achieve an excellent detection performance in SAR images. We firstly construct a CSPMRes2 (Cross Stage Partial network with Modified Res2Net) module for better feature extraction of ships. CSPMRes2 not only has the capability of multi-scale features extraction, but can also model inter-channel relationships and capture long-range dependencies with precise positional information of the feature map. In addition, aiming to directly overcome the shortcoming of feature fusion directly, the fusion proportion of feature maps is considered. Then we construct a feature pyramid network architecture for multi-scale ships detection, namely the FC-FPN (Feature Pyramid Network with Fusion Coefficients). The fusion coefficients in our FC-FPN are set for each feature map participating in fusion and are learned from the training phase of the ship detection network. After fusing feature maps, we pass the output through the CSPMRes2 module to equip FC-FPN with powerful features extraction capability. On the other hand, we also take the model size of the ship's detection network into account, then adopt YOLOv5 with a small model size (denoted as YOLOv5s) as the ship objects detection framework. Finally, we construct the MSSDNet by applying the CSPMRes2 module and FC-FPN module into YOLOv5s. Benefitted from the design of MSSDNet, the results of experiments on SSDD [20] and SARShip [31] datasets illustrate that our MSSDNet has a significant improvement in detection performance with smaller model size and faster inference time. The contributions of this work are summarized below.

1. We construct the MSSDNet with a small model size while having better speed and accuracy compared with the YOLOv5s baseline and other methods.

2. A CSPMRes2 module is proposed to extract the multi-scale discriminative features, which not only possess features extraction capability of 'scale' dimension but can also

capture the relationships of inter-channels and obtain salient information with precise spatial location information.

3. We construct an FC-FPN module that a learnable fusion coefficient set for each feature map participating in fusion to fuse feature maps adaptively, and we conduct the experiments of fusion coefficient to explore how the fusion coefficients affect the detection of ships.

The rest of this paper is arranged as follows: Section 2 describes the proposed network. Section 3 makes an analysis of the result of experiments and a comparison between the proposed network and other algorithms. Section 4 discusses some phenomena according to the experimental results. Finally, Section 5 gives conclusions about this paper.

2. The Proposed Method

The sample matching method in the working pipeline of MSSDNet, which is based on the shape between anchor boxes and ground truth, is different from the general ones based on the Intersection over Union (IoU) between the ground truth and anchor boxes. The sample matching method is shown in Figure 1, where *w*g and *hg* represent the width of ground truth, respectively, and *wi* and *hi* represent the width of the three anchor boxes, respectively. A SAR image firstly is resized to a fixed spatial resolution, and then will be divided into S × S grid cells. Each grid cell will set anchor boxes with different aspect ratios. If the width and height of the object match anchor boxes within an allowed range, then the anchor boxes will be responsible for detecting that object, while other anchor boxes will be the background. One object is allowed to have multiple anchor boxes. After sample matching, the bounding box can be obtained by predicting the offset of anchors and objects. The prediction of a bounding box has 6 components: *class*, *x*, *y*, *w*, *h*, *confidence*. The *class* represents which category it belongs to, the *x*, *y* coordinates represent the center offset of the bounding box relative to ground truth, the *w* and *h* are the width and height of the bounding box, the *x*, *y*, *w*, *h* are normalized to 0 and 1 according to the image size. The *confidence* score represents the probability that a bounding box contains an object. If there is no object in the bounding box, the *confidence* score should be zero. Furthermore, IoU between the ground truth and the predicted box indicates how close the predicted box is to the ground truth. The closer between the predicted box and ground truth, the more likely the predicted box contains an object. Thus, we make the *confidence* score of the predicted box equal to the IoU between the ground truth and the predicted box.

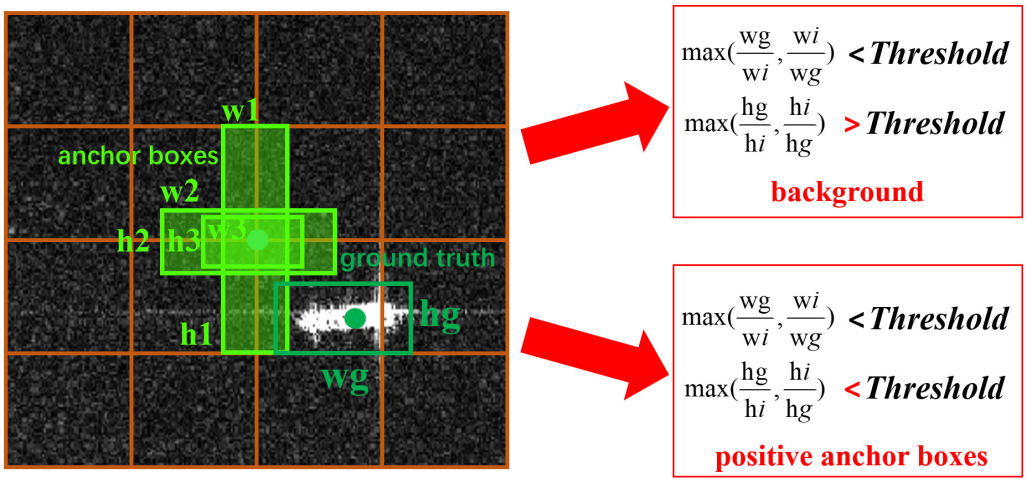


Figure 1. The sample matching of MSSDNet. The dark green box is ground truth, light green boxes are anchor boxes. The width and height of the anchor boxes are compared with the ground truth to decide if the anchor box is a background or positive anchor box.

The overview of the proposed MSSDNet is illustrated in Figure 2. Compared with the original YOLOv5s, we reconstruct the backbone of YOLOv5s by introducing our CSPMRes2

module and replacing the FPN of YOLOv5s with the FC-FPN module. The CSPMRes2 module is responsible for extracting features better, and FC-FPN fuses the feature maps adaptively. In the phase of testing, we use the COCO metric as the evaluation standard. We will describe the key modules of MSSDNet in detail.

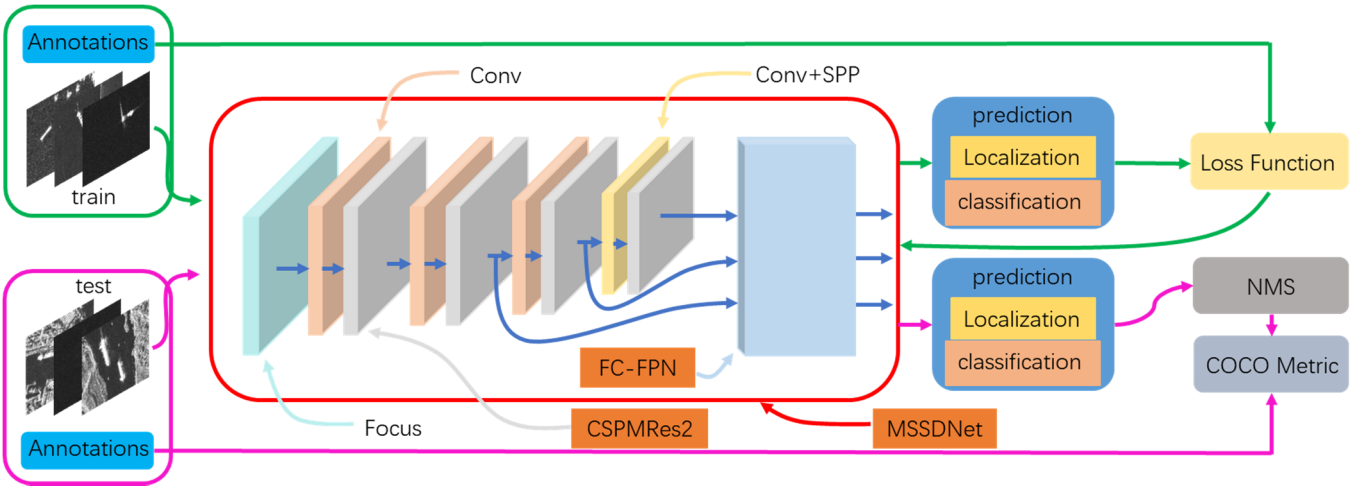


Figure 2. The overview of MSSDNet. Compared to YOLOv5s, we mainly improve the feature extraction capability of the backbone by using the CSPMRes2 module, and we also make FC-FPN replace the original FPN version for feature fusion adaptively.

2.1. CSPMRes2 Module

In order to increase the receptive field range of feature maps, several MRes2 submodules are introduced into the CSPMRes2 module as the feature extraction of scale dimension, as shown in Figure 3. In the figure, the red block represents the MRes2 module, the pink block represents the convolution module, and other blocks with different colors represent different feature maps. The input of the CSPMRes2 module is split into two branches through channel $x = [x0', x0'']$. Between $x0'$ and $x0''$, the former will go through MRes2 submodules; the latter is linked to the end of the CSPMRes2 module. The outputs of MRes2 submodules, $[x0', x_1, \ldots, x_k]$, will undergo a general convolution module to generate an output, x_T, then the $x0''$ and x_T will be concatenated together, after a general convolution module as the output of CSPMRes2 module. The equations of forward propagation and weight update of the CSPMRes2 module are shown in Equations (1) and (2), respectively. In the equations, g is gradient information, w is weights, and η is the learning rate.

$$\begin{aligned} x_k &= w_k * [x0', x_1, \ldots, x_{k-1}] \\ x_T &= w_T * [x0', x_1, \ldots, x_k] \\ x_U &= w_U * [x0'', x_T] \end{aligned} \tag{1}$$

$$\begin{aligned} w_k' &= f_k(w_k, \{g0', g_1, \ldots, g_{k-1}\}) \\ w_T' &= f_T(w_T, \{g0', g_1, \ldots, g_k\}) \\ w_U' &= f_U(w_U, \{g0'', g_T\}) \end{aligned} \tag{2}$$

$$\begin{aligned} f_k &= w_k - \eta * \{g0', g_1, \ldots, g_{k-1}\} \\ f_T &= w_T - \eta * \{g0', g_1, \ldots, g_k\} \\ f_U &= w_U - \eta * \{g0'', g_T\} \end{aligned} \tag{3}$$

We can see that the gradients of MRes2 submodules are integrated separately, and the bypassed $x0''$ is also integrated separately. CSPMRes2 module not only possesses characteristics of feature reuse but also reduces the number of duplicate gradient information [32].

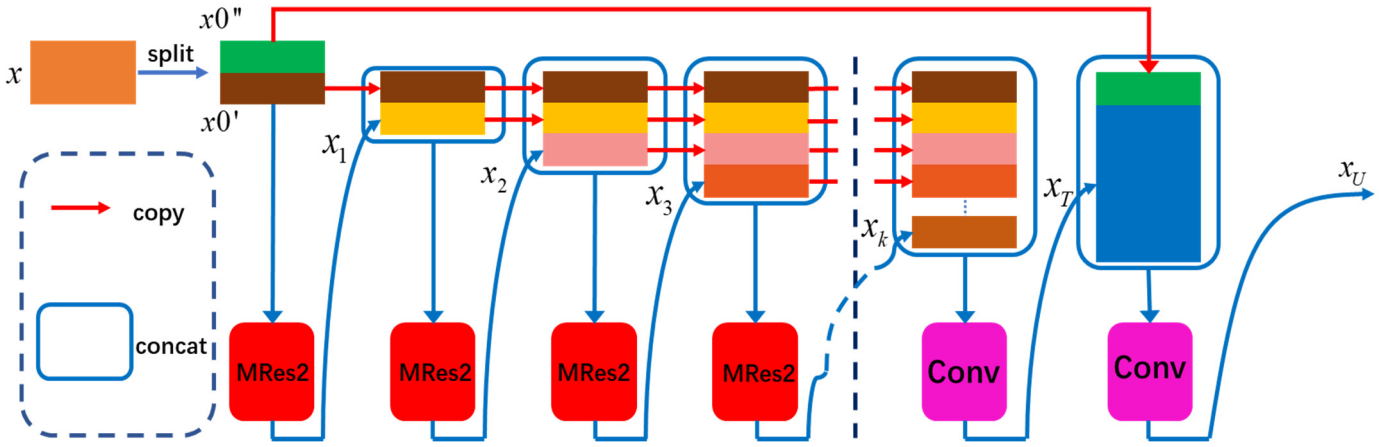


Figure 3. The CSPMRes2 module with MRes2 submodule. The input feature map is split into two parts: one goes through the MRes2 submodule, and another is bypassed to the end of the CSMRes2 module.

In the MRes2 module, as shown in Figure 4, the input will go through n 1×1 convolutions, respectively, to change the channels of feature maps, after which, we obtain n feature subsets, denoted as p_i, where $i = \{1, 2, \ldots n\}$. Each p_i has the same number of channels and spatial resolution, where the number of channels is $1/n$ of the input channels. Except for the feature subset p_1, each p_i will go through a convolution with kernel size 3×3, denoted as K_i. Moreover, except for the feature subset p_1 and p_2, each p_i will undergo a coordinate attention module (CAM) [33], denoted as C_i. We denote the output of C_i by t_i and the output of K_i by y_i.Thus, the t_i and y_i can be written as:

$$t_i = C_i(y_{i-1} + p_i) \quad 3 \leq i \leq n \tag{4}$$

$$y_i = \begin{cases} p_i & i = 1 \\ K_i(p_i) & i = 2 \\ t_i & 3 \leq i \leq n \end{cases} \tag{5}$$

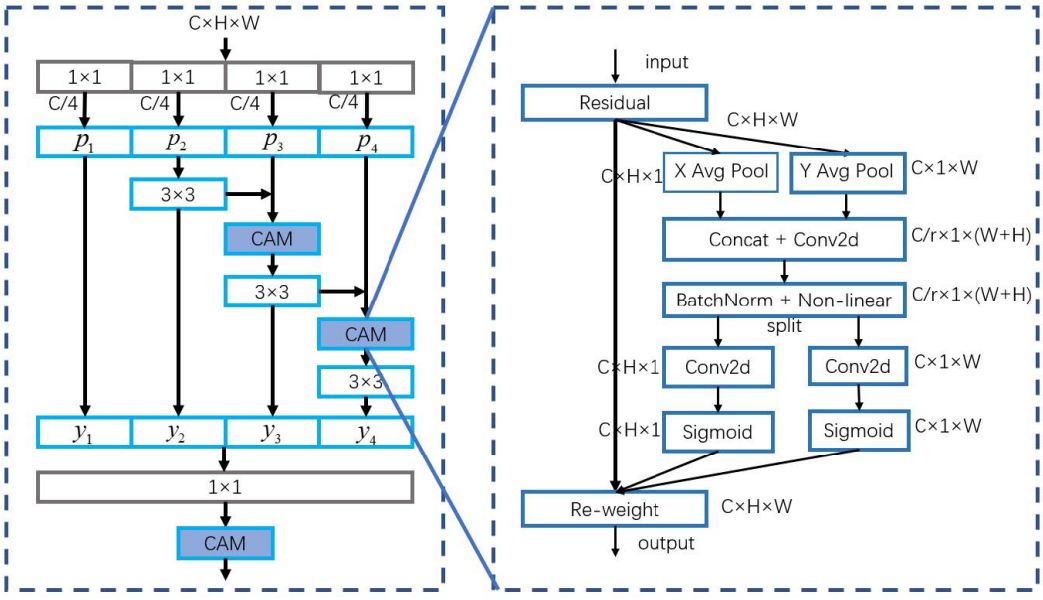


Figure 4. The MRes2 module introduced coordinate attention to capture relationships of inter-channels and obtain salient information with precise spatial location information.

Combine Equation (4) and (5), then we get:

$$y_i = \begin{cases} p_i & i = 1 \\ K_i(p_i) & i = 2 \\ C_i(y_{i-1} + p_i) & 3 \leq i \leq n \end{cases} \tag{6}$$

For optimized fuse information at the dimension of 'scale', we concatenate all the y_i, denoted by y, i.e., $y = [y_1, y_2, \ldots, y_n]$, then pass y through a 1×1 convolution. Finally, in order to capture relationships of inter-channels and obtain salient information with precise spatial location information, after making the feature map go through a 1×1 convolution, we pass it through a CAM as the final output of the MRes2 module.

The strategy of separation and combination makes the convolutions process features efficient. The CSPMRes2 module not only has multi-scale feature extraction capability [34] but also reduces a lot of duplicate gradient information.

2.2. FC-FPN Module

The architecture of the proposed FC-FPN is shown in Figure 5. A learnable fusion coefficient for each feature map participating in the fusion is set for getting adaptive feature fusion between different feature maps. For better extraction of multi-scale features, we make the output of adaptive feature fusion go through a CSPMRes2 module.

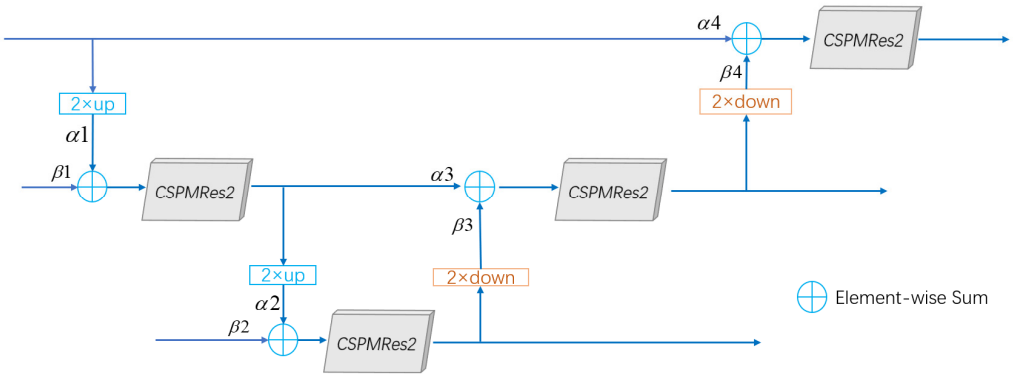


Figure 5. FC-FPN architecture. We set a learnable fusion coefficient for each feature map participating in feature fusion.

Assuming that f_1 and f_2 represent feature maps participating in feature fusion, α and β are the fusion coefficients of f_1 and f_2, respectively. We can get an output of features fusion as shown in Equation (7):

$$f = \alpha \cdot f_1 + \beta \cdot f_2 \tag{7}$$

The coefficient α and β can respectively adjust the contribution of f_1 and f_2. It can get the best features fusion result by making the α and β learnable. Furthermore, α and β are limited within a fixed range to ensure the stability of network training. In this paper, this optimal learning range is obtained by conducting the experiments of fusion coefficients on the SSDD dataset.

2.3. Architecture of MSSDNet

The detailed architecture of MSSDNet is illustrated in Figure 6, which is the application of the CSPMRes2 module and FC-FPN module in YOLOv5s. We can see the numbers and locations of the CSPMRes2 module and FC-FPN module. In the backbone of MSSDNet, the output of the CSPMRes2 module is the input of FC-FPN and adopts three feature maps with different scales to detect multi-scale ships in SAR images. In order to improve the features representation capability of FC-FPN, we use a CSPMRes2 module after each feature fusion operation.

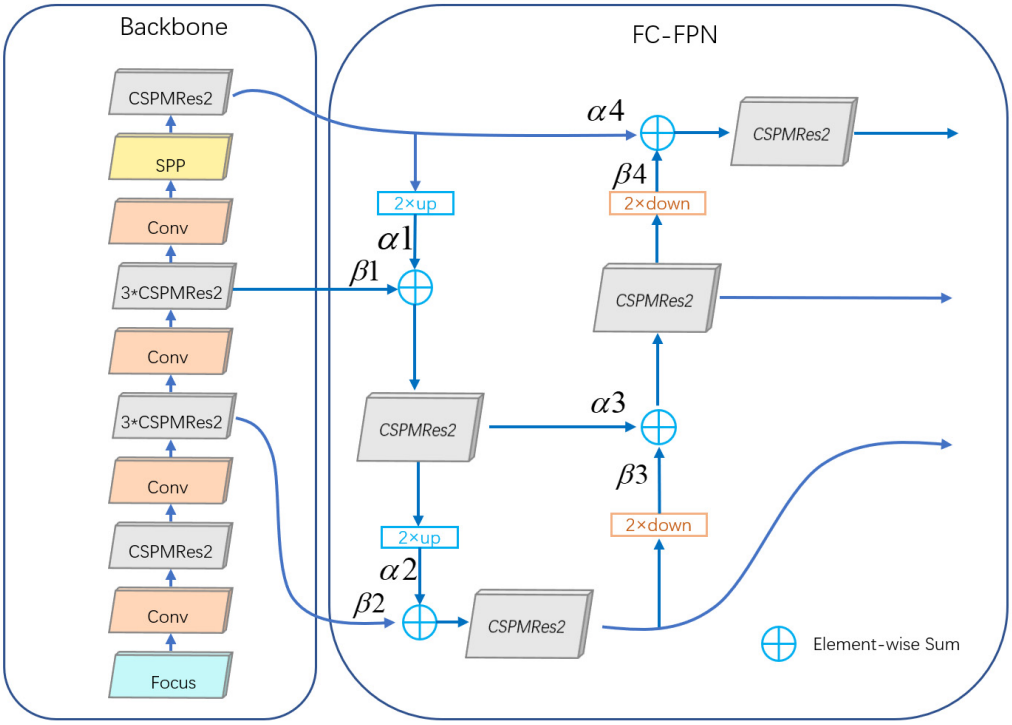


Figure 6. The architecture of MSSDNet. The detailed MSSDNet shows the numbers and location of the CSPMRes2 module and FC-FPN module.

3. Results

Two datasets: SSDD and SARShip are used to verify the effectiveness of the proposed method. Our method also makes a comparison with the other deep learning-based object detection algorithms: BorderDet [35], DeFCN [36], GFocalV2 [37], OTA [38], YOLOF [39] and PAA [40]. We conduct all experiments by using a PC with Intel® CoreTM i7-9800X CPU @3.80GHz ×16 and 32 GB of memory, and NVIDIA GeForce RTX 2080Ti GPU with 12GB of memory. The operating system is a 64-bit Ubuntu 18.04.5 LTS.

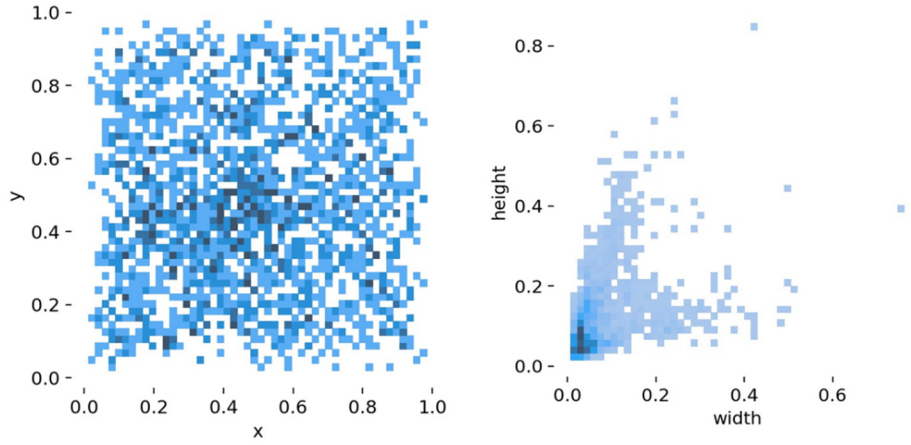
3.1. Experiment Settings

Since MSSDNet is constructed based on YOLOv5s, we use the experimental result of YOLOv5s as our baseline. The initial learning rate is set to 0.01, the optimizer is SGD, anchor aspect ratios are from the k-means algorithm, and the thresholds of NMS are set to 0.5. We set the size of the images to 512 × 512 for SSDD, and 256 × 256 for SARShip. Although the image size setting of SSDD and SARShip is different, there is no impact on the MSSDNet, which is because the size of network layers will change as the size of images change. To enhance the diversity of the training dataset, flip horizontal, and mosaic data augmentations are adopted in the phase of training. In addition, our model is trained from scratch, instead of using a pre-trained model.

3.2. Experiment Datasets

3.2.1. SSDD Dataset

There are 1160 images with a total of 2540 ships in the SSDD dataset. The average number of ships per image is 2.19. The dataset has a similar procedure to process bounding boxes and label annotations with PASCAL VOC [41]. We divide the training set and the testing set with the proportion of 8:2, and Figure 7 shows the visualization of the ship distribution. The center and width/height of ships are normalized to 0 and 1. The labels of each image of SSDD are stored in a file with the suffix xml, which will be converted into the file with the suffix txt required by MSSDNet.

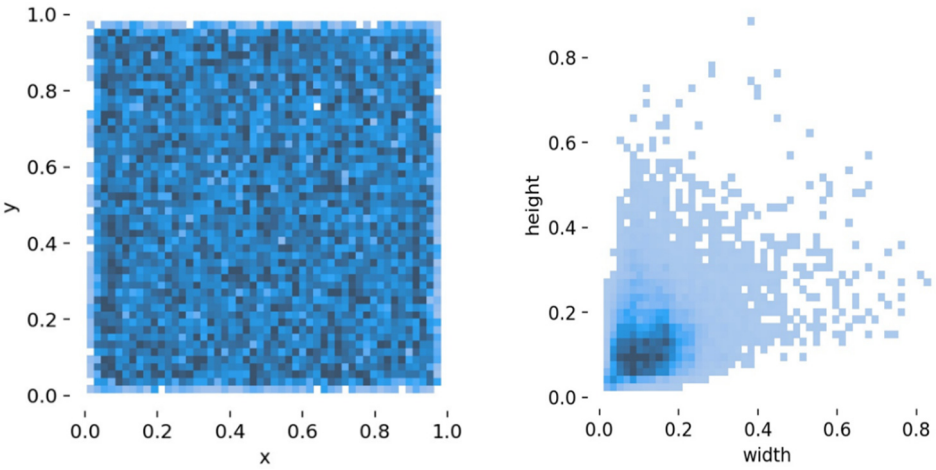


(**a**) (**b**)

Figure 7. Objects distribution of SSDD. (**a**) is the distribution of objects center, (**b**) is the distribution of objects width and height.

3.2.2. SARShip Dataset

The SARShip dataset contains 39,729 images, which contains a total of 50,885 ships. The image's size is 256 × 256. The dataset comes from the Gaofen-3 satellite and Eurasian Sentinel-1 satellite. Image resolutions range from 1.7 m to 25 m, and the polarization modes include HH, HV, VH, and VV. The dataset scenes include ports, inshore, islands, and offshore. The ships include oil tankers, bulk carriers, large container ships, and fishing vessels. The dataset is divided into the training set and the testing set with the proportion of 8:2. Figure 8 visualizes the distribution of ships, and the center and the width/height of ships are normalized to 0 and 1. The labels of each image in SARShip datasets are stored in a file with the suffix txt, which satisfies the label format required by MSSDNet.



(**a**) (**b**)

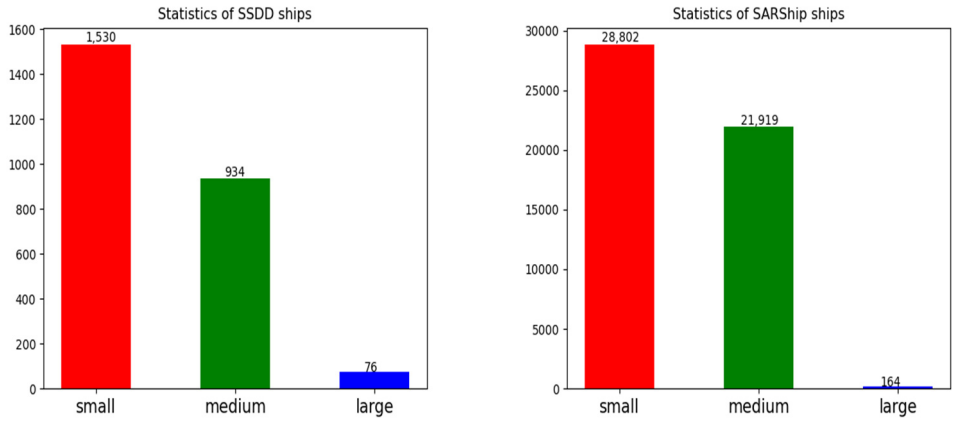
Figure 8. Objects distribution of SARShip. (**a**) is the distribution of objects center, (**b**) is the distribution of objects width and height.

In order to visualize the number of different size ships in the dataset, we make a statistic in terms of the definitions of COCO metrics [42] (as shown in Table 1) for different size ships, and display them in a histogram, as shown in Figure 9. Figure 9a shows that there are 1530 small ships, 934 medium ships, and 76 large ships in the SSDD dataset. Figure 9b shows that there are 28,802 small ships, 21,919 medium ships, and 164 large ships

in the SARShip dataset. In both the SSDD dataset and the SARShip dataset, the majority are the small and medium ships, while the large ships are very few.

Table 1. The meanings of COCO Metrics.

Metric	Meaning
AP	IoU = 0.50:0.05:0.95
AP_{50}	IoU = 0.50
AP_{75}	IoU = 0.75
AP_S	area < 32^2
AP_M	32^2 < area < 96^2
AP_L	area > 96^2



(**a**) (**b**)

Figure 9. Statistics of different size ships in the dataset. (**a**) statistics of SSDD ships, (**b**) statistics of SARShip ships.

3.3. Experiments on SSDD

According to the settings of experiments and evaluation standards, we conducted our experiments on SSDD, and the results are shown in Table 2. The MSSDNet has improved most of the COCO metrics compared with the YOLOv5s baseline. Especially, the values of AP_{75}, AP_S, AP_M, and AP_L metrics have significant improvement of 1.6%, 1.3%, 1.0%, and 1.4%, respectively. MSSDNet also has an improvement of 0.9% in the AP metric compared with YOLOv5s. The values of COCO metrics have demonstrated that MSSDNet can improve the multi-scale ships detection performance efficiently compared with the YOLOv5s baseline.

Table 2. The experimental result of MSSDNet on SSDD.

Methods	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
YOLOv5s	60.2	95.4	69.3	54.1	69.0	69.0
MSSDNet	61.1	95.6	70.9	55.4	70.0	70.4

As shown in Table 3, we performed the experiments on SSDD for some latest object detection methods. It can be seen that MSSDNet gets the best AP result of 61.1% compared with the other methods. In the methods of participation in comparison, OTA, and YOLOF have the greater results. OTA works better on medium ships but has poor performance in small and large ships compared with MSSDNet. YOLOF exceeds MSSDNet by 2.3% in the AP_L metric, but the values of AP_S and AP_M, AP_{75} metrics are far below MSSDNet, which means MSSDNet can obtain more precise object location information than YOLOF. The experimental results on SSDD have demonstrated the MSSDNet can efficiently handle the

detection of different size ships in SAR images, and its detection performance surpasses other great methods.

Table 3. The results of different methods on SSDD.

Methods	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
BorderDet	57.5	93.2	65.3	51.6	66.2	64.8
DeFCN	55.5	91.9	62.2	50.7	66.1	50.4
GFocalV2	56.2	92.1	64.1	51.5	65.9	61.1
OTA	59.1	93.3	69.0	52.5	70.1	63.4
YOLOF	59.2	94.5	65.8	53.0	68.8	72.7
PAA	56.0	91.6	64.0	51.1	65.7	53.1
YOLOv5s	60.2	95.4	69.3	54.1	69.0	69.0
MSSDNet	61.1	95.6	70.9	55.4	70.0	70.4

Figure 10 presents the results of methods of participation in comparison. The green boxes indicate the ground truths, the red boxes represent the predicted boxes, the yellow boxes represent missed detections, and the blue boxes represent false detections. In order to make the displayed detection results more representative, we select six detection images with complex backgrounds. It can be seen that BorderDet, DeFCN, and GFocalV2 have the worst performance in the number of missed detections. Except for the algorithms mentioned above, other algorithms also have some missed detections. OTA and PAA have the worst performance in false detections, and BorderDet and YOLOF have less false detections than OTA and PAA. Experimental results indicate that all the mentioned methods will get worse results in near shore, especially when the ships are arranged densely. Among all the algorithms, our MSSDNet has the best performance with no false detections and less missed detections, which demonstrates that the overall performance of MSSDNet is better than the other methods.

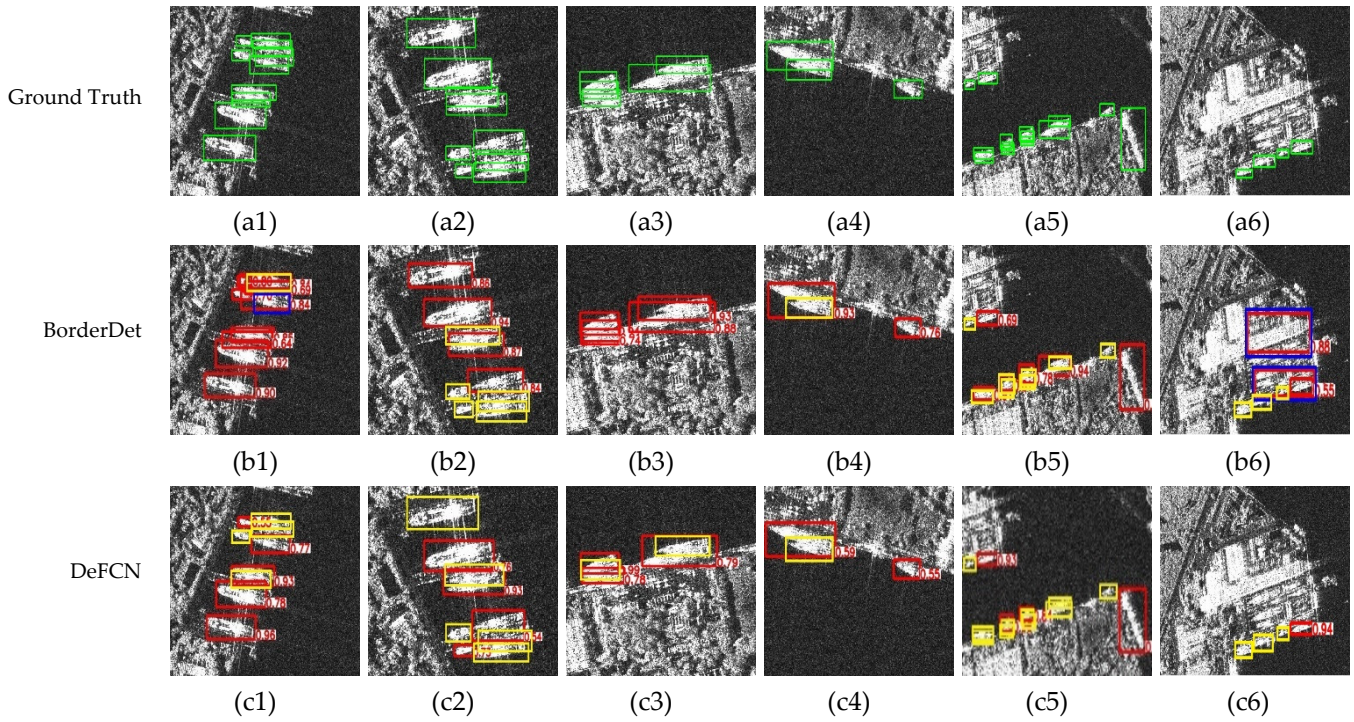


Figure 10. *Cont.*

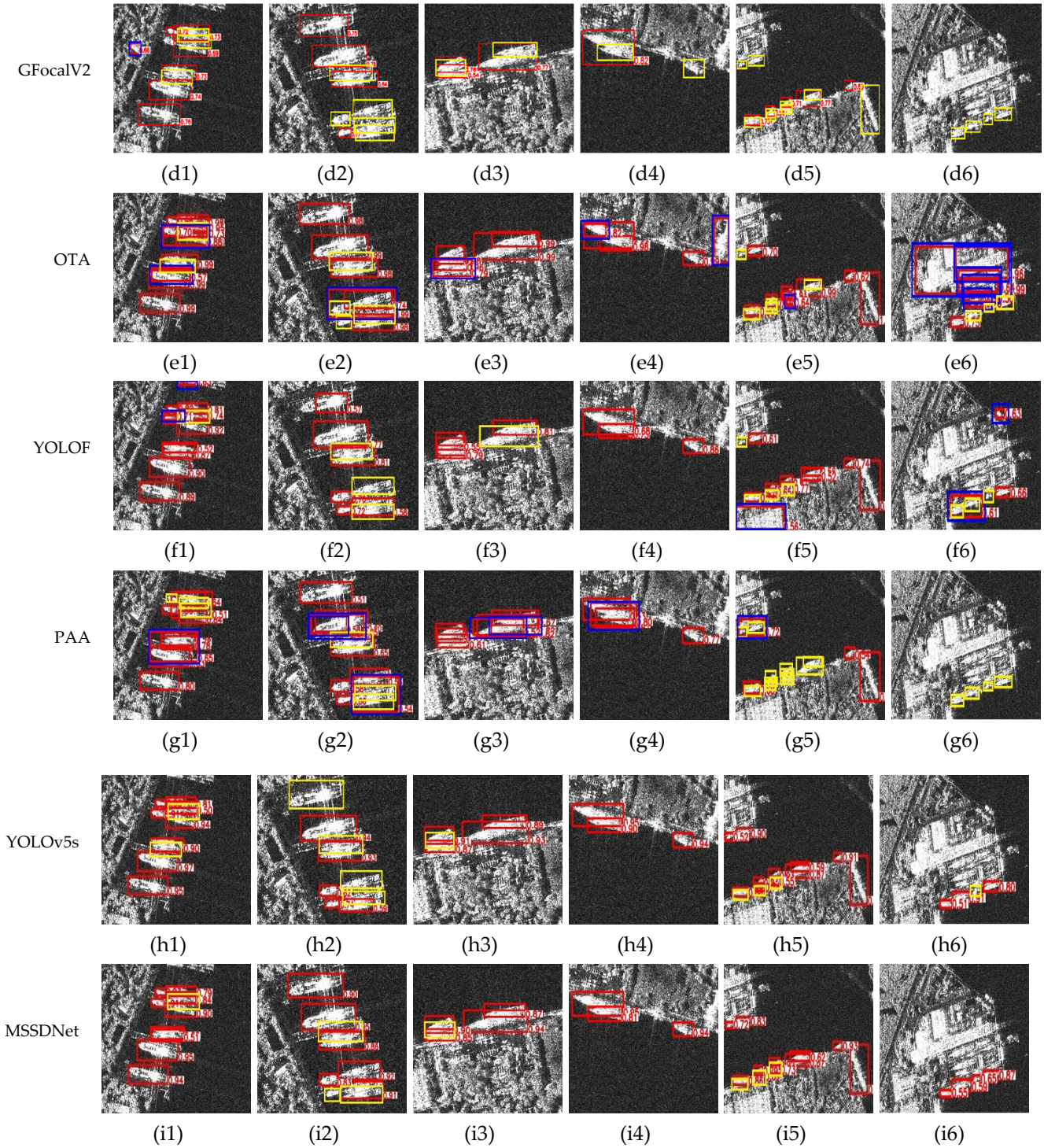


Figure 10. Experimental results in the SSDD dataset. (**a1**–**a6**): ground truth images; from (**b1**–**b6**) to (**i1**–**i6**): predicted results of BorderDet, DeFCN, GFocalV2, OTA, YOLOF, PAA, YOLOv5s, and MSSDNet, respectively. The green boxes are the ground truths, the red boxes are the predicted boxes, the yellow boxes represent missed detections, and the blue boxes represent the false detections.

3.4. Experiments on SARShip

The results of the proposed method on SARSship are shown in Table 4. The table shows that MSSDNet exceeds the YOLOv5s baseline in all COCO metrics. Especially the values of AP_S, AP_M, and AP_L prove that MSSDNet can improve the detection performance of ships with different scales at the same time. The AP_{75} metric is higher than the YOLOv5s

baseline, which indicates that MSSDNet can get more precise ships location information. The AP value of MSSDNet has an improvement of 1.5%, which shows the excellent overall performance compared with the YOLOv5s baseline.

Table 4. The experimental result of MSSDNet on SARShip.

Methods	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
YOLOv5s	58.6	94.6	65.4	52.8	65.6	59.4
MSSDNet	60.1	95.1	68.2	54.6	66.6	62.2

Table 5 compares the detection performance of MSSDNet and other methods on the SARShip dataset. In the AP metric, MSSDNet has the best result of 60.1%, surpassing all the other methods in the table, which indicates that MSSDNet has better overall performance. In both AP_M and AP_L metrics, MSSDNet does not achieve the satisfying results compared with OTA, but still surpasses most other methods. The values of the AP_{75} and AP_S metrics have shown that MSSDNet still can get more precise ships location information and have great detection performance of small ships compared with other methods. YOLOF can still work well on large ships but has worse results in other metrics. OTA has outstanding performance in AP_M and AP_L, but in AP_{75} and AP_S metrics, it has poorer detection performance than MSSDNet. It can be seen that MSSDNet has a relatively balanced performance on the COCO metrics compared with other methods, which can effectively focus on multiple metrics instead of focusing on a specific metric, e.g., OTA and YOLOF focus on AP_L metrics so that other metrics have poor performance.

Table 5. The results of different methods on SARShip.

Methods	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
BorderDet	56.7	93.8	62.3	49.6	65.3	57.3
DeFCN	54.5	93.5	58.1	49.8	61.0	47.8
GFocalV2	59.3	94.7	67.0	52.3	67.8	57.0
OTA	59.3	94.7	65.7	51.5	68.4	73.9
YOLOF	53.3	93.9	55.0	46.3	62.0	73.1
PAA	44.4	88.3	38.6	38.9	51.1	30.6
YOLOv5s	58.6	94.6	65.4	52.8	65.6	59.4
MSSDNet	60.1	95.1	68.2	54.6	66.6	62.2

Figure 11 presents the experimental results of other latest object detection methods on the SARShip dataset. The green boxes indicate the ground truths, the red boxes represent the predicted results, the yellow boxes represent missed detections, and the blue boxes represent false detections. In order to make the displayed detection results more representative, we selected six detection images with complex backgrounds. From the results of Figure 11, it is difficult to recognize the ships in near shore; DeFCN, YOLOF, and PAA have the worst performance in missed detections, while OTA, PAA, and YOLOv5s have more false detections. Although our MSSDNet has two missed detections and four false detections, the overall performance of MSSDNet is still much better than the compared methods, which has demonstrated the superiority of MSSDNet.

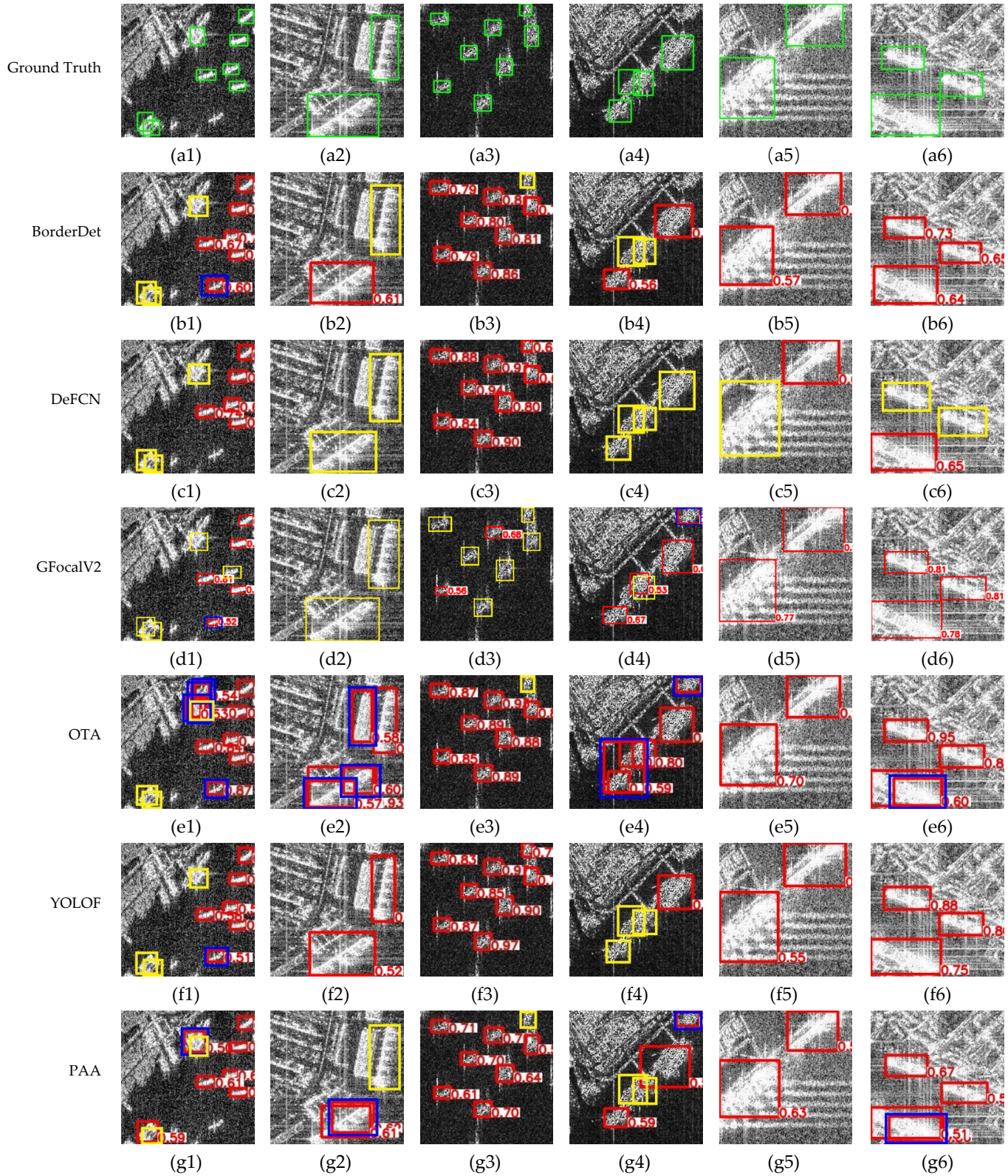


Figure 11. *Cont.*

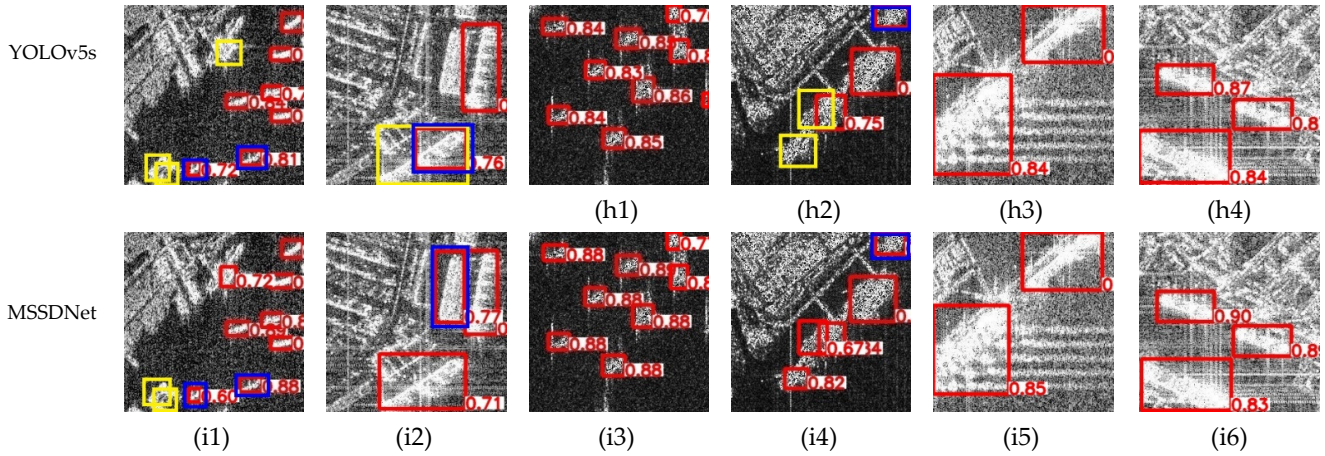


Figure 11. Experimental results in SARShip dataset. (**a1**–**a6**): ground truth images; from (**b1**–**b6**) to (**i1**–**i6**): predicted results of BorderDet, DeFCN, GFocalV2, OTA, YOLOF, PAA, YOLOv5s, and MSSDNet, respectively. The green boxes are the ground truths, the red boxes are the predicted boxes, the yellow boxes represent missed detections, and the blue boxes represent the false detections.

3.5. Ablation Experiments

In the ablation experiments, we adopt YOLOv5s as the baseline, and the SSDD dataset is used to verify the performance of the MSSDNet. We use the CSPMRes2 module and FC-FPN for the ablation study. Table 6 shows the results of the ablation study. The result shows that although the detection performance of MSSDNet with the CSPMRes2 module has been reduced in the detection of medium-size ships, it improves the detection performance of large ships compared with YOLOv5s. The result of MSSDNet with FC-FPN shows that the FC-FPN module can improve the detection performance of large ships in case of almost no loss in other COCO metrics compared to YOLOv5s. We can see that MSSDNet with only the FC-FPN module and MSSDNet with only the CSPMRes2 module have no improvement in small ships detection.

Table 6. Ablation experiments on SSDD.

Methods	CSPMRes2	FC-FPN	AP (%)	AP_{50} (%)	AP_{75} (%)	AP_S (%)	AP_M (%)	AP_L (%)
YOLOv5s			60.2	95.4	69.3	54.1	69.0	69.0
MSSDNet	√		60.2	95.5	68.1	54.5	68.6	70.6
		√	60.7	96.8	68.0	54.1	70.5	70.0
	√	√	61.1	95.6	70.9	55.4	70.0	70.4

In the paper, CSPMRes2 is a module with multi-scale feature extraction capability, which is one of the basic modules in the backbone of MSSDNet and can increase the receptive field of feature maps. FC-FPN is a module with fusion coefficients, which is the detection head of MSSDNet and can focus on ships with different sizes at the same time. We can see from Table 6 that using CSPMRes2 alone has a significant improvement in AP_L compared with YOLOv5s, which is because CSPMRes2 can increase the receptive field of feature maps, and that is beneficial to the detection of a large ship. Table 6 also shows that using FC-FPN alone can improve the detection of medium and large ships compared with YOLOv5s, which is because FC-FPN with fusion coefficients can balance the detections of different size ships. Whether using CSPMRes2 alone or FC-FPN alone, the AP_{75} and AP_S have no improvements compared with YOLOv5s. However, the combination of CSPMRes2 and FC-FPN have an improvement of 1.6% in AP_{75} and 1.3% AP_S, respectively. The above phenomenon shows that only the FC-FPN detection head of MSSDNet can effectively use the features generated by CSPMRes2, and the combination of the detection head of YOLOv5s and CSPMRes2 in the backbone only improves the detection of large ships. Since

FC-FPN has the capability to balance the detection of different size ships, when improving the detection of small ships, the detection of medium and large ships will be balanced. That is the main reason that the AP_{50}, AP_{L}, and AP_{M} of MSSDNet with CSPMRes2 and FC-FPN are lower than MSSDNet with a single module. MSSDet with CSPMRes2 and FC-FPN can balance overall performance instead of improving significant performance in certain metrics. The result of MSSDNet with the CSPMRes2 module and FC-FPN module shows that the combination of the CSPMRes2 module and FC-FPN respectively improve the AP_{75} by 1.6%, AP_{S} by 1.3%, APM by 1.0%, and AP_{L} by 1.4% compared with the YOLOv5s baseline. The ablation experimental results fully prove that the combination of the CSPMRes2 module and FC-FPN module can improve the detection performance of ships in SAR images, and get more accurate ships location information.

In order to explore how the different fusion coefficient ranges affect the detection performance of MSSDNet, we set a series range of values for fusion coefficients and conduct the experiments on SSDD. Figure 12 shows the experimental results, the value of horizontal coordinate represents the upper bound of the fusion coefficient range, e.g., the value 2 of horizontal coordinate represents that the range of fusion coefficient is between 0 and 2. We can see that MSSDNet can get the best detection performance when the range of fusion coefficient is limited to 0 and 2, and detection performance will degrade while the upper bound of the range is lower or higher than 2.

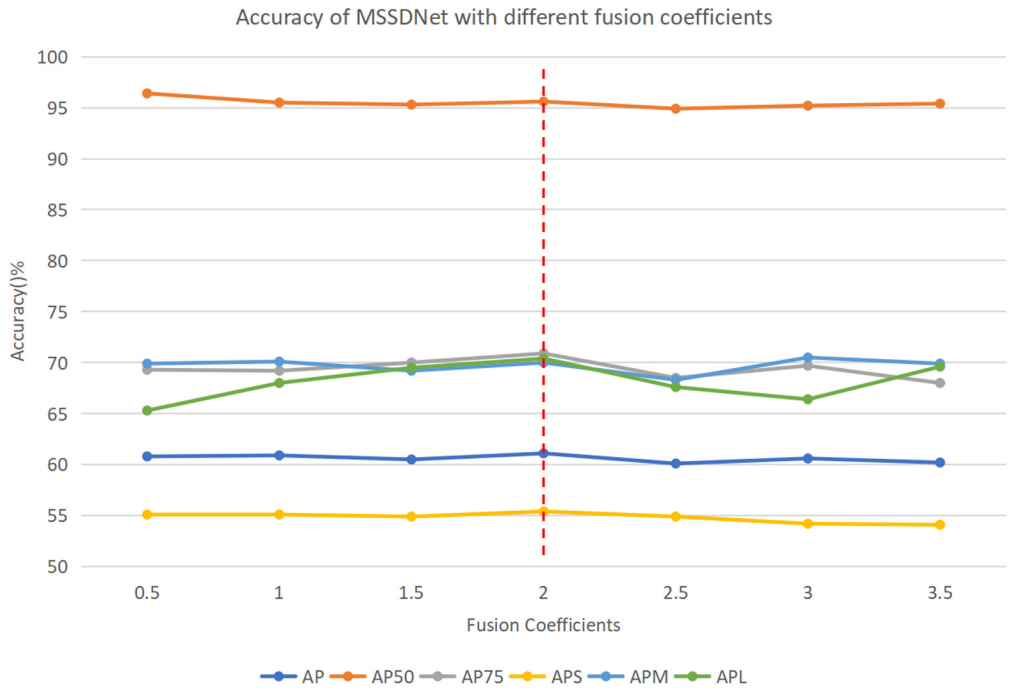


Figure 12. Accuracy of MSSDNet with different fusion coefficient ranges.

3.6. Comparison of Networks Model Size

We also compare the parameters of MSSDNet with the other methods. It can be seen from Table 7 that although MSSDNet increases 11.4MB parameters on the basis of YOLOv5s baseline, MSSDNet still has a great advantage of parameters compared with other methods, which means that MSSDNet has fewer parameters while having good performance, which is extremely competitive with other methods.

Table 7. Comparison of networks performance.

Methods	AP (%)		Inference Time (Milliseconds/Image)		Model Size (MB)
	SARShip	SSDD	SARShip	SSDD	
BorderDet	56.7	57.5	43.1	38.7	264.1
DeFCN	54.5	55.5	31.9	29.7	260.9
GFocalV2	59.3	56.2	61.0	62.9	427.3
OTA	59.3	59.1	32.8	29.0	256.2
YOLOF	53.3	59.2	43.5	76.4	368.5
PAA	44.4	56.0	46.0	87.9	1063.2
YOLOv5s	58.6	60.2	1.6	22.3	14.4
MSSDNet	60.1	61.1	3.1	24.2	25.8

3.7. Comparison of Inference Time

We compare the inference time of MSSDNet with the other methods. In the test of inference time, the input size of all methods is 512 × 512 on the SSDD dataset, and 256 × 256 on the SARShip dataset. Table 7 shows that the inference time of MSSDNet is only lower than the YOLOv5s baseline. This is because MSSDNet increases some computations on the basis of the YOLOv5s baseline for better accuracy. In addition, CSPMRe2Net is 10.3 times faster than the third inference time of DeFCN on the SARShip dataset and 1.2 times faster than the third inference time of OTA on the SSDD dataset. MSSDNet has shown the advantage of inference time compared with other methods.

4. Discussion

The experimental results on SSDD and SARShip datasets have demonstrated the superiority of MSSDNet, and the ablation study of CSPMRes2 and FC-FPN modules proves that the combination of adaptive feature fusion and multi-scale feature extraction can make a significant improvement on ship detection performance. However, what needs to be noticed is that the learning range of fusion coefficients has an excessive dependence on the training data. In addition, from the results of Figures 10 and 11, we can see that increase in background complexity and ships density have impacts on detection results. In the future, we will consider eliminating the data reliance for fusion coefficients and solving background complexity problems by constructing a ship detection network with stronger characterization capability. On the other hand, the SAR images are taken from a bird's-eye view, which means that there is no overlap between ships, thus we will make attempts on detection methods with oriented annotations.

5. Conclusions

In this paper, the MSSDNet is proposed to detect ships of different sizes in SAR images. The CSPMRes2 module and FC-FPN module are the vital components of MSSDNet, where the CSPMRes2 module is responsible for improving the feature extraction capability of the network, and the FC-FPN module in MSSDNet balances the detection of ships with multi-scale features in SAR images. The ablation study in this paper has confirmed the effectiveness of the two modules; MSSDNet based on the CSPMRes2 module and FC-FPN module can improve the precision of multi-scale ships detection. In addition, it can generate more precise predicted boxes. According to the experimental results on SSDD and SARShip datasets, MSSDNet has achieved higher overall detection performance than other methods. Because the CSPMes2 module just increases a few parameters, the MSSDNet based on the CSPMRes2 module does not increase too many parameters. Benefitting from the small amount of network parameters, in the comparisons of network model size and inference time, both the model size and inference time are lower than other methods, which is of great importance for the field of aviation, aerospace, and the military.

Author Contributions: K.Z., H.W. and M.Z. provided the ideas; K.Z. and J.T. implemented this algorithm with PyTorch; K.Z. wrote this paper; M.Z. and H.W. revised this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No.12003018), the Fundamental Research Funds for the Central Universities (No. XJS191305), and the China Postdoctoral Science Foundation (No. 2018M633471).

Data Availability Statement: The public datasets are used in this study, no new data are created or analyzed. Data sharing is not applicable to this article.

Acknowledgments: Thanks to the authors of SARShip and SSDD for providing the SAR image dataset and the authors of YOLOv5 for contributing an excellent object detection algorithm.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, D.; Liang, Q.; Liu, H.; Liu, Q.; Liu, H.; Liao, G. A Novel Multidimensional Domain Deep Learning Network for SAR Ship Detection. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–13. [CrossRef]
2. Wu, Z.; Hou, B.; Jiao, L. Multiscale CNN with Autoencoder Regularization Joint Contextual Attention Network for SAR Image Classification. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1200–1213. [CrossRef]
3. Xie, E.; Ding, J.; Wang, W.; Zhan, X.; Xu, H.; Sun, P.; Li, Z.; Luo, P. Detco: Unsupervised Contrastive Learning for Object Detection. In Proceedings of the Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Online, 11–17 October 2021; pp. 8392–8401.
4. Qian, X.; Lin, S.; Cheng, G.; Yao, X.; Ren, H.; Wang, W. Object Detection in Remote Sensing Images Based on Improved Bounding Box Regression and Multi-Level Features Fusion. *Remote Sens.* **2020**, *12*, 143. [CrossRef]
5. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C. Sparse R-Cnn: End-to-End Object Detection with Learnable Proposals. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 14454–14463.
6. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
7. Girshick, R.B. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
8. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
9. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
10. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
11. Redmon, J.; Farhadi, A. YOLO9000: Better faster stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
13. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.; Liao, H. YOLOv4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. Cheng, G.; Si, Y.; Hong, H.; Yao, X.; Guo, L. Cross-scale feature fusion for object detection in optical remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 431–435. [CrossRef]
16. Huang, Z.; Pan, Z.; Lei, B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sens.* **2017**, *9*, 907. [CrossRef]
17. Yu, J.; Zhou, G.; Zhou, S.; Qin, M. A Fast and Lightweight Detection Network for Multi-Scale SAR Ship Detection under Complex Backgrounds. *Remote Sens.* **2022**, *14*, 31. [CrossRef]
18. Cheng, G.; Lang, C.; Wu, M.; Xie, X.; Yao, X.; Han, J. Feature enhancement network for object detection in optical remote sensing images. *J. Remote Sens.* **2021**, *2021*, 9805389. [CrossRef]
19. Kang, M.; Leng, X.; Lin, Z.; Ji, K. A Modified Faster R-CNN Based on CFAR Algorithm for SAR Ship Detection. In Proceedings of the 2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP), Shanghai, China, 18–21 May 2017.
20. Li, J.; Qu, C.; Shao, J. Ship detection in SAR images based on an improved faster R-CNN. In Proceedings of the SAR in Big Data Era (BIGSARDATA), Beijing, China, 13–14 November 2017.
21. Jiao, J.; Zhang, Y.; Sun, H.; Yang, X.; Gao, X.; Hong, W.; Fu, K.; Sun, X. A Densely Connected End-to-End Neural Network for Multiscale and Multiscene SAR Ship Detection. *IEEE Access* **2018**, *6*, 20881–20892. [CrossRef]
22. Kang, M.; Ji, K.; Leng, X.; Lin, Z. Contextual Region-Based Convolutional Neural Network with Multilayer Fusion for SAR Ship Detection. *Remote Sens.* **2017**, *9*, 860. [CrossRef]

23. Liu, N.; Cao, Z.; Cui, Z.; Pi, Y.; Dang, S. Multi-Scale Proposal Generation for Ship Detection in SAR Images. *Remote Sens.* **2019**, *11*, 526. [CrossRef]
24. Fu, J.; Sun, X.; Wang, Z.; Fu, K. An Anchor-Free Method Based on Feature Balancing and Refinement Network for Multiscale Ship Detection in SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *59*, 1331–1344. [CrossRef]
25. Cui, Z.; Li, Q.; Cao, Z.; Liu, N. Dense Attention Pyramid Networks for Multi-Scale Ship Detection in SAR Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8983–8997. [CrossRef]
26. Gao, F.; He, Y.; Wang, J.; Hussain, A.; Zhou, H. Anchor-free Convolutional Network with Dense Attention Feature Aggregation for Ship Detection in SAR Images. *Remote Sens.* **2020**, *12*, 2619. [CrossRef]
27. Guo, H.; Yang, X.; Wang, N.; Gao, X. A CenterNet++ model for ship detection in SAR images. *Pattern Recognit.* **2021**, *112*, 107787. [CrossRef]
28. Wu, Z.; Hou, B.; Ren, B.; Ren, Z.; Wang, S.; Jiao, L. A Deep Detection Network Based on Interaction of Instance Segmentation and Object Detection for SAR Images. *Remote Sens.* **2021**, *13*, 2582. [CrossRef]
29. Yu, L.; Wu, H.; Zhong, Z.; Zheng, L.; Deng, Q.; Hu, H. TWC-Net: A SAR Ship Detection Using Two-Way Convolution and Multiscale Feature Mapping. *Remote Sens.* **2021**, *13*, 2558. [CrossRef]
30. Zhang, T.; Zhang, X.; Ke, X. Quad-FPN: A Novel Quad Feature Pyramid Network for SAR Ship Detection. *Remote Sens.* **2021**, *13*, 2771. [CrossRef]
31. Wang, Y.; Wang, C.; Zhang, H.; Dong, Y.; Wei, S. A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote Sens.* **2019**, *11*, 765. [CrossRef]
32. Wang, C.-Y.; Liao, H.-Y.M.; Wu, Y.-H.; Chen, P.-Y.; Hsieh, J.-W.; Yeh, I.-H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) workshops, Online, 14–19 June 2020; pp. 390–391.
33. Hou, Q.; Zhou, D.; Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 13713–13722.
34. Gao, S.; Cheng, M.-M.; Zhao, K.; Zhang, X.-Y.; Yang, M.-H.; Torr, P.H. Res2net: A New Multi-Scale Backbone Architecture. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 652–662. [CrossRef] [PubMed]
35. Qiu, H.; Ma, Y.; Li, Z.; Liu, S.; Sun, J. Borderdet: Border Feature for Dense Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 549–564.
36. Wang, J.; Song, L.; Li, Z.; Sun, H.; Sun, J.; Zheng, N. End-to-End Object Detection with Fully Convolutional Network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 15849–15858.
37. Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss v2: Learning Reliable Localization Quality Estimation for Dense Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 11632–11641.
38. Ge, Z.; Liu, S.; Li, Z.; Yoshie, O.; Sun, J. OTA: Optimal Transport Assignment for Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 303–312.
39. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You Only Look One-Level Feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 13039–13048.
40. Kim, K.; Lee, H.S. Probabilistic Anchor Assignment with Iou Prediction for Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 355–371.
41. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
42. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft Coco: Common Objects in Context. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 6–12 September 2014; pp. 740–755.