Journal of *Marine Science and Engineering*

MDPI

Article

Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV

Tao Liu [1], Bo Pang [1], Lei Zhang [2,*], Wei Yang [3] and Xiaoqiang Sun [1]

[1] College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; liutao@hrbeu.edu.cn (T.L.); pangbo@hrbeu.edu.cn (B.P.); sunxiaoqiang@hrbeu.edu.cn (X.S.)
[2] College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
[3] 708th Research Institute of China Shipbuilding Industry, Shanghai 200000, China; zcczx2018@hrbeu.edu.cn
* Correspondence: zhanglei103@hrbeu.edu.cn; Tel.: +86-13895793667

Abstract: Unmanned surface vehicles (USVs) have been extensively used in various dangerous maritime tasks. Vision-based sea surface object detection algorithms can improve the environment perception abilities of USVs. In recent years, the object detection algorithms based on neural networks have greatly enhanced the accuracy and speed of object detection. However, the balance between speed and accuracy is a difficulty in the application of object detection algorithms for USVs. Most of the existing object detection algorithms have limited performance when they are applied in the object detection technology for USVs. Therefore, a sea surface object detection algorithm based on You Only Look Once v4 (YOLO v4) was proposed. Reverse Depthwise Separable Convolution (RDSC) was developed and applied to the backbone network and feature fusion network of YOLO v4. The number of weights of the improved YOLO v4 is reduced by more than 40% compared with the original number. A large number of ablation experiments were conducted on the improved YOLO v4 in the sea ship dataset SeaShips and a buoy dataset SeaBuoys. The experimental results showed that the detection speed of the improved YOLO v4 increased by more than 20%, and mAP increased by 1.78% and 0.95%, respectively, in the two datasets. The improved YOLO v4 effectively improved the speed and accuracy in the sea surface object detection task. The improved YOLO v4 algorithm fused with RDSC has a smaller network size and better real-time performance. It can be easily applied in the hardware platforms with weak computing power and has shown great application potential in the sea surface object detection.

Keywords: YOLO v4; sea surface object detection; reverse depthwise separable convolution; unmanned surface vehicles

Citation: Liu, T.; Pang, B.; Zhang, L.; Yang, W.; Sun, X. Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV. *J. Mar. Sci. Eng.* **2021**, *9*, 753. https://doi.org/10.3390/jmse9070753

Academic Editor: Alessandro Ridolfi

Received: 1 June 2021
Accepted: 5 July 2021
Published: 7 July 2021

1. Introduction

In recent years, marine economy has played an increasingly important role in the economic development of countries. With the increasing human maritime activities, unmanned surface vehicles (USVs) have been widely used in military and civil activities. USV is a kind of unmanned surface craft that can replace humans to perform many dangerous tasks, such as reconnaissance, search, and navigation. It is equipped with various sensors such as photoelectric system, navigation radar, and lidar. It performs environmental perception through the optical vision images collected by the photoelectric camera. The task of object detection is the key to realizing the perception of the sea surface environment, and it is also the key technology to enhance the intelligent level of USV [1]. However, the computing power of the unmanned system hardware platform is generally not high, thus, it is of great significance to develop an object detection algorithm with both speed and accuracy that can be applied in the task of USV for detecting sea surface objects.

For the field of image recognition, traditional algorithms still have a good effect, using the characteristics of thermal imaging pictures to judge the working status of the angle grinder [2]. Before deep learning and object detection algorithms were combined, the object detection requested manually designed features [3]. The failure to achieve end-to-end detection leads to low detection speed. Besides, since the manually designed features have no strong robustness for the diverse changes of input, it is difficult to apply them to the complex sea environment perception task of USV. In recent years, with the rapid development of computer hardware such as central processing unit (CPU) and graphics processing unit (GPU), the computing power of computers has been greatly enhanced. Moreover, deep learning software frameworks such as TensorFlow have made neural network-based object detection algorithms gradually prevail and are widely used in various practical engineering projects. Compared with traditional object detection algorithms, the object detection algorithms based on neural networks are still developing rapidly owing to their faster detection and more accurate detection results.

The development of neural network-based object detection algorithms is mainly stimulated by the rapid growth of convolutional neural networks (CNN). Weight sharing and local receptive field are the two important features of CNN, which not only reduce the complexity of the network model, but also have scale invariance to input translation, rotation, and scaling, thus contributing to the extensive application of CNN in the field of computer vision. CNN is the backbone network of modern object detection algorithms and is responsible for feature extraction. Its powerful feature extraction ability is the key to modern object detection algorithms. Since AlexNet [4] won the championship in the ImageNet image classification task competition, CNN began to develop rapidly in the field of computer vision. Simonyan et al. [5] proposed the concept of block and constructed a neural network model by reusing basic blocks. Each basic block contains a CNN layer, pooling layer and an activation function. The CNN layer does not change the size of the input, and the pooling layer performs downsampling on the input. Such structure provides guidance for subsequent related work. Lin et al. [6] developed convolution with a kernel size of 1, which can enhance the nonlinearity of the CNN model, reduce the number of weights of the network model, and can be easily integrated with various CNN models. Therefore, it has been widely used in the mainstream CNN models. The concept of branch was proposed in the GoogLeNet series [7,8]. Convolutions with kernels of different size were used to construct sparse connection, which not only reduces the number of weights, but also improves accuracy. The ResNet proposed by He et al. [9] makes the CNN model deeper without affecting the accuracy through a special structure, and it is one of the best CNN structures. Xie et al. [10] proposed a highly modular ResNeXt model, which decreases the difficulty of manually designing the network structure. DenseNet and CSPNet fuse the information of the shallow feature maps and the deep feature maps in the CNN model, so that the information of the shallow feature maps can be fully used [11,12].

With the rapid improvement of CNN's detection abilities, neural network-based object detection algorithms have also been rapidly developed. The mainstream neural network-based object detection algorithms are mainly one-stage or two-stage. The two-stage series algorithms [13–16] divide the process of object detection into two steps. The first step is to generate some candidate frames containing the objects to be detected, and then to determine the position and size of the object. The second step is to classify the candidate frames and determine the category of the objects. Since the position regression and classification of the objects are divided into two steps, the speed of two-stage algorithms is smaller than that of one-stage series algorithms, but the accuracy is higher. Correspondingly, the one-stage series algorithms [17–21] can directly output the bounding box coordinates and categories of the object, which are more advantageous in terms of speed. Therefore, the one-stage series algorithms have been widely applied in actual engineering projects. However, the process of pursuing accuracy often brings the risk of reducing the detection speed. In addition to a good backbone network, many components have been applied in the object detection algorithms based on neural networks. The pyramid-shaped feature fusion

structure was first proposed in [22]. This structure combines the information of feature maps of different scales and levels to enhance the performance of the object detection network model. Subsequently, the feature fusion network structure was further optimized in [23–26] and becomes more complicated. In addition to the feature fusion structure, the attention mechanism has also been increasingly applied to the field of object detection. In [27,28], the attention mechanism was proposed and applied to the channel dimension of the feature map. The network autonomously learns weights and allocates them to each channel of the feature map. Such attention mechanism can be easily embedded in different CNN architectures. Woo et al. [29] applied the attention mechanism to the channel and spatial dimensions of the feature map. Although these components improve accuracy, they also increase the amount of calculation.

Neural network-based object detection algorithms have been widely used in sea surface object detection task. Many related research focuses on adding different components to industry-leading object detection algorithms, or increasing the volume of the network structure to achieve good detection results, improve accuracy or speed. Since the Squeeze-and-Excitation (SE) module can be easily added to the CNN structure, it was added to the backbone network of the object detection algorithms [30–32] to improve the network model's ability to detect ships on the sea. However, adding the SE module will cause a decrease in the real-time performance of the network. Some studies improved the two-stage series algorithms and applied them to a sea surface object detection task [33–36]. Although good results have been achieved in accuracy, it is difficult to ensure satisfying real-time performance, which is limited by the detection method of the two-stage series networks. Li et al. [37] improved the backbone network of YOLO v3, and used the DenseNet structure to enhance the stability of object detection. Jie et al. [38] modified the network classifier based on YOLO v3, and introduced the Soft-NMS algorithm. Wang et al. [39] proposed Path Argumentation Fusion Network to fuse the information of feature maps multiple times to increase the ability of YOLO v3 for the detection of small objects. Liu et al. [40] developed different methods of anchor setting to improve the ability of YOLO v3 to detect sea surface objects without increasing the computational burden.

However, USVs have high requirements for the accuracy and speed of the object detection algorithm. How to improve the speed and accuracy of the object detection algorithm simultaneously is a major difficulty. In this paper, YOLO v4 was improved and applied to the sea surface object detection task of USV. YOLO v4 is currently one of the optimal object detection algorithms regarding the balance between speed and accuracy, thus it is applicable to the sea surface object detection task. In the backbone network and feature fusion network of YOLO v4, there are a large number of convolutional blocks stacked up by convolutions with a kernel size of 1 and 3. Inspired by Depthwise Separable Convolution (DSC) [41], we made some improvements to DSC and proposed RDSC. RDSC improved the backbone network and feature fusion network of the original YOLO v4. Not too many components are added to YOLO v4 or expand the network volume but the complexity of the network model is reduced while ensuring accuracy, which is of great significance for the sea surface object detection of USVs with weak performance of the computing hardware platform.

The main contributions of this paper are as follows: (1) RDSC was applied to the backbone network and feature fusion network of YOLO v4, which greatly improves the detection speed of YOLO v4, and enhances accuracy at the same time. Experimental verification was performed on the sea surface ship dataset and sea surface buoy dataset. (2) Comparative experiments were conducted on DSC and RDSC, and the influence of DSC and RDSC on YOLO v4 was analyzed. Besides, the weight distribution of the improved network model was studied. Our proposed method reduces the number of weights on the one hand and improves the utilization of weights on the other hand.

2. The Backbone Network and Feature Fusion Network of YOLO v4

2.1. Backbone Network

The backbone network of YOLO v4 [20] fuses the Cross-Stage-Partial-connections (CSP) structure [11] on the basis of Darknet53 [19], which is called CSPDarknet53. In the structure of CSPDarknet53, each convolution block splits the input feature map into two parts, A and B. Part A is retained, and part B passes through the residual network unit (ResUnit), that is, the repeated residual network, and finally the part A and the convolution results of part B are connected in the channel dimension. Splitting is the key to reducing the number of weights. Moreover, since the shallow features are reused, the CSP structure enhances the detection ability of the CNN framework. In CSPDarknet53, there are three feature maps of different sizes as the input of feature fusion. After the Spatial Pyramid Pooling (SPP) structure [42] is connected to the smallest feature map, the SPP structure can increase the receptive field and separate the most important contextual features.

Figure 1 shows the structure of the convolution block in CSPDarknet53. There are several such basic blocks in CSPDarknet53. First, after the input feature map is downsampled by Conv1, the length and width of the feature map are halved, and the number of channels is twice the original number. Then the downsampling result generates two branches through Conv2 and Conv3. It is worth noting that the number of channels in the feature maps of these two branches has become half of the original number. The number of channels in the feature map after Conv2 is halved, and then the feature map passes through the residual unit (ResUnit), which greatly reduces the number of weights. Finally, the feature maps of the two branches are connected in the channel dimension.

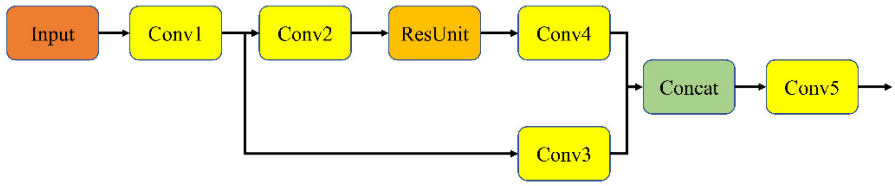


Figure 1. Structure of the convolution block in CSPDarknet5.

Figure 2 shows the SPP structure in CSPDarknet53. The input feature map passes through three maxpool layers with windows of different size, respectively, to generate three output feature maps. These three output feature maps and the input feature map are connected in the channel dimension to obtain the final result. Due to the small amount of calculation of the pooling layer, the SPP structure will not bring burden to the network's computing speed, instead, it can expand the receptive field, separate contextual features, and improve the model's detection ability.

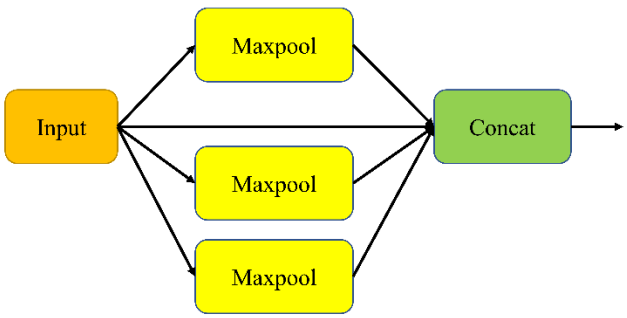


Figure 2. SPP structure connected to the smallest feature map in CSPDarknet53.

2.2. Feature Fusion Network

Path Aggregation Network (PANet) is a feature fusion network in YOLO v4. PANet has a similar structure with the Feature Pyramid Networks (FPN), and both use a feature

pyramid structure. However, in PANet, there are two different paths for feature fusion, that is, the top-down path and the bottom-up path. In the CNN architecture, low-level feature maps have more location information, and high-level feature maps have more semantic information. The top-down path propagates high-level semantic information to the lower level, and the bottom-up path spreads the location information to the higher level, so that the utilization of the information of bottom layer can be improved. The two paths fully fuse the information of different feature maps. In YOLO v4, there are three feature maps for final prediction. The structure of the PANet in YOLO v4 is shown in Figure 3.

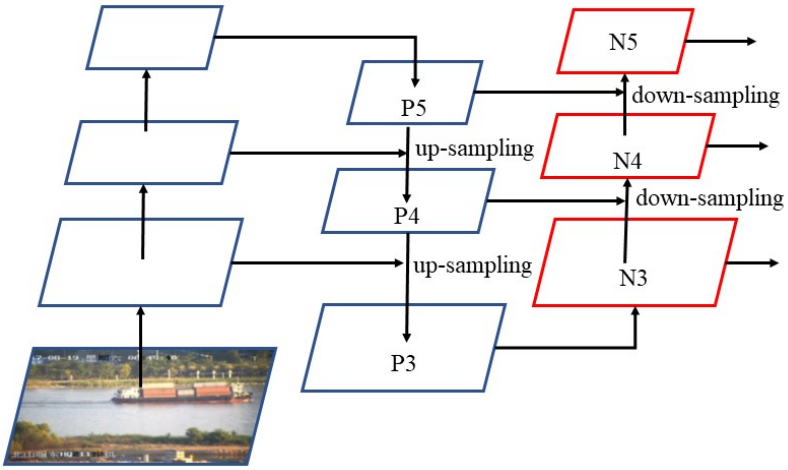


Figure 3. The structure of the feature fusion network in YOLO v4.

In Figure 3, P5 to P3 are the top-down paths, propagating high-level semantic information to low-level. N3 to N5 are bottom-up paths, propagating low-level location information to high-level.

3. Methods

3.1. RDSC in the Backbone Network

Convolution with a kernel size of 1 can enhance the nonlinearity of the network model and reduce the number of weights of the network model. It has been widely used in the framework of YOLO v4. In the ResUnit of the backbone network, each feature map must be subjected to the operation of 1×1 convolutions and 3×3 convolutions. In order to further improve the detection speed of the network model, based on the characteristics of the backbone network of YOLO v4, RDSC was applied to the backbone network of YOLO v4. RDSC is composed of 1×1 convolutional layer, Batch Normalization (BN) layer, 3×3 Depthwise Convolution layer and Mish activation function. Figure 4 shows the details of RDSC.

Figure 4 shows the structural differences between RDSC and DSC. RDSC is connected to 3×3 convolution after 1×1 convolution and uses the Mish activation function. The RDSC structure is similar to the ResUnit in YOLO v4, which is a stack of 1×1 convolution and 3×3 convolution, but the introduction of Depthwise Convolution significantly reduces the number of weights. This is the main reason why the improved YOLO v4 can achieve faster detection. Standard convolution kernel $K \in R^{m \times m \times n_1 \times n_2}$, Depthwise convolution kernel $DK \in R^{m \times m \times n_1}$, where m is the size of the convolution kernel, n_1 is the number of channels in the input feature map, and n_2 is the number of channels in the output feature map. The number of weights for the standard convolution kernel and Depthwise convolution kernel are respectively $c_1 = m \times m \times n_1 \times n_2$, $c_2 = m \times m \times n_1$.The number of weights of the standard convolution kernel is x times the number of weights of the Depthwise convolution kernel, $x = \frac{c_1}{c_2} = n_2$. Figure 5 shows the comparison between the original ResUnit and the improved ResUnit.

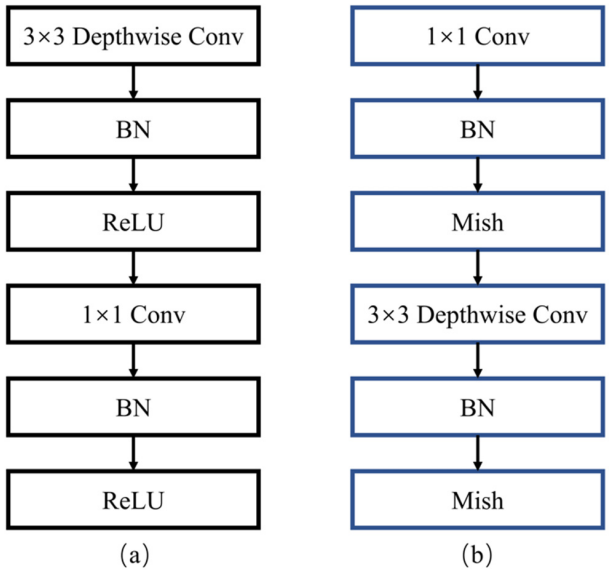


Figure 4. The structure of DSC (**a**); the structure of RDSC (**b**).

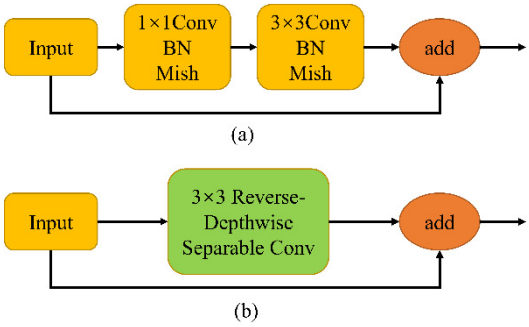


Figure 5. Structure of the original ResUnit (**a**); structure of the improved ResUnit (**b**).

It is worth noting that Figure 5a is the structure in the original YOLO v4, in which all convolutions are ordinary convolutions, and the 3×3 convolution in the RDSC module in Figure 5b is Depthwise Convolution. The ResUnit is the major structure of CSPDarknet53. A set of stacked 1×1 convolution and 3×3 convolution in the ResUnit was replaced with 1 RDSC. The use of RDSC greatly reduces the number of weights. If RDSC is decomposed, it is still a 1×1 convolution which is connected to a 3×3 convolution. From a certain perspective, this is a sparse version of CSPDarknet53.

3.2. RDSC in Feature Fusion Network

In the feature fusion network of YOLO v4, every two feature maps will undergo five convolution operations after upsampling (or downsampling) and connection in the channel dimension, and then the operations such as upsampling are repeated, so that the feature map will continue to fuse with other feature maps. These five convolution operations are composed of three 1×1 convolutions and two 3×3 convolutions, and each 1×1 convolution is connected to a 3×3 convolution. Under normal circumstances, the input of the feature fusion network comes from the end of the backbone network, the number of channels is relatively large, and the number of weights of convolution will also increase exponentially. In order to reduce the number of weights of the network, four of the five convolutions (two 1×1 convolutions, and two 3×3 convolution) were replaced with 2 RDSCs. Correspondingly, a more sparse feature fusion network was created. Figure 6 shows the details of the improved feature fusion network.

It is worth noting that Figure 6a is the feature fusion structure in the original YOLO v4, in which all convolutions are ordinary convolutions, and the convolution in the RDSC module in Figure 6b is Depthwise Convolution (dotted line section).

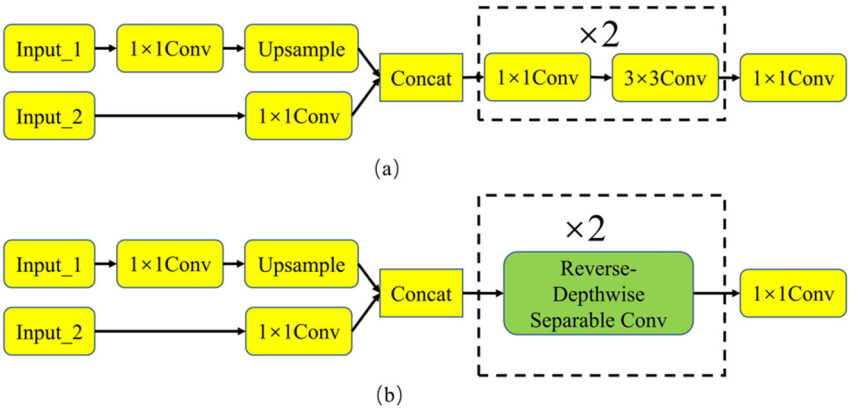


Figure 6. Feature fusion structure before improvement (**a**); feature fusion structure after improvement (**b**).

4. Results and Discussion

4.1. Introduction to USV and Datasets

Figure 7 shows the appearance of the USV developed by Harbin Engineering University. Figure 8 shows the high-precision photoelectric camera mounted on the USV. This camera has a low-light-level camera and an infrared thermal imager and is equipped with a servo system to achieve 360-degree pitch and rotation, so that full range of object tracking can be realized. The low-light-level camera outputs video through the Ethernet network, and performs control and status output through the serial interface. The images in the SeaBuoys dataset were collected by the camera of USV in the autonomous course.

Figure 7. The USV platform developed by Harbin Engineering University.

Figure 8. High-precision photoelectric camera.

Figure 9 shows the working process of the photoelectric camera. In the first step, the servo system controls the orientation of the camera to the target. In the second step, the photoelectric camera starts to collect video. In the third step, the computer runs the target detection algorithm to output the coordinates of the target. Finally, the servo system continues to track the target according to the output coordinates.

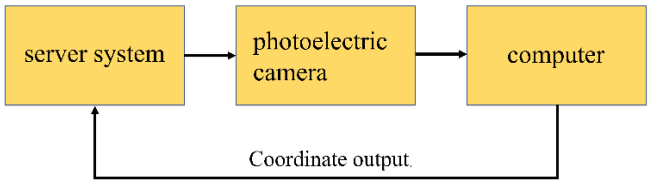
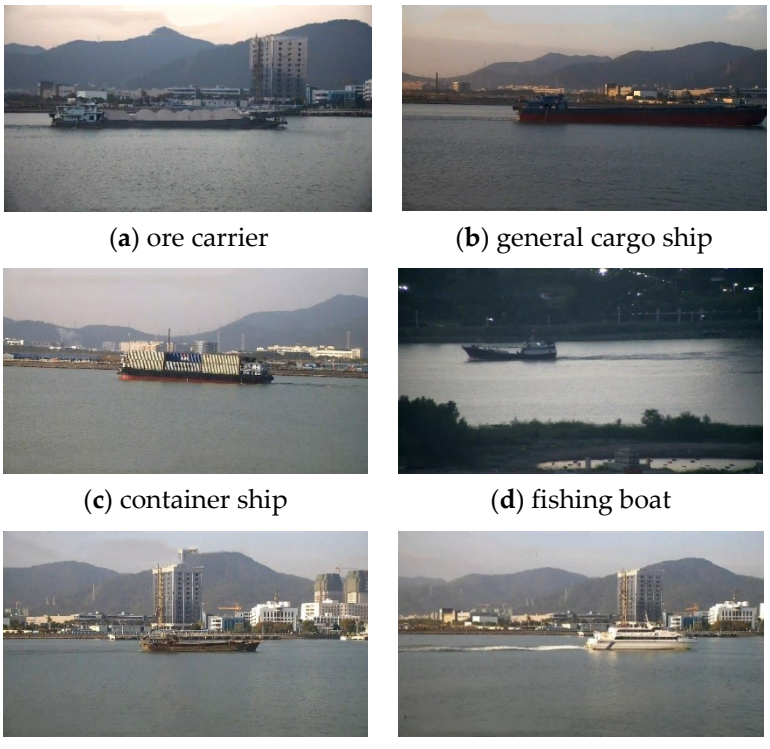


Figure 9. Working block diagram of photoelectric camera.

The SeaShips dataset [43] contains a total of 31,455 images of ships in RGB format (7000 published), involving 6 types of ships (ore carrier, bulk cargo carrier, general cargo ship, container ship, fishing boat, and passenger ship). All images are of the same size, 1920 × 1080, and were collected from the videos of real sea conditions. Figure 10 displays the appearance of different types of ships in the dataset.



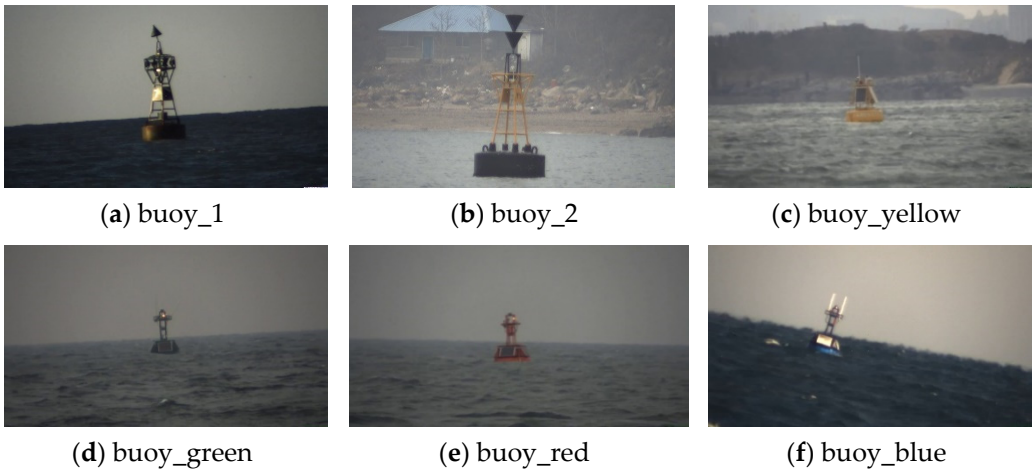
(**a**) ore carrier (**b**) general cargo ship

(**c**) container ship (**d**) fishing boat

(**e**) bulk cargo carrier (**f**) passenger ship

Figure 10. The appearance and corresponding labels of different types of ships in SeaShips.

Real engineering test was conducted on the USV vision hardware platform, and the SeaBuoys dataset was collected and produced. The dataset contains 6 types of buoys, and a total of 5751 images with a size of 1024 × 576 in RGB format. The appearance and the labels of the buoys are shown in Figure 11.



(**a**) buoy_1 (**b**) buoy_2 (**c**) buoy_yellow

(**d**) buoy_green (**e**) buoy_red (**f**) buoy_blue

Figure 11. The appearance and labels of different buoys in SeaBuoys.

4.2. Experimental Details

The software framework used in this experiment is TensorFlow 2.1.0-gpu version (Google, Santa Clara, CA, USA), which was run under the Linux system. In terms of hardware, the CPU is i7-9700 (Intel, Santa Clara, CA, USA), the GPU is a single NVIDIA GeForce RTX 2080TI (NVIDIA, Santa Clara, CA, USA).

The method for the optimization of network training is the gradient descent method, the weight decay coefficient is 0.0005, the impulse is 0.9, the size of the input image during training is 416 × 416, the size of the input image during test is 544 × 544, and the learning rate was set by two methods: cosine learning rate and warm-up [44]. The number of the rounds of warm-up was set to 4. In the first 4 rounds of training, the learning rate was increased by a small value to the initial learning rate of 0.0001, and then decreased to the final learning rate of 0.000001 according to the law of cosine learning rate. The total number of the rounds of training is 40. Figure 12 shows the changing trend of the learning rate during training.

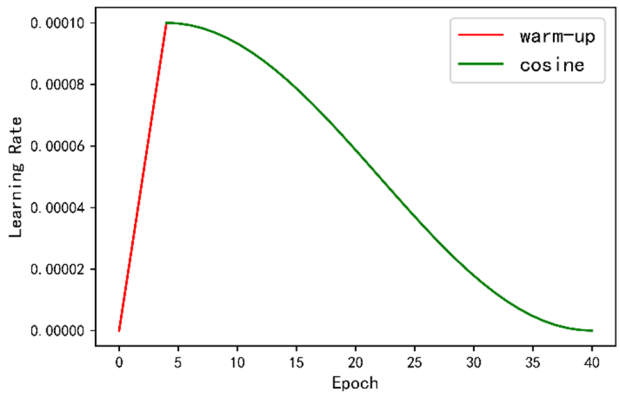


Figure 12. The variation curve of learning rate during training.

4.3. Ablation Experiment in SeaShips

In total, 90% of the SeaShips dataset was classified into the training set, and the remaining 10% was used as the test set. The original network structure of YOLO v4 and the solution proposed in this paper were compared in this section. In order to compare the experimental results of different schemes more comprehensively, on the basis of the scheme proposed in this paper, the RDSCs of the residual unit were all replaced with DSCs. In addition, every convolutional layer of YOLO v4 (including all 3 × 3 convolutions and 1 × 1 convolutions) was replaced with DSC or RDSC. Three indicators, mean Average Precision

(mAP), Frames Per Second (FPS), and the number of weights, were used for comparing the above-mentioned schemes. The experimental results are as follows.

Table 1 shows the experimental results of YOLO v4 in 4 different schemes and the original version of YOLO v4. The scheme proposed in this paper achieves the best results regarding both mAP and FPS, which are 94.58% and 68FPS, respectively. Compared with the performance of the original YOLO v4, mAP was increased by 1.78%, the detection speed increased by 13FPS, and the number of weights also reduced by more than 40%. Although the application of DSC in the ResUnit can also reduce the number of weights and improve the detection speed, the mAP is smaller than the value in the scheme proposed in this paper.

Table 1. Experimental results of different schemes under SeaShips.

Baseline	Method	mAP (%)	FPS	Number of Weights
YOLO v4	-	92.80	55	63,963,584
	Ours	94.58	68	35,524,224
	DSC in ResUnit	93.53	68	35,524,224
	DSC in all layers	87.41	58	14,814,724
	RDSC in all layers	90.00	53	14,814,724

In order to analyze the impact of RDSC on YOLO v4 more comprehensively, on the basis of the original YOLO v4, all convolutions were replaced with RDSCs, and another scheme was proposed for comparison, where all convolutions were replaced with DCSs. It can be seen from the experimental results that the mAP of the former is higher, but the FPS is smaller. Compared with the original version of YOLO v4, the two schemes all have reduced mAP. It is worth noting that these two schemes have only about 41% of the weights of the scheme proposed in this paper, but the detection speed is smaller. The reason is that the scheme proposed in this paper replaces the stack of 1×1 convolutions and 3×3 convolutions with RDSCs, which reduces the number of weights without changing the number of layers of the network, However, replacing all convolutions with DSCs or RDSCs will bring additional convolutional layers and BN layers, thus, although the number of weights is greatly reduced, the actual detection speed is decreased. Replacing all convolutional layers of YOLO v4 with RDSCs will lead to a decrease in mAP, by contrast, our method will increase mAP, indicating that for the same RDSC structure, different application methods also have a great impact on the detection performance of the model.

In order to further analyze the experimental results, the weight files of YOLO v4 and our scheme were analyzed, and the proportion of weights with values close to 0 to the total number of weights was calculated. The weights that meet the 3 indicators were selected, which are weights with values between -0.001 and 0.001, -0.005 and 0.005 and -0.01 and 0.01, and are expressed by $|w| < 0.001$, $|w| < 0.005$ and $|w| < 0.01$ in the following analysis. Figure 13 shows the statistical results.

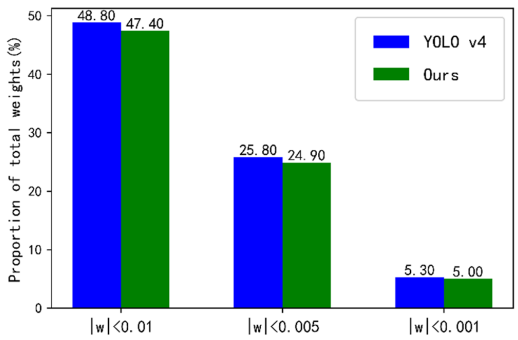


Figure 13. The statistical diagram of the weight distribution of YOLO v4 and our method in the test in SeaShips.

It can be seen from Figure 13 that the proportion of the weights with $|w| < 0.01$, $|w| < 0.005$ and $|w| < 0.001$ in our method to the total number of weights is reduced by 1.4%, 0.9% and 0.3%, respectively, compared with the results of YOLO v4. The small ratio of the weights with values close to 0 indicates that our framework has a higher ratio of weight utilization.

4.4. Ablation Experiment in SeaBuoys

Similar to the experiment conducted in SeaShips, 90% of the images in SeaBuoys were used as the training set and 10% was used as the test set. The above-mentioned schemes were compared. mAP, FPS and the number of weights were selected as the evaluation indicators for ablation experiment. The experimental results in SeaBuoys are shown in the table below.

It can be seen from Table 2 that in the experiment conducted in SeaBuoys, our method still achieves the best experimental results. The mAP is 99.07% and the FPS is 68. The values are increased by 0.95% and 13FPS, respectively, compared with the results achieved by YOLO v4. The mAP of the schemes where all convolutions were replaced with RDSCs or DSCs is also smaller than the value of YOLO v4. Our method achieved the best results on both datasets, demonstrating that our method is applicable to sea surface object detection task. Similarly, the weight files trained in the SeaBuoys dataset were also analyzed, and the results are provided in Figure 14.

Table 2. Experimental results of different schemes under SeaBuoys.

Baseline	Method	mAP (%)	FPS	Number of Weights
YOLO v4	-	98.12	55	63,963,584
	Ours	99.07	68	35,524,224
	DSC in ResUnit	97.53	68	35,524,224
	DSC in all layers	91.41	58	14,814,724
	RDSC in all layers	93.23	53	14,814,724

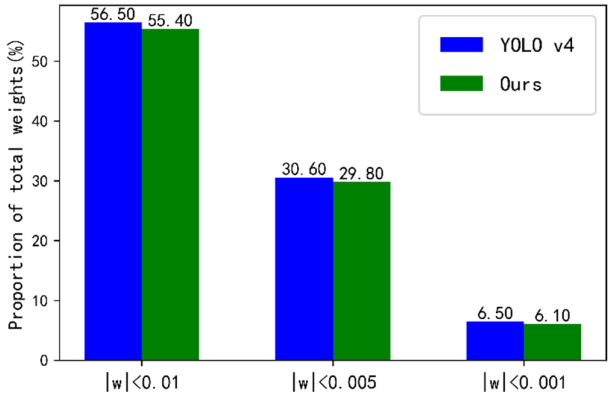


Figure 14. Statistical diagram of weight distribution of YOLO v4 and our method in the experiment conducted in SeaBuoys.

In Figure 14, the ratio of the weights with $|w| < 0.01$, $|w| < 0.005$ and $|w| < 0.001$ to the total number of weights in our method is reduced by 1.1%, 0.8%, and 0.4%, respectively, compared to the results of YOLO v4.

Table 3 shows the comparison between our proposed framework and several mainstream object detection algorithms. The experimental results showed that our proposed framework has good performance.

Table 3. Experimental results of several mainstream object detection algorithms.

Model	FPS	mAP in SeaShips (%)	mAP in SeaBuoys (%)
Faster RCNN + ResNet50	7	88.25	94.28
EfficientDet-D1	22	84.09	89.89
EfficientDet-D0	29	78.02	84.47
YOLO v4	55	92.80	98.12
Cross YOLO v3 [40]	45	92.85	98.25
Ours	68	94.58	99.07

In Table 3, different network models were experimentally compared under the condition of the same experimental equipment. Our method achieved the best results in terms of detection speed and accuracy. Compared with the one-stage series algorithms, the two-stage series algorithms such as Faster-RCNN have a relatively huge disadvantage in terms of speed. This is caused by different detection methods, not because of the excessively large network model and too many components.

In the ablation experiment, the detection speed and accuracy of the improved network framework were analyzed. Compared with the original network, the proposed network has greatly improved performance, especially the detection speed. In order to explore the influence of RDSC on the network model in more detail, DSC was used to replace RDSC for comparison, and DSC and RDSC were also applied to different positions of YOLO v4. The experimental results of these schemes showed that the mAP and FPS of our method reached the highest value. In the ablation experiment, the weight file was also analyzed. The distribution of the weight value demonstrated that the utilization rate of the weight in our method is higher.

Figure 15 shows the detection results of our methods under different marine environments such as night and heavy fog. The results suggested that our method has strong robustness for large and small objects under different ocean conditions. Figure 16 shows the target detection results of the unmanned surface vessel in real sea conditions.

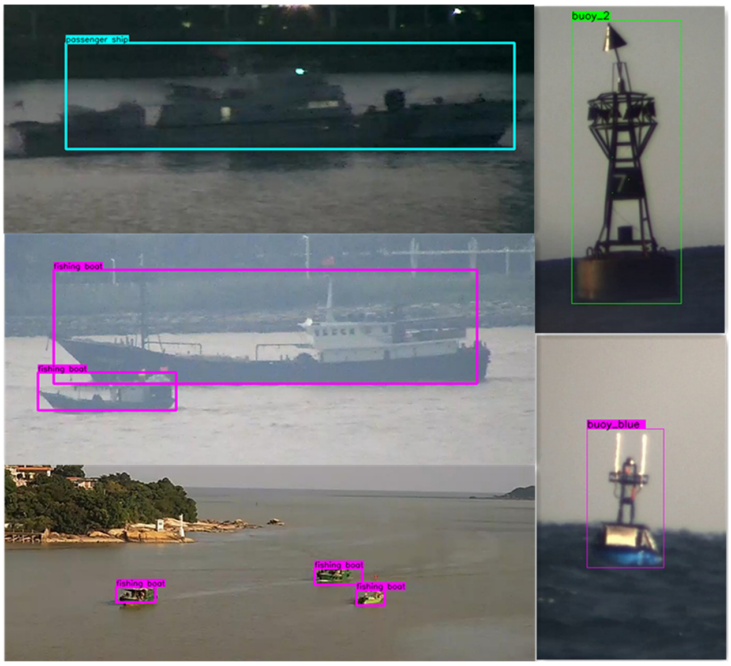


Figure 15. The detection results of our method in the dark or foggy environment. The left side of the picture is the detection result in the SeaShips dataset, and the right side is the detection result in SeaBuoys, which includes large or small targets.

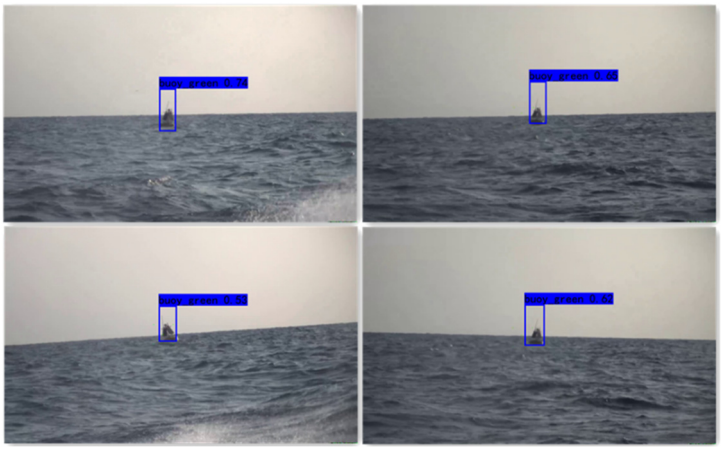


Figure 16. This is the detection result of the sea surface buoy in the real situation. The target in the picture is small and the light condition is poor, in this case, our method still performs well. The label of the buoy in the picture is buoy_green.

5. Conclusions

Based on the engineering application of real-time sea surface object detection of USVs, the network structure of YOLO v4 was improved in this paper, and experiments were conducted in SeaShips and SeaBuoys datasets. The mAP values in the two datasets reached 94.58% and 99.07%, respectively, and the FPS increased by more than 20%. The experimental results revealed that the improved YOLO v4 has great potential in the sea surface object detection task. Compared with the original version of YOLO v4, the improved YOLO v4 increased mAP and FPS in both datasets, achieving a good balance between speed and accuracy. RDSC was proposed. Compared with ordinary convolutional networks, RDSC greatly reduced the number of weights, so we greatly increased the detection speed of YOLO v4, and the structure of YOLO v4 was carefully designed. We tried to replace all convolutions of YOLO v4 with RDSC, but the result was not ideal. This is another contribution of our work. Use RDSC in the right place. The overall structure of YOLO v4 was retained. This allows our scheme to improve the accuracy and speed of detection without adding additional components to the original network framework. In the future, we hope that our method can have a wider range of applications, such as some other engineering backgrounds, not only limited to sea targets, we hope to have good performance in more datasets.

Author Contributions: Conceptualization, T.L., B.P., L.Z., W.Y. and X.S.; methodology, T.L., L.Z. and B.P.; software, B.P.; validation, T.L., L.Z., W.Y. and X.S.; investigation, T.L. and B.P.; resources, T.L. and L.Z.; writing—original draft preparation, T.L. and B.P.; writing—review and editing, T.L., L.Z.; funding acquisition, L.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Fundamental Research Funds for the Central Universities (3072020CFT0102), Equipment Pre-research Key Laboratory Fund (6142215190207), China National Offshore Oil Corp (CNOOC) Research Center, the National Natural Science Foundation of China Project (51409059) and Ministry of Industry and Information Technology's scientific research project "Natural Gas Hydrate Drilling and Production Ship Engineering Development".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Dataset download link: http://www.lmars.whu.edu.cn/prof_web/shaozhenfeng/datasets/SeaShips%287000%29.zip. accessed date: any time Code link: https://pan.baidu.com/s/13FmV5YbYp7iGVJgcZAL3jg. Extraction code: ymud.

Acknowledgments: The authors would like to express their gratitude to all their colleagues in the National Key Laboratory of Science and Technology on Autonomous Underwater Vehicle, Harbin Engineering University.

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

References

1. Zhang, W.; Gao, X.-Z.; Yang, C.-F.; Jiang, F.; Chen, Z.-Y. A object detection and tracking method for security in intelligence of unmanned surface vehicles. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *2020*, 1–13. [CrossRef]
2. Glowacz, A. Ventilation diagnosis of angle grinder using thermal imaging. *Sensors* **2021**, *21*, 2853. [CrossRef]
3. Hashmi, M.F.; Pal, R.; Saxena, R.; Keskar, A.G. A new approach for real time object detection and tracking on high resolution and multi-camera surveillance videos using GPU. *J. Central South Univ.* **2016**, *23*, 130–144. [CrossRef]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 60. [CrossRef]
5. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
6. Lin, M.; Chen, Q.; Yan, S.-C. Network in network. *arXiv* **2013**, arXiv:1312.4400.
7. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
8. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
10. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
11. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. *Densely Connected Convolutional Networks*; IEEE Computer Society Press: Piscataway, NJ, USA, 2016.
12. Wang, C.; Liao, H.; Yeh, I.; Wu, Y.; Chen, P.; Hsieh, J.W. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020.
13. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
14. 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]
15. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. *Mask R-CNN*; IEEE Computer Society Press: Piscataway, NJ, USA, 2017.
16. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6154–6162.
17. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
18. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
19. Redmon, J.; Farhadi, A. YOLOv3: An incremental improvement. *arXiv* **2008**, arXiv:1804.02767.
20. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-J.M. YOLOv4 Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934.
21. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Transactions on Petri Nets and Other Models of Concurrency XV, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
22. Lin, T.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. *Feature Pyramid Networks for Object Detection*; IEEE Computer Society Press: Piscataway, NJ, USA, 2017.
23. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020.
24. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. NAS-FPN: Learning scalable feature pyramid architecture for object detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 7029–7038.
25. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
26. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
27. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-excitation networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef] [PubMed]
28. Li, X.; Wang, W.; Hu, X.; Yang, J. Selective kernel networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
29. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.

30. Wang, Q.; Shen, F.; Cheng, L.; Jiang, J.; He, G.; Sheng, W.; Jing, N.; Mao, Z. Ship detection based on fused features and rebuilt YOLOv3 networks in optical remote-sensing images. *Int. J. Remote Sens.* **2021**, *42*, 520–536. [CrossRef]
31. Cui, H.; Yang, Y.; Liu, M.; Shi, T.; Qi, Q. Ship Detection: An Improved YOLOv3 Method. In Proceedings of the OCEANS 2019, Marseille, France, 17–20 June 2019.
32. Lin, Z.; Ji, K.; Leng, X.; Kuang, G. Squeeze and excitation rank faster R-CNN for ship detection in SAR images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 751–755. [CrossRef]
33. Xiao, Q.; Cheng, Y.; Xiao, M.; Zhang, J.; Shi, H.; Niu, L.; Ge, C.; Lang, H. Improved region convolutional neural network for ship detection in multiresolution synthetic aperture radar images. *Concurr. Comput. Pract. Exp.* **2020**, *32*, e5820. [CrossRef]
34. Qi, L.; Li, B.; Chen, L.; Wang, D.; Dong, L.; Jia, X.; Huang, J.; Ge, C.; Xue, G. Ship target detection algorithm based on improved faster R-CNN. *Electronics* **2019**, *8*, 959. [CrossRef]
35. Fan, W.; Zhou, F.; Bai, X.; Tao, M.; Tian, T. Ship detection using deep convolutional neural networks for PolSar images. *Remote Sens.* **2019**, *11*, 2862. [CrossRef]
36. Zhang, D.; Zhan, J.; Tan, L.; Gao, Y.; Župan, R. Comparison of two deep learning methods for ship target recognition with optical remotely sensed data. *Neural Comput. Appl.* **2021**, *33*, 4639–4649. [CrossRef]
37. Li, Y.; Guo, J.; Guo, X.; Liu, K.; Zhao, W.; Luo, Y.; Wang, Z. A novel target detection method of the unmanned surface vehicle under all-weather conditions with an improved YOLOV3. *Sensors* **2020**, *20*, 4885. [CrossRef]
38. Jie, Y.; Leonidas, L.; Mumtaz, F.; Ali, M. Ship detection and tracking in Inland waterways using improved YOLOv3 and deep SORT. *Symmetry* **2021**, *13*, 308. [CrossRef]
39. Wang, J.; Lin, Y.; Guo, J.; Zhuang, L. SSS-YOLO: Towards more accurate detection for small ships in SAR image. *Remote Sens. Lett.* **2021**, *12*, 93–102. [CrossRef]
40. Liu, T.; Pang, B.; Ai, S.; Sun, X. Study on visual detection algorithm of sea surface targets based on improved YOLOv3. *Sensors* **2020**, *20*, 7263. [CrossRef] [PubMed]
41. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef] [PubMed]
43. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. SeaShips: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimedia* **2018**, *20*, 2593–2604. [CrossRef]
44. He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; Li, M. Bag of tricks for image classification with convolutional neural networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.