International Journal of
Geo-Information

MDPI

Article

A Data Augmentation Strategy Based on Simulated Samples for Ship Detection in RGB Remote Sensing Images

Yiming Yan, Zhichao Tan and Nan Su *

Department of Information Engineering, Harbin Engineering University, Harbin 150001, China;
yanyiming@hrbeu.edu.cn (Y.Y.); tony@hrbeu.edu.cn (Z.T.)
* Correspondence: sunan08@hrbeu.edu.cn; Tel.: +86-186-0450-4579

Received: 26 February 2019; Accepted: 26 May 2019; Published: 13 June 2019

check for updates

Abstract: In this paper, we propose a data augmentation method for ship detection. Inshore ship detection using optical remote sensing imaging is a challenging task owing to an insufficient number of training samples. Although the multilayered neural network method has achieved excellent results in recent research, a large number of training samples is indispensable to guarantee the accuracy and robustness of ship detection. The majority of researchers adopt such strategies as clipping, scaling, color transformation, and flipping to enhance the samples. Nevertheless, these methods do not essentially increase the quality of the dataset. A novel data augmentation strategy was thus proposed in this study by using simulated remote sensing ship images to augment the positive training samples. The simulated images are generated by true background images and three-dimensional models on the same scale as real ships. A faster region-based convolutional neural network (Faster R-CNN) based on Res101netwok was trained by the dataset, which is composed of both simulated and true images. A series of experiments is designed under small sample conditions; the experimental results show that better detection is obtained with our data augmentation strategy.

Keywords: data augmentation; optical remote sensing image; ship detection; simulated samples; deep learning

1. Introduction

Technology aimed at the detection of ships has prospects for wide application in the military and civilian fields. With the development of remote sensing technology, inshore ship detection has become a hot topic for the application of remote sensing images. The biggest challenge for inshore ship detection is the changing background. In addition, ships, which form the main mode of transportation at sea, can be of varying types and shapes. This makes their accurate detection difficult.

In recent years, with the advancement of deep learning techniques (especially the convolutional neural network (CNN)), research and development on object detection techniques has significantly progressed. A number of methods that use convolutional neural networks to detect ships have been proposed in recent years. Lin et al. utilized a fully convolutional network (FCN) to tackle the problem of inshore ship detection [1]. Zou et al proposed singular value decompensation network (SVD-net), which is designed on the basis of the convolutional neural network and the singular value decompensation algorithm [2]. Li et al. proposed hierarchical selective filtering network (HSF-net), a regional proposal network to generate ship candidates from feature maps produced by a deep convolutional neural network [3].

In addition, region-based CNN (R-CNN) [4], one of the earliest algorithms to employ CNN to detect objects, has achieved success and demonstrated a capacity to detect objects. Following R-CNN, the Spatial Pyramid Pooling network (SPP-net) [5] was proposed, which has a Spatial Pyramid Pooling layer, improved accuracy, and reduced runtime. Subsequently, Fast R-CNN [6], which utilizes a region of interest (RoI) pooling layer, was developed to further increase the speed. In Faster R-CNN [7], region proposal networks (RPNs) are utilized to calculate region proposals, following which the Fast R-CNN network classifies them. These two networks share the same feature map, which provides a speed advantage over Fast R-CNN. You Look Only Once (YOLO) [8], through a region proposal network, produces a large number of potential bounding boxes that may contain objects. A classifier is then used to determine whether each bounding box contains objects. Single Shot Multibox Detector (SSD) [9] further improved the detection accuracy and speed. These object detection methods have been applied to ship detection. Liu et al. introduced the rotated-region-based CNN (RR-CNN), which can identify features of rotated regions and precisely locate rotated objects [10]. Yang et al. proposed a framework called Rotation Dense Feature Pyramid Networks (R-DFPN), which effectively detects ships in different scenes, including oceans and ports [11]. They proposed a rotation anchor strategy to predict the minimum circumscribed rectangle, reduce the size of the redundant detection region, and improve recall, thus achieving excellent results.

However, these deep artificial neural networks require a large corpus of training data in order to work effectively. Ship detection requires not only image data, but also extensive manual annotation, which is costly, labor-intensive, and time-consuming. However, there is a lack of images of ships that make special appearances or are in unique situations. In addition, when a sufficient number of ship images cannot be obtained, the detection results are poor. An effective solution, thus, is to artificially inflate the dataset with data augmentation, including both geometric and photometric transformations. Geometric transformations alter the geometry of the images, and include flipping, cropping, scaling, and rotating. Photometric transformations amend color channels, offering changes in illumination and color [12] together with color jittering, noise perturbation, and Fancy Principle Component Analysis (PCA). In addition to these methods, a number of new approaches have been proposed and good results have been obtained with them. Translational and rotational symmetry has been exploited [13]. The use of Perlin noise with a Fusion Net has resulted in higher accuracy than conventional data augmentation [14]. Fawzi et al. [15] proposed a new automatic and adaptive algorithm to choose transformations of the samples. Qiu et al. [16] analyzed the performance of geometric and non-geometric operations in FCNs, and highlighted the significance of analyzing deep neural networks with skip architectures. The Generative Adversarial Network (GAN) [17] has also been used to make progress in data augmentation. GAN-based methods have produced high-quality images in some datasets [18,19]. However, the GAN-based methods are difficult to train and require a certain amount of data.

These methods could make good use of the information in the images; however, they do not essentially increase the quality of the dataset. To address this problem, this paper proposes a method that combines a three-dimensional model of a ship with remote sensing images to construct simulation images that help to augment the training data. Better results can be obtained by adding new information to the images.

The main contributions of this paper are as follows. To address the problem of an insufficient number of training samples, a training strategy to produce simulation images is proposed that augments the dataset. Meanwhile, new information on ship models is added to the images. This results in networks that are trained more effectively. In addition, the bounding boxes of the simulated ship objects are automatically annotated. The application of this method to inshore images saves time and effort as compared to manual annotation.

2. Methods

Three-dimensional (3D) ship models, which represent the appearance and structure of real ships, are generated from ship-scale models. These models can replace the original ships that appear in real images. By projecting ship models of different angles onto real images, simulated images with simulated ships are generated. This results in an increase in the number of ship objects. With this strategy, we propose a data augmentation method for ship detection.

Figure 1 shows the proposed ship detection framework. The framework consists of two main parts, that is, the generation of simulated images and the composition of the training sample dataset. Specifically, a series of simulated images is generated by projecting specific ship models onto real inshore images. Then, the training sample dataset is composed of these simulation images. To enhance the training samples, we can choose whether to add real images to the training sample dataset. Finally, we train the Faster R-CNN with the training samples to obtain a fine-tuned detection model and use it to detect the ship objects in the test samples.

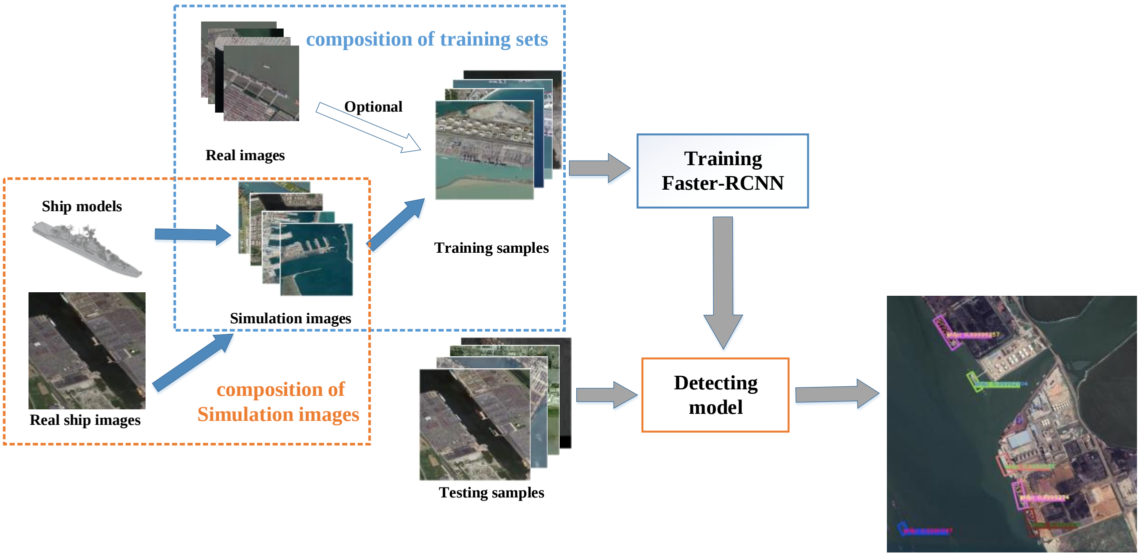


Figure 1. The framework of the proposed ship detection method.

In the simulation samples, there are two types of ship objects: real ship objects and simulated ship objects, which can be represented by the formula (1).

$$S = \{S_{real} \ \cup \sum_{i=1}^{N} S_{si} \ i = 1,2,\ldots,N\} \tag{1}$$

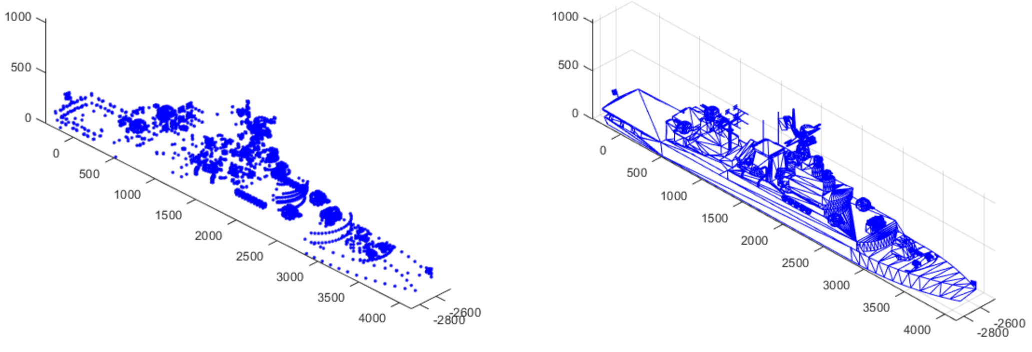
where S_{real} represents real objects and S_{si} represents simulated ships added to each inshore image in N inshore images. Moreover, these simulated ship objects are composed of the sum of S_{si}.

2.1. Ship Models Generated in Three-Dimensional Space

In the process of collection of the images, according to the frequency at which ships appear in images, three categories of ships—cargo ships, cruise ships, and warships—were selected. We collected three-dimensional point cloud data on two different sub-types of each of these three kinds of ships. Each ship's three-dimensional data were generated with three-dimensional laser scanning. Considering the physical issue with real ships, we scan the scaled model of the ship to fit the shape and structure.

These point clouds generate the ship model, and the denser the point cloud, the more accurate the ship model. Each ship's point cloud model was plotted as shown in Figure 2a. Then, points in the three-dimensional space form a series of triangles. After constructing the topology relation among the triangles, a preliminary ship model was generated as shown in Figure 2b. These triangular surface plots are shown in the following formula:

$$Model_T = T\left(t_k, \ p_j\right) \quad k = 1,2,\ldots,M \quad j = 1,2,\ldots,N. \tag{2}$$

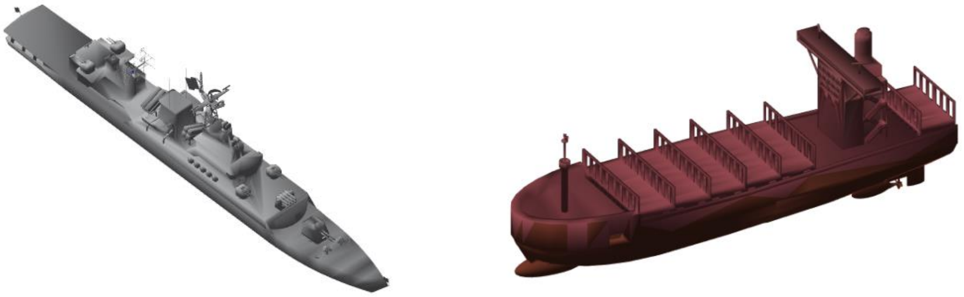


(**a**) (**b**)

Figure 2. (**a**) A ship's point cloud model. (**b**) A ship model with triangles.

In formula (2), p_j represents each point in the point cloud, and t_k represents the order in which the points are connected in the triangles and the index relation between every triangle.

Different colors were given to the different types of ships, such as red or green for cargo ships and gray or black for warships. Then, we provided the right amount of reflection on the landscape surface of the ship model, and adjusted the light and contrast. After the above steps, the final ship models were generated, which are shown in Figure 3.



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

Figure 3. The final ship models. (**a**) The warship. (**b**) The cargo ship.

2.2. Models Projected onto the Inshore Images

After the ship models were generated, we transferred the ship models from model space to world space. First, we needed to adjust the size of the ship model generated in the previous step to the size of the corresponding real image. For example, if the ship is 100 m in length and the resolution of the image is 2 m, then the ship's length is 50 pixels (100 m/2 m) in the image. The transformation relationship is shown in formula (3).

$$Model_N = Model_r \cdot K = Model_r \cdot I \cdot \frac{R}{P} \tag{3}$$

In formula (3), $Model_r$ represents the real ship, and $Model_N$ represents the reduced ship model. Variables I, R, and P represent the scale fraction of the image, the ratio between real ships and proportional ship models, and the length of a pixel unit in the actual image, respectively.

With the real inshore image as the base, we increased the third dimension coordinates (Z-axis), establishing a world coordinate system. The point $p\ (X_M,\ Y_M,\ Z_M)$ of the ship model in model space is translated to the point p' (X_C, Y_C, Z_C) of the ship model in world space. The transformation relation is shown in formula (4).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} \tag{4}$$

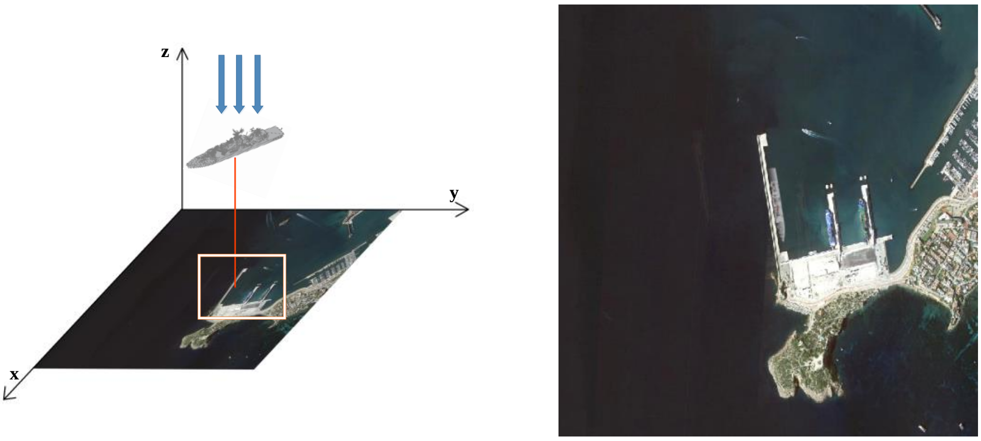
$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \sin\alpha\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma \\ -\sin\beta & \cos\beta\sin\gamma & \cos\beta\cos\gamma \end{bmatrix} \tag{5}$$

In formula (4), T describes a new position of the ship model in world space relative to model space, where t_x is the X-axis orientation in the new space, t_y is the Y-axis orientation in the new space, and t_z is the Z-axis orientation in the new space. The matrix R is the sum of the effects of rotation around the three axes, respectively, chaining three transformations together by multiplying the matrices one after the other. For the R matrix, only the yaw angle of the ship's rotation needs to be determined; the pitch angle and roll angle are both zero.

For each ship model to be projected onto the inshore image, it was necessary to select three points ($\mathrm{x_p}$, $\mathrm{y_p}$, $\mathrm{x_o}$, $\mathrm{y_o}$, $\mathrm{x_t}$, $\mathrm{y_t}$) on the image manually to determine the projection position and the angle of the ship model. $\mathrm{x_p}$ and $\mathrm{y_p}$ determine the position of the ship's projection, and $\mathrm{x_o}$, $\mathrm{y_o}$, $\mathrm{x_t}$, and $\mathrm{y_t}$ determine the angle of the yaw β. The rotation angle of the yaw β is determined from the point of origin ($\mathrm{x_o}$, $\mathrm{y_o}$) and the point of termination ($\mathrm{x_t}$, $\mathrm{y_t}$). Thus, the transformation relation is shown in formula (6).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} == \begin{bmatrix} \cos\beta & 0 & \sin\beta & x_p \\ 0 & 1 & 0 & y_p \\ -\sin\beta & 0 & \cos\beta & a \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} \quad \beta = \arctan\frac{(y_t - y_o)}{(x_t - x_o)} \tag{6}$$

In formula (6), the value a is a certain value that represents that the ship model is placed at a certain height above the image. An orthographic projection was used to project the model onto the image to generate simulated images. The schematic diagram of the generation of simulation images is shown in Figure 4a, and Figure 4b shows a generated simulation image.



(**a**)

(**b**)

Figure 4. (**a**) Schematic of simulation image generation. (**b**) The image generated from Figure 3a.

As shown in Figure 4, the point $p'\ (X_C,\ Y_C,\ Z_C)$ of the ship model in world space is translated to the point $p''\ (u,\ v)$ of projection on the inshore image; the transformation relation is shown in formula (7).

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/Z_c & 0 \end{bmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{7}$$

In order to determine the projection positions of the ship models and generate the simulation images, the inshore images were segmented into sea and land area images, as shown in Figure 4a. In the sea area, the projection position of the ship model along the coast was usually chosen to simulate real ships docking along the coast. The generated simulation image is shown in Figure 5b, in which the simulated ships are located in a specific region.

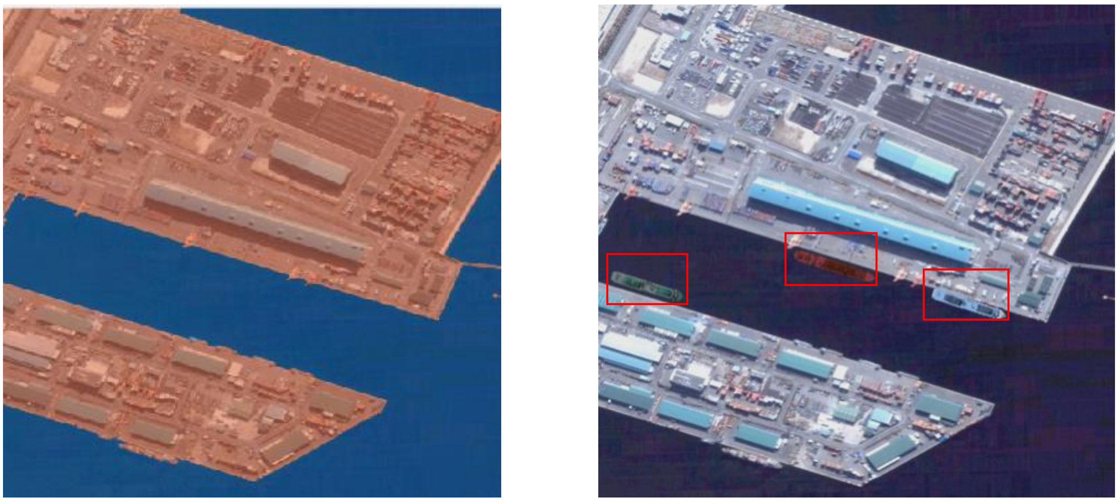
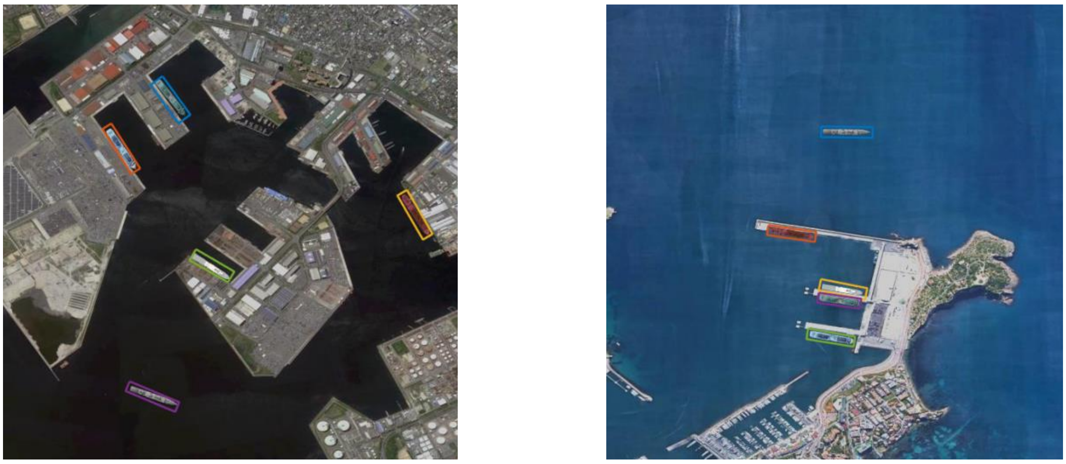

(**a**) (**b**)

Figure 5. (**a**) Diagram of sea–land segmentation. (**b**) Generated simulation image.

When the location and angle of the projections were obtained in the previous step, the bounding boxes of projections were also obtained. Each projection location was annotated by quadrilateral bounding boxes. Each bounding box was denoted as "*m1, n1, m2, n2, m3, n3, m4, n4*", where *mi* and *ni* denote the positions of the oriented bounding boxes' vertices in the image. Thus, we can obtain the annotation of projections directly. Figure 6 shows the images in which simulated ships with bounding boxes are located.



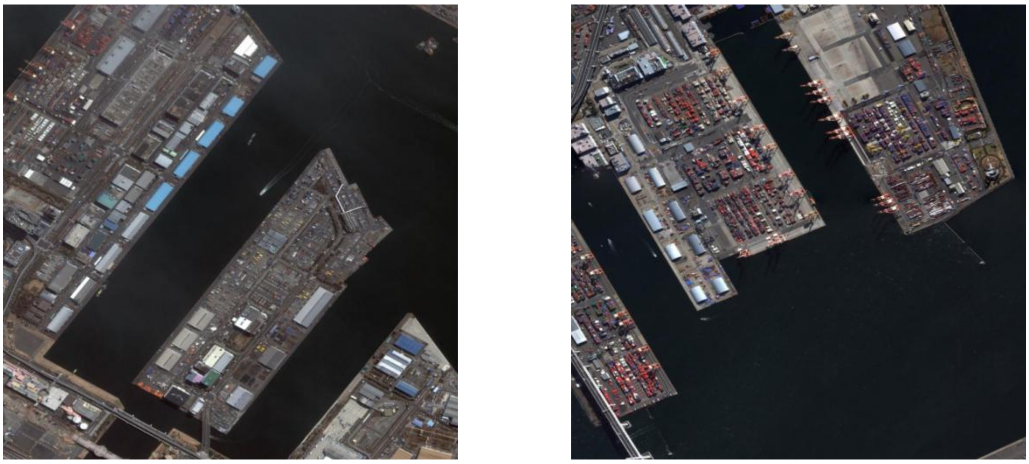
(**a**) (**b**)

Figure 6. (**a**,**b**) are part of the simulated image set with annotations.

3. Experiments and Analysis

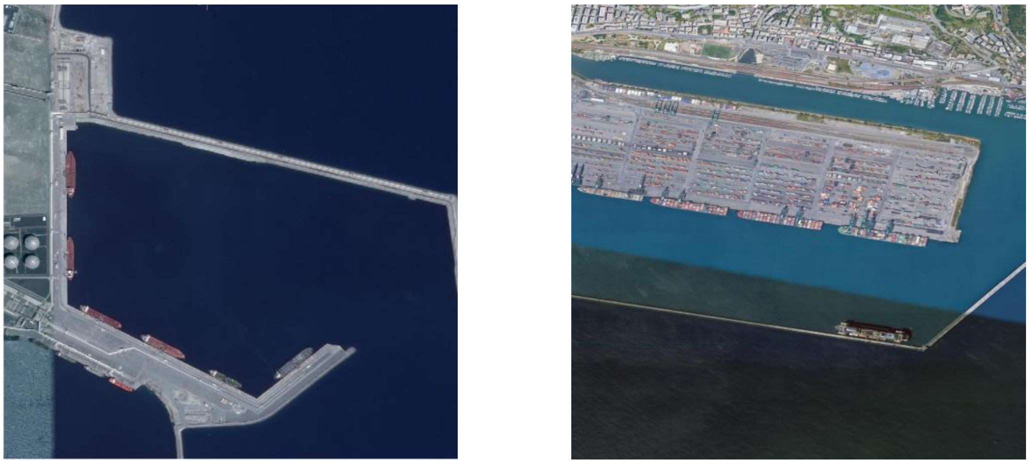
3.1. Experiment Dataset

The optical remote sensing images that were used in the experiment were obtained from Google Earth. The size of the simulation image that was generated through projection is 683*683, and the spatial resolution of the image is approximately 3 m. The experiment dataset contains 250 images, and is made up of two parts: 50 images without ships and 200 images with ships. The 200 images were randomly separated into two subsets: 100 images were used to generate simulation images; the others were used for testing. Some images are shown in Figure 7.



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

Figure 7. *Cont.*



(**c**) (**d**)

Figure 7. (**a**,**b**) are images without ships. (**c**,**d**) are images with ships.

3.2. Performance Evaluation

Intersection over Union (IOU) was used measure the accuracy of the ship object detector. The Intersection over Union threshold was set to 0.3. For each ship detection result, the IOU value is calculated, and if IOU > 0.3, it is considered to be a true detection; otherwise, it is considered to be a false detection. In the experiments, precision, recall, and the F-measure were used to quantitatively evaluate our ship detection results on the test images. Precision, recall, and the F-measure are defined, respectively, as:

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \tag{10}$$

where TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives. The F-measure is a comprehensive evaluation index.

3.3. Method and Network Chosen for the Detection of Ships in Images

Due to the slender shape of a ship, the rotation anchor method was chosen to precisely predict the minimum circumscribed rectangle. In this section, the Rotation Dense Feature Pyramid Networks (R-DFPN) [11] method is adopted. In addition, a pre-trained ResNet-101 [20] model is used for further experiments.

3.4. Experimental Results from the Full Ship Simulation

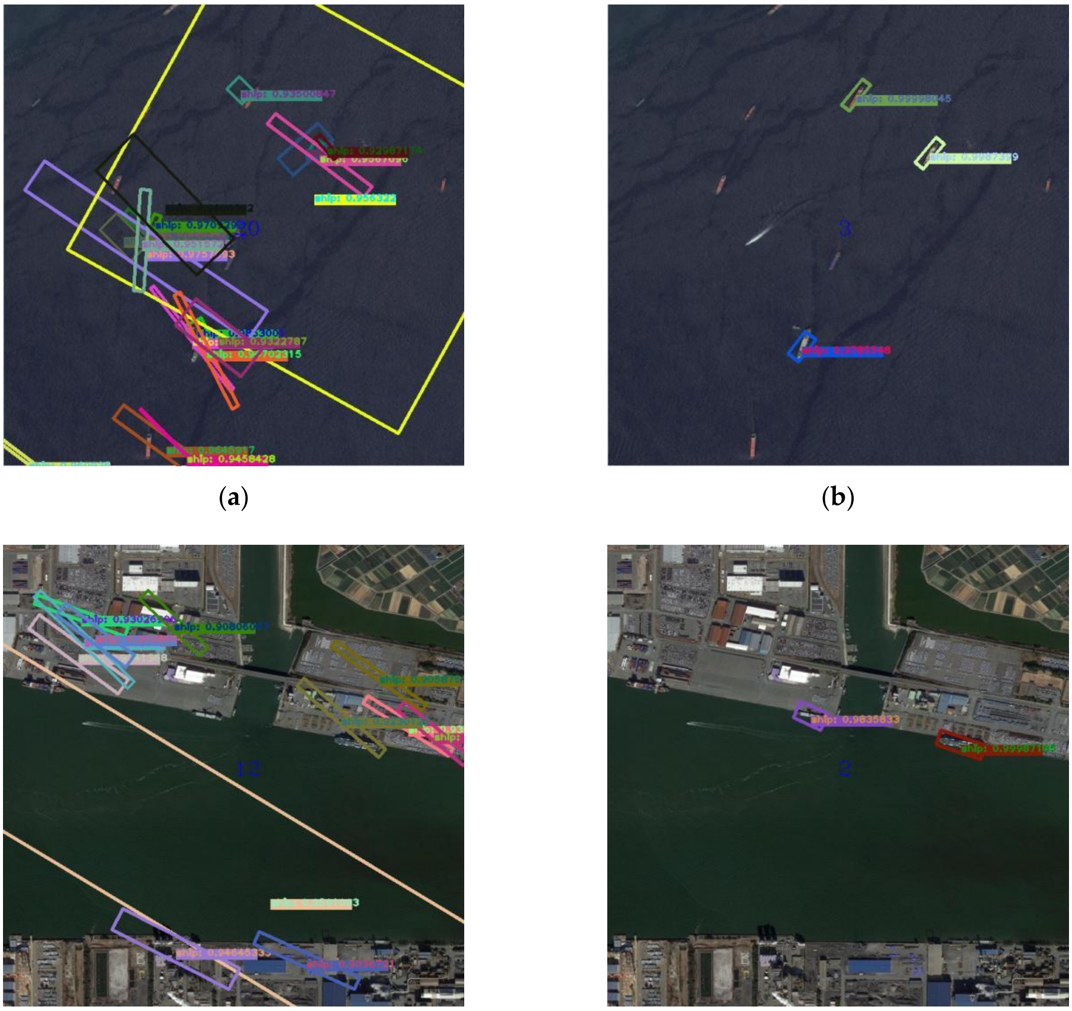
Here, we selected 50 inshore images with no ship objects in them. Then, these 50 inshore images were processed with our data augmentation method (to combine the images and the ship models; the number of ship models in each image was 5), and 50 simulated inshore images were generated. After obtaining the 50 simulated images with about 250 samples, these simulated images were used to train the network, and the trained models were tested with 100 real inshore remote sensing images with 685 samples. The results of the test are shown in Table 1 and Figure 8.

In order to eliminate interference from the pre-trained model in the experimental results, a set of contrast tests was set up. The original pre-trained model and the model that was trained with the 50 simulated images were used to test the 100 images.

Table 1. The results from different models.

Model	Precision	Recall	F-Measure
pre-trained model	5.04%	1.73%	2.58%
Model trained with 50 simulation images	40.24%	83.49%	54.31%

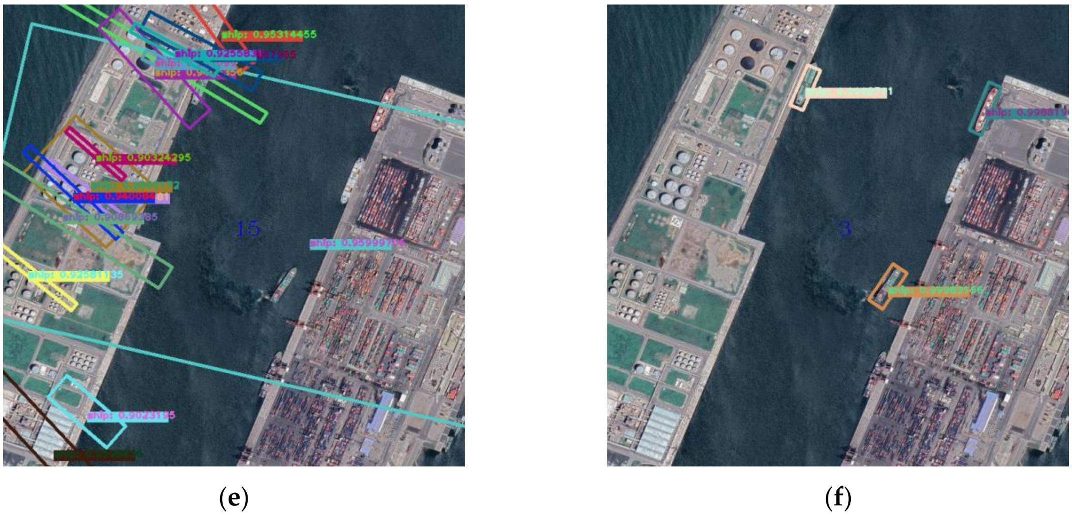
It is apparent that the result obtained using the pre-trained model alone is poor. It is difficult for the pre-trained model to detect ships in the true images, which can be seen from Figure 8a,c,e. Thus, we can exclude interference from the pre-trained model. The results also show that the detection network that was trained with these 50 images can detect some of the ships in the true images. As shown in Figure 8b,d,f, it can be seen that some of the ships can be detected with the detection network that was trained using the 50 simulation images. The recall rate is 40.24%, and the precision rate is 83.49%. The comparison and analysis of these experiments verify the ability of the proposed data augmentation method. These simulated images can be used to replace the real images and reduce the need for them in the detection model.



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

(**c**) (**d**)

Figure 8. *Cont.*



(**e**) (**f**)

Figure 8. (**a**,**c**,**e**) are some of the results from the pre-trained model. (**b**,**d**,**f**) are some of the results from the model that was trained with the 50 simulation images.

Three inshore images from Figure 8 (one of the ocean, two of the harbor) were chosen to show the detection results of the different methods. In the image of the entire sea, the pre-trained network model produced false alarms and was hardly able to detect the ship objects, while our model was able to detect the ships in the inshore images. In the next two images, the pre-trained network model also failed to detect the ship objects, while our model was able to detect ships in the inshore images.

3.5. Comparison with Other Data Augmentation Methods

In this part of the experiment, 60 inshore remote sensing ship images were selected. Then, these 60 images were further processed with our data augmentation method and used to generate 60 simulated inshore images. In each simulated image, the number of ships that was projected into the image is approximately 3. Thus, these simulated images contain both real ship objects and simulated ship objects.

In this section, a proper comparison with other data augmentation methods is performed. The other data augmentation methods we chose were noise perturbation, flipping, and rotation. We experimented on the real images and the simulated images with these data augmentation methods. In the data augmentation method of noise perturbation, Gaussian noise was added to the images. Moreover, in the data augmentation method of flipping, the images were flipped up and down, and left and right. Then, in the data augmentation method of rotation, the images were rotated three times; the angles of rotation were 90°, 180°, and 270°. The two groups of 60 images were processed using these data augmentation methods. The same 100 images with 651 samples were tested using the models obtained from these groups, and the results are compared as shown in Table 2 and Figure 9. After the 60 real images were processed with the different data augmentation methods, we trained the network with these images. Then, the detection model was used to test 100 images with 651 samples. We selected two representative images from the 100 test inshore images: an image of the harbor near the coast; and an image of the sea's surface. Figure 9 shows the test results of these two images for each method.

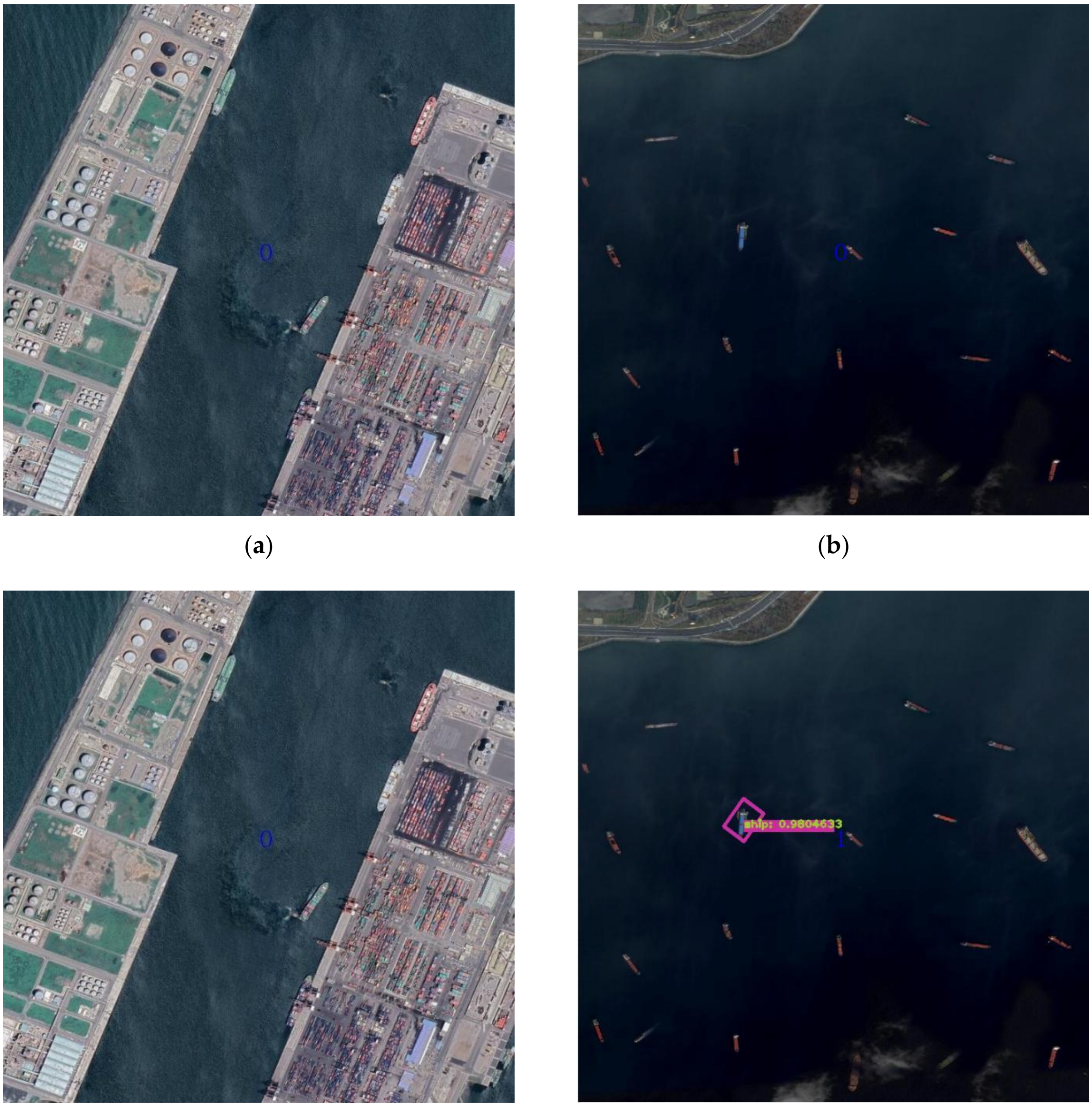
From Table 2 and Figure 9, we can see that the precision and F-measure of the 60 real images are 2.31% and 4.5%, respectively. As shown in Figure 9a,b, the ships were barely detected. When the noise perturbation data augmentation method was used on the 60 real images, the precision and F-measure were 5.53% and 10.48%, respectively. As shown in Figure 9c,d, a few ships were detected in the test images when the method of flipping was used on the real images, as the number of images was increased and the results were better. The precision was 33.64%, and the F-measure was 49.38%. It can also be seen in Figure 9e,f that some ships were detected when method of rotation was used on the real images. The precision was 42.95%, and the F-measure was 57.46%. As shown in Figure 9g,h, when the flip and rotation methods were both applied to the real images, the best results were obtained. The precision and the F-measure of the real images are 69.17% and 75.60%, respectively. As seen in Figure 9i,j, most of the ships in the images were detected. These results show that the data augmentation method can effectively improve the detection results by increasing the number of training samples.

Table 2. The results with different training image sets.

Method	Number of Images	Samples	Precision	Recall	F-Measure
60 real images	60	394	2.31%	100%	4.5%
60 real images with noise perturbation + 60 real images	120	788	5.53%	100%	10.48%
Flip 60 real images + 60 real images	180	1182	33.64%	92.80%	49.38%
Rotate 60 real images + 60 real images	240	1576	42.95%	89.58%	57.46%
Flip and rotate 60 real images + 60 real images	360	2364	69.17%	83.64%	75.60%
60 simulated images	60	582	12.44%	98.79%	22.10%
60 simulated images with noise perturbation + 60 simulated images	120	1164	19.35%	96.92%	32.27%
Flip 60 simulated images + 60 simulated images	180	1746	**61.74%**	**84.55%**	**71.36%**
Rotate simulated 60 simulated images + 60 simulated images	240	2328	**69.26%**	**78.57%**	**73.62%**
Flip and rotate 60 simulated images + 60 simulated images	360	3492	**75.57%**	**82.82%**	**79.04%**

Compared with the 60 real images, the results were further improved when these same data augmentation methods were applied to the 60 simulated images. The precision and F-measure of the simulated 60 images are 12.44% and 22.10%, respectively. Compared with the 60 real images, the precision and F-measure are 2.31% and 4.5%, respectively. As shown in Figure 9k,l, some ships were detected. Moreover, for the noise perturbation method, the precision changed from 5.53% to 12.44%, and the F-measure changed from 10.48% to 22.10%, which can be seen in Figure 9m,n, in which some ships were detected. For the flipping method, the precision changed from 33.64% to 61.74%, and the F-measure changed from 49.38% to 71.36%. As shown in Figure 9o,p, most of the ships in the images were detected. For the rotation method, the precision changed from 42.95% to 69.26%, and the F-measure changed from 57.46% to 73.62%. It can also be seen in Figure 9q,r that most of the ships in the images were detected. When using both the flip and rotate methods to process images, the best detection results were obtained. The model that was trained with real images obtained a precision of 69.17%, a recall of 83.64%, and an F-measure of 75.60%. The model that was trained with simulated images obtained a precision of 75.57%, a recall of 82.82%, and an F-measure of 79.04%.

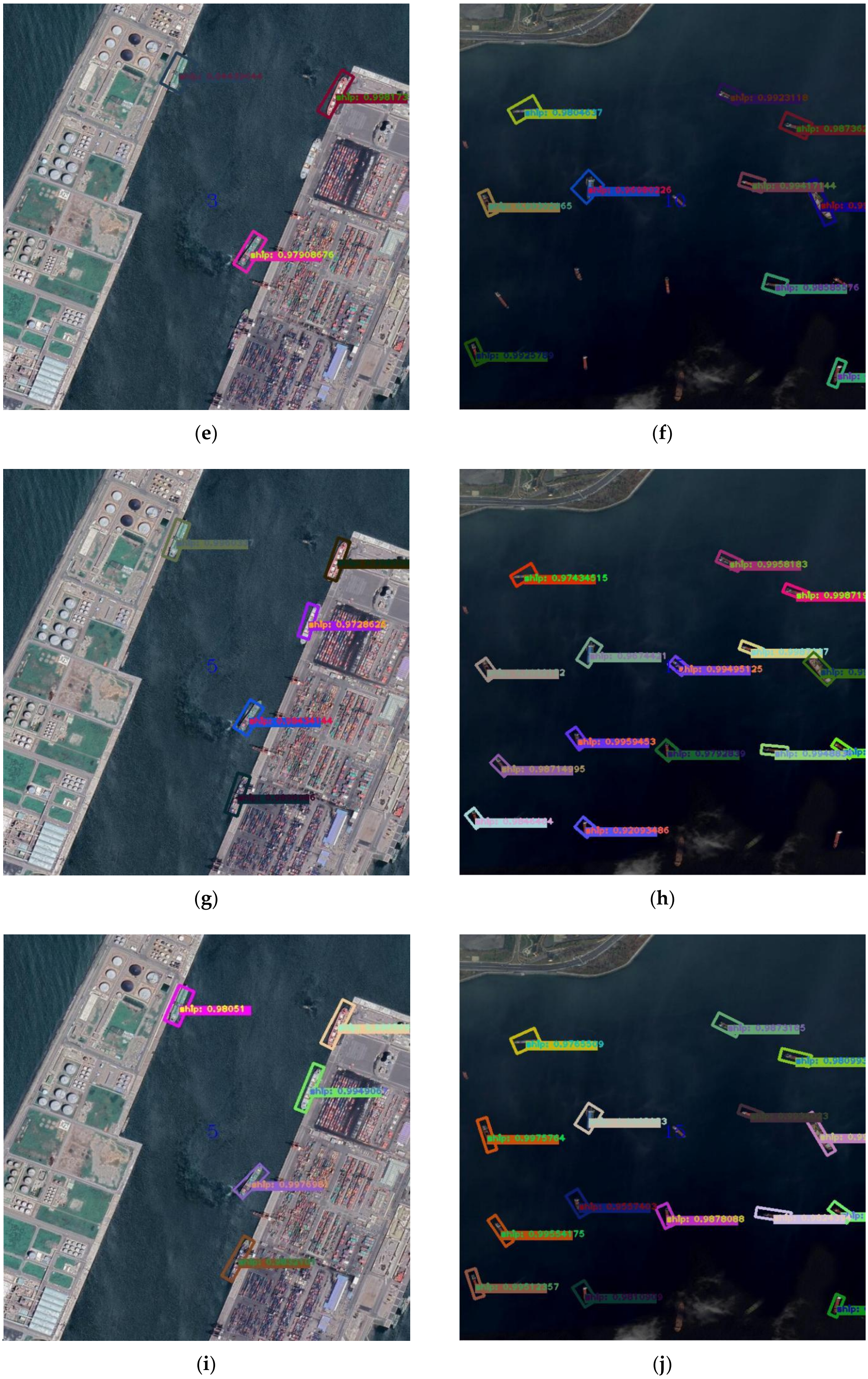
We may safely conclude that the number of samples increases with the adoption of data augmentation methods. The precision of ship detection has been greatly improved, though the recall of detection has decreased slightly. This indicates that, as the number of ship samples increases, the detection performance is correspondingly improved. The recall rate is reduced as precision and F-measure increase. The results show that our data augmentation method, when combined with the traditional method, can obtain better detection results. The result also proves that our proposed method is competent. This data augmentation method can improve the results of ship object detection. From the results of the 60 real images and 60 simulated images, we can conclude that the simulated ship objects that were constructed by projection can replace the ships in these real images. This method, when used together with other data augmentation methods, can increase the total number of ship objects without changing the number of images. In this way, the results of ship object detection in inshore images can be effectively improved.



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

(**c**) (**d**)

Figure 9. *Cont.*

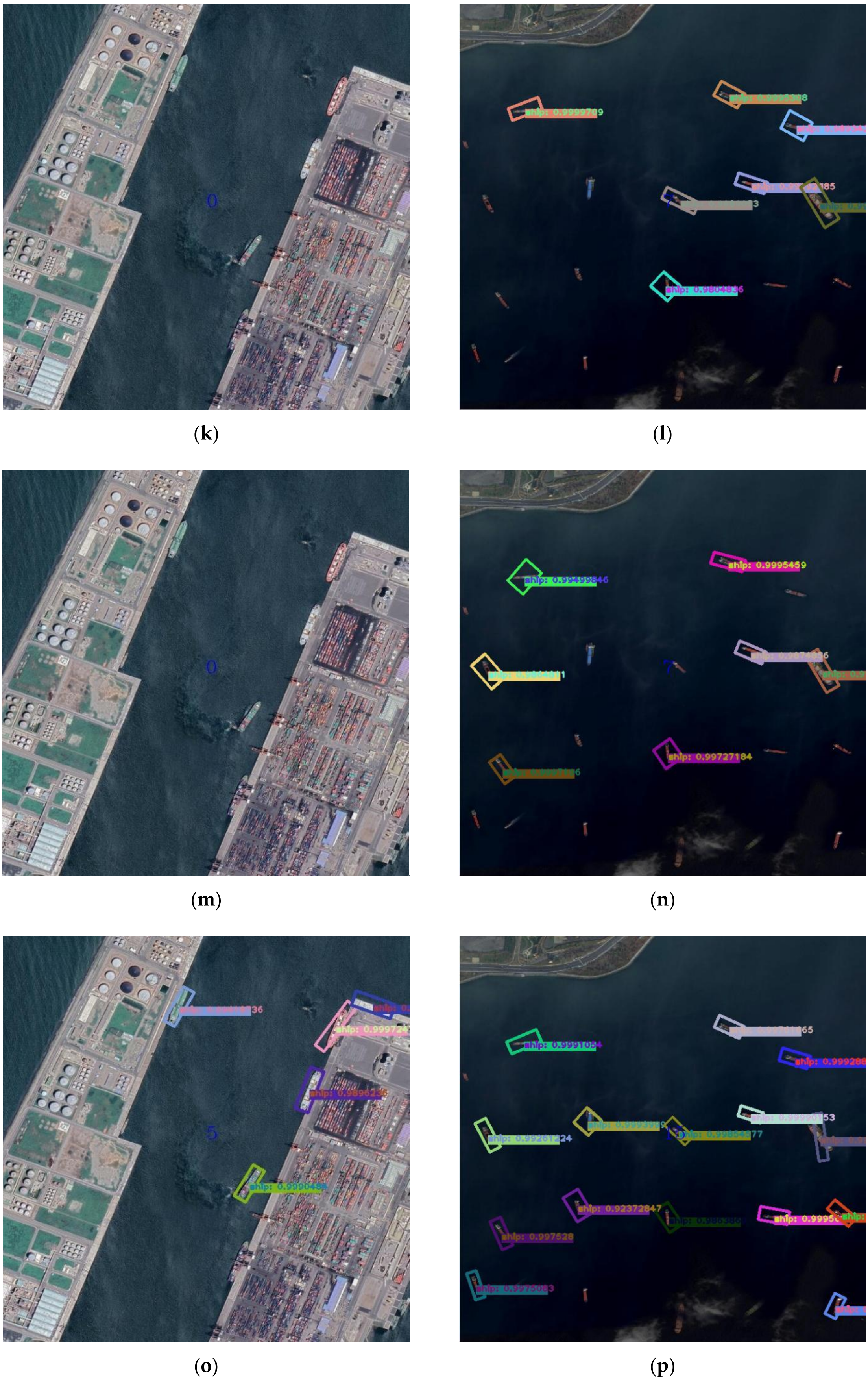


(e) (f)

(g) (h)

(i) (j)

Figure 9. *Cont.*

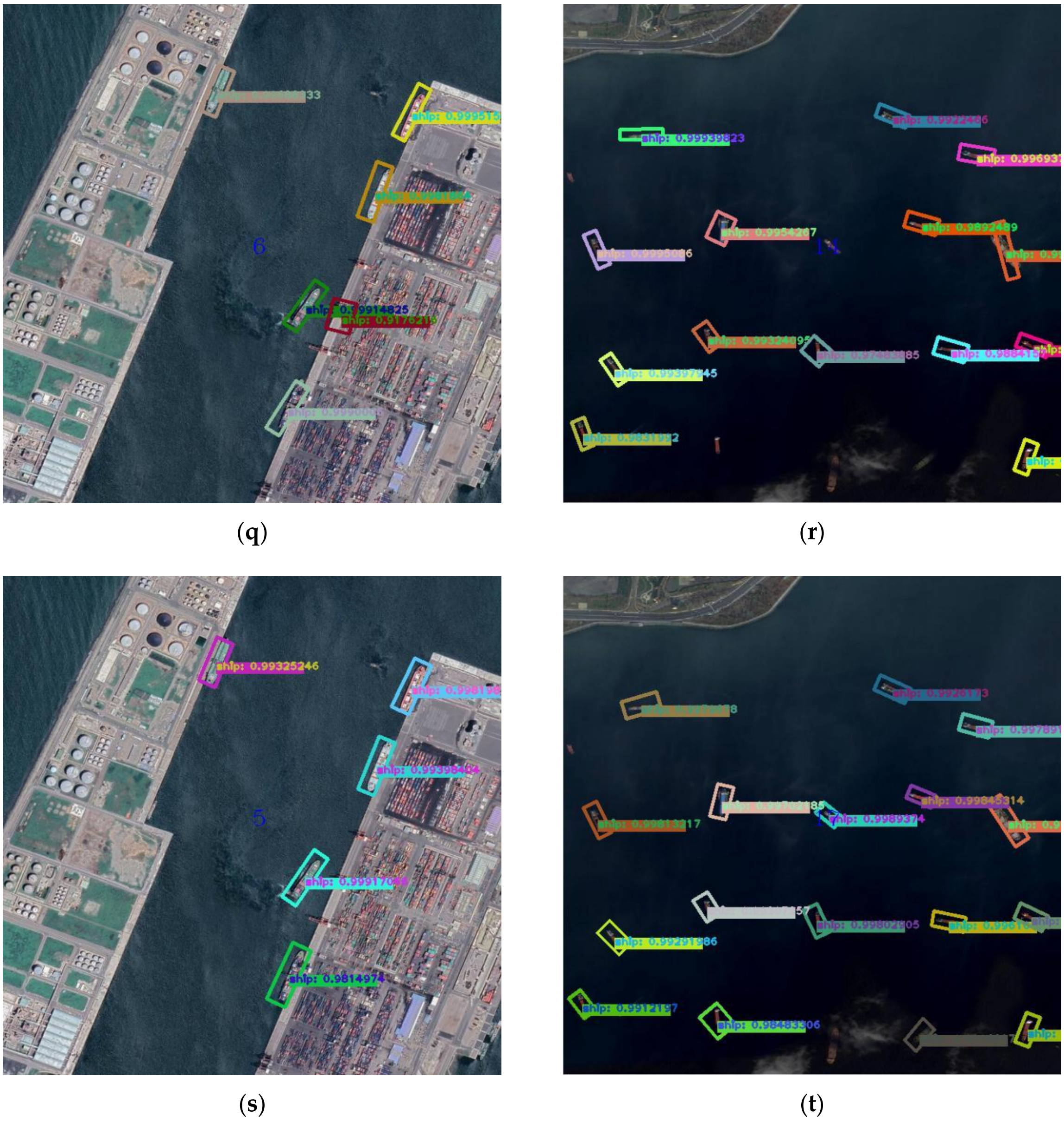


(**k**) (**l**)

(**m**) (**n**)

(**o**) (**p**)

Figure 9. *Cont.*



(**q**) (**r**)

(**s**) (**t**)

Figure 9. (**a**,**b**) are some of the results of 60 real images. (**c**,**d**) are some of the results of real images with noise perturbation. (**e**,**f**) are some of the results of real images with flipping. (**g**,**h**) are some of the results of real images with rotation. (**I**,**j**) are some of the results of real images with flipping and rotation. (**k**,**l**) are some of the results of the 60 simulated images. (**m**,**n**) are some of the results of simulated images with noise perturbation. (**o**,**p**) are some of the results of simulated images with flipping. (**q**,**r**) are some of the results of simulated images with rotation. (**s**,**t**) are some of the results of simulated images with flipping and rotation.

3.6. Results with Different RPN_IOU Thresholds

In this section, we discuss the effects of different RPN_IOU thresholds on the results. Since some of the ship objects in our training images are simulated ships, there may be a difference between these simulated ship objects and the real ship objects, which may influence the process of training the RPN network. If the difference affects the RPN_IOU positive threshold, some simulated ship objects may not be recognized in the training process. If the difference affects the RPN_IOU negative threshold, some simulated ship objects may be recognized as background information. We will compare the differences between different RPN_IOU thresholds and analyze the influence on the simulated images.

In this part of the experiment, 100 inshore remote sensing ship images with 634 samples were collected. These 100 images were further processed with our method. The same 100 images were used for testing. The original RPN_IOU positive threshold and the RPN_IOU negative threshold are 0.7 and 0.2, respectively. The results of keeping the RPN_IOU negative threshold at 0.2 and changing the RPN_IOU positive threshold from 0.5 to 0.9 are shown in Table 3 and Figure 10. We considered the difference between the simulated ship objects and the real ship objects to be small and have no influence on the RPN network if no sudden change could be observed in the test results.

Table 3. The result with the different region proposal network Intersection over Union (RPN_IOU) positive thresholds.

RPN_IOU Positive Threshold	Precision	Recall	F-Measure
0.5	82.30%	74.81%	78.69%
0.6	80.42%	78.97%	79.69%
0.7	80.23%	75.87%	77.99%
0.8	80.99%	**80.08%**	**80.54%**
0.9	80.82%	78.74%	79.77%

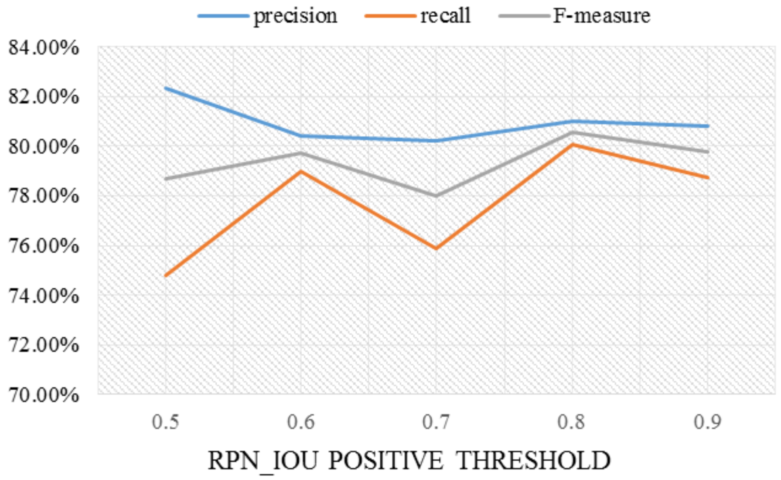


Figure 10. The results with the different RPN_IOU positive thresholds.

For our data, it can be seen from the experimental results that the change in the RPN_IOU positive threshold from 0.5 to 0.9 had little impact on the experimental results. It can be considered that these simulated ship objects are the same as real ship objects; i.e., simulated ship objects can be applied to replace the ships in the real images. The best results were obtained with an RPN_IOU positive threshold of 0.8. Table 4 and Figure 11 show the results of keeping the RPN_IOU positive threshold at 0.8 and changing the RPN_IOU negative threshold from 0.2 to 0.5.

Table 4. The results with the different RPN_IOU negative thresholds.

RPN_IOU Negative Threshold	Precision	Recall	F-Measure
0.2	**80.99%**	80.08%	**80.54%**
0.3	75.42%	81.19%	77.66%
0.4	60.75%	86.72%	71.45%
0.5	47.61%	**89.86%**	62.24%

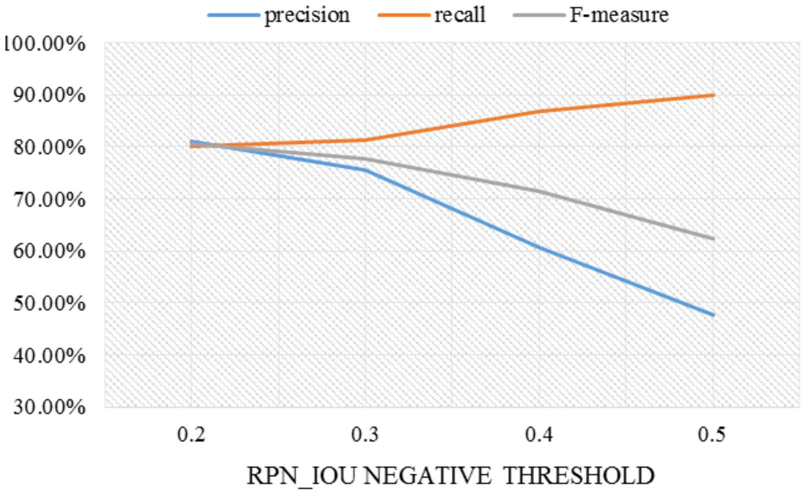


Figure 11. The results with the different RPN_IOU negative thresholds.

When the RPN_IOU negative threshold changes from 0.2 to 0.5, the precision gradually decreases and the recall gradually increases, and the F-measure also decreases. The best results are obtained when the RPN_IOU negative threshold is 0.2. We can conclude that the simulated ship objects are not recognized as background information.

4. Conclusions

In this paper, we proposed a training strategy of making simulated images to augment the data and reduce manual annotation. In the proposed method, the ship models are projected onto real images to generate simulated images and improve the quality of the dataset. The experimental results verified that the simulated images were realistic and suitable for dataset extension. At the same time, our method can be combined with traditional data enhancement methods, which greatly increase the number of samples. The experiments show that the ship detection results have been improved with our data augmentation method.

However, some problems are still need to be improved. We use only a few types of ship models and a small number of ship models. The selection of projection positions of ship models needs further improvement to avoid the occurrence of ship–land overlap. The color of the ship's three-dimensional model also needs to be improved. In the future, we will apply textures to the model based on the corresponding pattern of the real ship, and not just color.

Author Contributions: Conceptualization, Yiming Yan and Nan Su; Methodology, Yiming Yan and Nan Su; Software, Zhichao Tan and Nan Su; Writing—original draft, Zhichao Tan; Writing—review & editing, Yiming Yan and Zhichao Tan.

Funding: This research was supported by the Fund of the National Natural Science Foundation of China under Grant No. 61801142, No. 61601135, and No. 61675051, the Natural Science Foundation of Heilongjiang Province of China under Grant No. QC201706802, the China Postdoctoral Science Foundation No. 2018M631912, the Postdoctoral Science Foundation of Heilongjiang Province of Chain No LBH-Z18060, and the Fundamental Research Funds for the Central Universities (No. 3072019CFM0801).

Acknowledgments: We would like to acknowledge the reviewers for their helpful comments and the editors for the editing assistance. We also thank Ruochen Zhao for his suggestions to the language editing.

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

References

1. Lin, H.; Shi, Z.; Zou, Z. Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *99*, 1–5. [CrossRef]
2. Zou, Z.; Shi, Z. Ship detection in spaceborne optical image with SVD networks. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 5832–5845. [CrossRef]

3. Li, Q.; Mou, L.; Liu, Q.; Wang, Y.; Zhu, X.X. HSF-Net: Multiscale deep feature embedding for ship detection in optical remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, 1–15. [CrossRef]
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 1904–1916. [CrossRef] [PubMed]
6. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 1440–1448.
7. 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]
8. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. you only look once: Unified, real-time object detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 779–788.
9. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Fu, C.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
10. Liu, Z.; Hu, J.; Weng, L.; Yang, Y. Rotated region based CNN for ship detection. In Proceeding of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 900–904.
11. Yang, X.; Sun, H.; Fu, K.; Yang, J.; Sun, X.; Yan, M.; Guo, Z. Automatic ship detection of remote sensing images from Google Earth in complex scenes based on multi-scale rotation dense feature pyramid networks. *Remote Sens.* **2018**, *10*, 132. [CrossRef]
12. Taylor, L.; Nitschke, G. Improving deep learning with generic data augmentation. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1542–1547.
13. Dieleman, S.; Willett, K.W.; Dambre, J. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Not. R. Astron. Soc.* **2015**, *450*, 1441–1459. [CrossRef]
14. Bae, H.J.; Kim, C.W.; Kim, N.; Park, B.; Kim, N.; Seo, J.B.; Lee, S.M. A perlin noise-based augmentation strategy for deep learning with small data samples of HRCT images. *Sci. Rep.* **2018**, *8*, 17687. [CrossRef] [PubMed]
15. Fawzi, A.; Samulowitz, H.; Turaga, D.; Frossard, P. Adaptive data augmentation for image classification. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3688–3692.
16. Qiu, Y.M.; Qin, X.L.; Zhang, J. Low effectiveness of non-geometric-operation data augmentations for lesion segmentation with fully convolution networks. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 299–304.
17. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2014; pp. 2672–2680.
18. Tang, B.; Tu, Y.; Zhang, S.; Lin, Y. Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access* **2018**, *6*, 15713–15722. [CrossRef]
19. Yang, J.; Zhao, Z.; Zhang, H.; Shi, Y. Data augmentation for X-ray prohibited item images using generative adversarial networks. *IEEE Access* **2019**, *7*, 28894–28902. [CrossRef]
20. 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 2016), Las Vegas, NV, USA, 26 June–1 July; pp. 770–778.