Research Article

An Intelligent Ship Image/Video Detection and Classification Method with Improved Regressive Deep Convolutional Neural Network

Zhijian Huang^{1,2}, Bowen Sui¹, Jiayi Wen¹, and Guohe Jiang¹

¹Lab of Intelligent Control and Computation, Shanghai Maritime University, Shanghai 201306, China

²Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881, USA

Correspondence should be addressed to Zhijian Huang; zjhuang@shmtu.edu.cn and Guohe Jiang; ghjiang@shmtu.edu.cn

Received 22 December 2019; Revised 6 March 2020; Accepted 12 March 2020; Published 9 April 2020

Academic Editor: Átila Bueno

Copyright © 2020 Zhijian Huang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The shipping industry is developing towards intelligence rapidly. An accurate and fast method for ship image/video detection and classification is of great significance for not only the port management, but also the safe driving of Unmanned Surface Vehicle (USV). Thus, this paper makes a self-built dataset for the ship image/video detection and classification, and its method based on an improved regressive deep convolutional neural network is presented. This method promotes the regressive convolutional neural network from four aspects. First, the feature extraction layer is lightweighted by referring to YOLOv2. Second, a new feature pyramid network layer is designed by improving its structure in YOLOv3. Third, a proper frame and scale suitable for ships are designed with a clustering algorithm to reduced 60% anchors. Last, the activation function is verified and optimized. Then, the detecting experiment on 7 types of ships shows that the proposed method has advantage compared with the YOLO series networks and other intelligent methods. This method can solve the problem of low recognition rate and real-time performance for ship image/video detection and classification with a small dataset. On the testing-set, the final *mAP* is 0.9209, the Recall is 0.9818, the *AIOU* is 0.7991, and the FPS is 78–80 in video detection. Thus, this method provides a highly accurate and real-time ship detection method for the intelligent port management and visual processing of the USV. In addition, the proposed regressive deep convolutional network also has a better comprehensive performance than that of YOLOv2/v3.

1. Introduction

In the age of artificial intelligence, the shipping industry is developing towards intelligence rapidly. The ship image/video detection and classification with the help of computer vision have been applied in the port supervision service and Unmanned Surface Vehicle (USV) technology. An accurate and rapid detection method is of great significance to not only the port management, but also the safe operation of the USV.

The traditional methods of ship detection and classification are as the following two: (1) the method based on the structure and shape characteristics of ships. In 2012, Feflatyev et al. presented a novel algorithm for the open-sea.

The ship detection precision of 88% is achieved on a large dataset collected from a prototype system [1]. In 2013, Chen et al. improved an RCS density-coding method when acquiring ship features and completed the ship identification task with a high-resolution Synthetic aperture radar (SAR) dataset [2]. The accuracy of this method reached 91.54%. In 2016, Yüksel et al. extracted ship features from the contour image of a 3D ship model, and extracted ship features from optical image for ship recognition [3]. Also in 2016, Li et al. proposed a novel method for the inshore ship detection via the ship head classification and body boundary determination [4]. In 2017, Zhang et al. developed a new ship target-detection algorithm of visual maritime surveillance. The three main steps, including the horizon detection,

background modeling, and background subtraction, are all based on the discrete cosine transform [5]. (2). The method based on threshold. It is usually very practical to detect ships directly with the threshold method. In 1996, Eldhuset proposed a method based on the local threshold, which takes the ship out of the background and uses filtering window method in detection [6]. In 1999, Zhou et al. designed a global threshold algorithm which can complete the adaptive calculation and ship detection using the statistical characteristics of dataset images, that is, the adaptive threshold method [7]. In 2013, Rey used statistical data to solve feature when calculating the overall threshold value of ship images, which is a method based on the probability density function to detect ships on water [8]. In 2018, Li and Li proposed a method based on the high and low thresholds to detect ship edge feature and achieved a high accuracy of ship edge detection [9].

Although the above studies have achieved good results, the traditional methods are mostly based on the ship structure and shape for manual feature design. Even if the best nonlinear classifier is used to classify these manually designed features, the accuracy of ship detection cannot meet the practical needs. Therefore, these methods cannot achieve good results in the case of complex background and small hull differences in a real environment, and the recognition rate of multiple-ship classification is also not ideal.

Fortunately, after a development of more than ten years, the target detection based on the deep Convolutional Neural Network (CNN) has made a great progress in the application of human face, pedestrian, and other scenes. The CNN was first proposed by professor LeCun from Toronto university in Canada. The depth and width of the CNN have been continuously increased, and its accuracy for image recognition has also been continuously increased. The commonly used CNN includes the Lenet-5 [10], AlexNet [11], VGG [12], GoogLeNet [13], ResNet [14], and DenseNet [15]. At the same time, there are some researches in the application of the deep CNN for ship recognition and detection. The deep convolutional network for target detection can be divided into two categories: (1) the region-based methods, such as the R-CNN [16], Fast-RCNN [17], and Faster-RCNN [18]; (2) the regression-based methods, such as the SSD [19], YOLO [20], YOLOv2 [21], and YOLOv3 [22]. The regression-based deep convolutional network uses the CNN as a regression and returns the position information of the target in the image through an end-to-end training and gets the final bounding box and classification results.

In 2017, Kang et al. presented a contextual region-based CNN with multilayer fusion for SAR ship detection [23]. In 2018, Wang et al. proposed a ship detection algorithm combining the CFAR and CNN. This algorithm is more accurate and faster in the remote-sensing ocean satellite-image with complex distribution [24]. In 2018, Li et al. developed a HSF-Net. This net finds the multiscale deep feature embedding for ship detection in optical remote-sensing imagery [25]. Also in 2018, Yang et al. proposed an

automatic ship detection of remote-sensing images from Google Earth based on multiscale rotation dense feature pyramid networks [26]. In 2019, Gao et al. applied the Faster R-CNN to detect ships without the need for land masking by incorporating a large number of images containing only terrestrial regions as negative samples without any manual marking [27]. Also in 2019, Lin et al. proposed a squeeze and excitation rank Faster R-CNN for ship detection in SAR images, which shows a much better detection effect and speed than the traditional state-of-the-art methods [28].

The above detection methods which are mainly based on remote sensing or radar images hardly meet real-time requirement due to timeliness of image acquisition. Thus, in 2016, Zhao et al. proposed a real-time algorithm based on the deep CNN and combined with the HOG and HSV algorithms to achieve a good ship identification effect [29]. In 2017, Yang et al. used the Faster R-CNN to achieve the video detection of river vessels [30]. In 2018, Shao et al. built a new large-scale dataset of ships, which is designed for training and evaluating ship object detection algorithms. The dataset currently consists of 31455 images and covers six common ship types [31]. In 2019, Shao et al. proposed to use visual images captured by an on-land surveillance camera network to achieve real-time detection based on a saliency-aware CNN framework [32].

However, with the improvement of the accuracy and real-time requirements of ship detection and classification in the practical application, it is necessary to propose a ship image/video detection and classification method based on an improved regressive deep convolution network. Thus, this paper makes a self-built dataset for 7 kinds of ship image/video detection and classification, and its method based on an improved regressive deep CNN is presented. This method promotes the regressive CNN from four aspects. First, the feature extraction layer is lightweighted by referring to YOLOv2. Second, a new Feature Pyramid Network (FPN) layer is designed by improving its network structure in YOLOv3. Third, a proper frame and scale suitable for the ships are designed with the clustering algorithm to reduce 60% anchors. Last, the optimal activation function is verified and optimized. Then, this method can solve the problem of low recognition rate and real-time performance for ship image/video detection and classification through an end-to-end training. The experiment on 7 types of ships shows that the proposed method is better in ship image/video detection and classification compared with the YOLO series network and other intelligent methods. On the testing-set, the final *mAP* is 0.9209, the *Recall* is 0.9818, the *AIOU* is 0.7991, and the *FPS* is 78–80 in video detection, which takes into account both the accuracy and real-time performance for the ship detection. Thus, this method provides a highly accurate and real-time ship detection method for the intelligent port management and visual processing of the USV. In addition, this paper also proposes a regressive deep convolutional network with a better comprehensive performance than YOLOv2 and YOLOv3.

2. The Regressive Deep Convolutional Neural Network (RDCNN)

The basic structure of the regressive deep CNN is mainly consisted of the input layer, convolution layer, pooling layer, full-connection layer, and output layer.

2.1. The Input Layer. The function of the input layer is to receive input image and store it in matrix form. Assuming that the regressive deep CNN has a structure of L layer, then x^l represents the feature of No. l layer, $l = 1, 2, \dots, L$. In it, x^l is composed of multiple feature graphs, which can be represented as $x^l = \{x_1^l, \dots, x_j^l\}$, j is the number of the feature graphs in l layer. Thus, the corresponding feature of a color input image can be represented as $x^1 = \{x_1^1, x_2^1, x_3^1\}$, where x_1^1 , x_2^1 and x_3^1 represents the data of red, green, and blue channels, respectively.

2.2. The Convolutional Layer. The function of the convolution layer is to extract features through convolution operation. With a proper design, the feature expression ability of the regressive deep CNN will be strengthened with the increasing of convolution layers. The feature graph of No. l convolution layer can be calculated as

$$x_j^l = f\left(\sum_{i=1}^{j-1} G_{i,j}^l (k_{i,j}^l \otimes x_i^{l-1}) + b_j^l\right), \quad (1)$$

where $k_{i,j}^l$ and b_j^l are the weights of the convolution kernel and biases of the convolution layer, respectively; $G_{i,j}^l$ is the connection matrix between No. l convolution layer and the feature graph of the previous $l-1$ convolution layer; the symbol $\otimes$ represents the convolution operation; and $f(x)$ is the activation function. When $G_{i,j}^l$ is 1, x_i^{l-1} is associated with x_j^l ; when $G_{i,j}^l$ is 0, they are no correlations.

2.3. The Pooling Layer. The function of the pooling layer is to reduce the feature dimension. The pooling layer is generally located behind the convolutional layer, and the pooling operation can maintain a certain spatial invariance. The feature graph x_j^l of the pooling operation in the l layer can be calculated as

$$x_j^l = p(x_j^{l-1}), \quad (2)$$

where $p(x)$ represents the pooling operation.

2.4. The Fully Connected Layer. The function of the fully connected layer is to transform the deep feature obtained in the front layers into a feature vector. Thus, this layer is usually set behind the feature extraction layer. The feature vector x^l in the fully connected layer can be calculated as

$$x^l = f(w^l x^{l-1} + b^l), \quad (3)$$

where w^l is the connecting weight between two adjacent network layers and b^l is the offset and $f(x)$ is the activation function.

2.5. The Loss Function. The regressive deep CNN obtains the predicted value through a forward propagation. Then, the error between the predicted value and real value is usually calculated with the following cross-entropy loss function:

$$\text{Loss} = -\frac{1}{n} \sum_x [y \ln \tilde{y} + (1 - y) \ln (1 - \tilde{y})], \quad (4)$$

where x are the input samples, y is the predicted output, $\tilde{y}$ is the actual output, and n represents the total number of the input samples in one batch.

2.6. The Network Performance Index. For the regressive deep CNN, the IOU represents the overlap rate between the detection window (B_{gt}) generated by the network model and the actually marked window (B_{dt}), that is, the ratio of their intersection and union areas. area($\cdot$) means the area, and the IOU can be calculated as

$$\text{IOU} = \frac{\text{area}(B_{dt} \cap B_{gt})}{\text{area}(B_{dt} \cup B_{gt})}. \quad (5)$$

For the experiment of this paper, the detection result of $\text{IOU} \geq 0.5$ is set as a real positive sample, and the detection result of $\text{IOU} \leq 0.5$ is set as a false-negative sample.

As there are many kinds of targets detected in this paper, the AIOU (the average value of IOU) is used, that is, the average ratio between the intersection and union areas of the predicted and actual boundary boxes on the testing-set, which is denoted as

$$\text{AIOU} = \frac{1}{n} \sum_{i=0}^{n-1} \text{IOU}, \quad (6)$$

where n represents the number of detected targets.

The Recall (R) rate is used to represent the percentage of the positive samples in the samples that are correctly predicted:

$$\text{Recall} = \frac{t_p}{t_p + f_n}, \quad (7)$$

where t_p represents a true positive sample, and f_n represents a false-negative sample.

The Precision (P) indicates how many samples of the positive prediction are truly positive samples:

$$\text{Precision} = \frac{t_p}{t_p + f_p}, \quad (8)$$

where f_p represents a false positive sample.

The AP is an index used to measure the network identification accuracy, which is generally represented by the area enclosed by the Recall rate and Precision curves. Assuming that the curve of the recall rate and precision rate is PR, then

$$\text{AP} = \int_0^1 \text{PR} \, dr. \quad (9)$$

As there are 7 targets detected in this paper, the mAP is used to represent the network identification accuracy, that is, the average value of AP :

$$mAP = \frac{\left(\int_0^1 PR dr \right)}{N}, \quad (10)$$

where n represents the number of the predicted categories, that is 7.

In addition, in order to measure the network speed for video detection, the frames per second (FPS) is also used as a performance index.

3. The Improved RDCNN Based on YOLOv2/v3

This research presents an improved RDCNN mainly based on the YOLO series, which also refers to the advantages of the current popular regression deep convolution networks. By promoting the feature extraction layer of YOLOv2 and the FPN of YOLOv3, the improved network overcomes the detection shortcomings of YOLOv2 and the training and recognition speed shortcomings of YOLOv3. The improved network also redesigns the anchors with the clustering algorithm and optimizes the effects of the activation function both according to the ship image/video detection and classification. Finally, this algorithm achieves a good accuracy and real-time performance in the ship image/video detection and classification.

The improved network structure built in this research is shown in Figure 1. This network structure mainly consists of three parts: the feature extraction layer, FPN layer, and prediction layer, which are specifically described below.

3.1. The Lightweighted Feature Extraction Layer. The feature extraction layer is very important in building the network structure. If the feature extraction layer is too large, it may get better deep features, but it will also slow down the speed of the whole network. For example, in YOLOv3, the darknet-53 is used as the feature extraction layer. This extraction layer is relatively slow in training and detection speed due to the deep layer numbers. In order to improve the presented network with a lightweight feature extraction layer, first, this network adopts the Darknet-19 feature extraction layer of YOLOv2, and the structure is shown in the left of Figure 1. This feature extraction layer has the advantage of relatively few network layers and faster calculation speed and can also extract deep features well when inputting a color ship image or video of $416 \times 416 \times 3$ size.

In addition, with the increase of the feature extraction layer numbers, the network generally can obtain deeper features with a more expressive power. However, simply increasing the number of network layers will result in a gradient dispersion or explosion phenomena. In order to solve this problem, in the later experiment, a batch normalization strategy is added between the convolution (Conv2d) and activation (Leaky-Relu) of each convolution operation in the Darknet-19 feature extraction layer, which

is shown in Figure 2. This strategy can effectively control the gradient problem caused by the network deepening.

3.2. The New FPN Layer with a Clustering Algorithm. For the feature extraction, the feature information in shallow layer is relatively small, but its location is accurate. This has the advantage for predicting small objects. On the contrary, the feature information in deep layer is rich, but its location is relatively rough. This is suitable for predicting large objects.

Thus, in order to make the network obtain a better detection result, the improved network promotes the multiscale prediction idea of YOLOv3 to design a new FPN layer, which is shown in the right of Figure 1. This method up samples the deep feature map into 26×26 size after predicting a deep feature map of 13×13 size from the feature extraction layer and then merges the upsampled 26×26 feature map with the shallow 26×26 feature map. Finally, the network can detect and forecast the input image at two scales.

In addition, to get a better network structure, the clustering algorithm is also used, and the effect of the collected data is fine-tuned and optimized for the ship image/video detection and classification. Finally, the obtained anchor values are shown in Table 1, which predicts the feature maps of 13×13 and 26×26 scales, setting 5 different anchor frames on each scale.

Therefore, for a 416×416 size image, the improved network predicts a total of 4225 fixed prediction frames, compared with YOLOv3, which has 9 anchor frames on 3 scales and 10647 fixed prediction frames in total. Obviously, the number of anchor frames in the improved network is reduced by 6422, that is, about 60%.

3.3. The Prediction Layer. Through the prediction on the convolution layer, the spatial information can be well preserved. For the improved network, the prediction method of YOLOv2 is adopted in the prediction layer. Each predicting frame predicts 7 ship categories and 5 frame information (t_x, t_y, t_w, t_h, t_o), of which the first four parameters are the detecting object coordinates and t_o is the predicting confidence. In this paper, the loss function of YOLOv2 is also used in the prediction layer.

3.4. The Optimization of the Activation Function for the Improved RDCNN. In order to optimize the influence of the activation function, combined with the network structure proposed in this paper, the *ELU* and *Leak-Relu* activation functions of equations (11) and (12) are also used and tested except for the commonly used *Relu*.

$$ELU(x) = \begin{cases} e^x - 1, & \text{if } x \leq 0, \\ x, & \text{if } x > 0, \end{cases} \quad (11)$$

$$Leak Relu(x) = \begin{cases} 0.1x, & \text{if } x \leq 0, \\ x, & \text{if } x > 0. \end{cases} \quad (12)$$

FIGURE 1: The network structure of the improved RDCNN by promoting YOLOv2 and v3.

FIGURE 2: The addition of batch normalization into the Darknet-19 and other convolution layers.

TABLE 1: The anchor values of different scales proper for the ship image/video detection and classification.

ID	13 × 13	26 × 26
1	(1.34221, 1.52115)	(1.34221, 1.52115)
2	(3.42217, 4.30406)	(3.42217, 4.30406)
3	(5.05587, 8.09892)	(5.05587, 8.09892)
4	(9.76561, 4.30316)	(9.76561, 4.30316)
5	(11.52521, 11.1835)	(11.52521, 11.1835)

Through the experimental comparison, the activation function with the best ship image/video detection and classification effect can be optimized. The results on the testing-set are obtained, which is shown in Table 2.

In the experiment, the *Leaky-Relu* activation function has the best comprehensive detection effect and is less operable than the *Relu* and *ELU* activation functions. Thus, the *Leaky-Relu* is selected as the optimized activation function.

4. The Making of Ship Dataset and Experimental Environment

4.1. The Making of Ship Dataset. At present, the popular target-detection datasets are VOC and COCO, but these datasets classify ships as only one kind. In a specific application, it often needs to classify ships more precisely. Therefore, in this research, the dataset of ship images is built after collecting and labeling by ourselves.

The main way to collect the ship images is the Internet. As the images are found from the Internet, the pixels resolution are different, and the size of the images are also different, such as $500 \times 400 \times 3$ and $500 \times 318 \times 3$. The images containing the ships are cut roughly according to the length to width ratio of 1:1. The scale of ship proportion to the

TABLE 2: The experiment effects of different activation functions.

Activation function	AIIOU	mAP	Recall
Relu	0.7247	0.7932	0.9492
ELU	0.7306	0.8005	0.9559
Leaky-Relu	0.7329	0.7987	0.9616

whole image in each image is also different, even very different, which can be seen from Figures 3–5 of the database images or the detected images. These naturally produced images of different specification and quality are more conducive to the training effect and generalization ability. Before training, they were all resized to $416 \times 416 \times 3$ size images.

After the dataset is collected, it needs to be labeled before using as the network input. The labeling tool used in this paper is LabelIMG. In the LabelIMG, the target object can be selected in the image with a rectangle box and be saved with a label. Then, a file with the suffix of xml can be got. This file contains the path, name, resolution of the original image, as well as the coordinates, and name information of the target object in the image.

There are many types of ships in real application. In order to facilitate research and save costs, this paper only collects 7 representative types of ships: the sailing ship, container ship, yacht, cruise ship, ferry, coast guard ship, and fishing boat. After filtering and classification, the final dataset size is 4200 manual-selected images, which includes 600 images in each category. The 480 images in each category are randomly selected as the training-set, and each remaining 120 images are set as the testing-set. In this way, the total size of the training-set is 3360 images, and the total size of the testing-set is 840 images. The typical images of each category in the dataset are shown in Figure 3.

4.2. The Experimental Environment Configuration. The experimental environment of this research is configured as follows. The CPU: Intel i7-7700 with 4.2 GHz main frequency; the memory: 16G; the GPU: two of Nvidia GTX1080 Ti; the operating system: Ubuntu 16.04. In order to make full use of the GPU to accelerate the network training, the

FIGURE 3: The dataset image demonstration: the coast guard ship, container ship, cruise ship, ferry, fishing boat, sailboat, and yacht.

FIGURE 4: The representative detection results of the improved RDCNN network method.

FIGURE 5: The comparison of detection effect among the method of YOLOv2 (a), YOLOv3 (b), and the improved RDCNN (c).

CUDA 9.0 and its matching CUDNN are installed in the system. In addition, the OpenCV3.4 is also installed in the environment to display the results of the network detection and classification.

During the experiment, the setting of the experimental parameters is very important. There are many parameters to be set in our improved RDCNN and YOLOv2/v3, such as the batch number, down sampling size, momentum parameter, and learning rate. The setting of these parameters will affect not only the normal operation of the network, but also the training effect. For example, when the setting number of the batch is too large, the network will not run if the memory of the workstation is not big enough.

Considering the conditions of our experimental environment, and also for comparing convenience, the same parameters are set for the improved RDCNN and YOLOv2/v3. The network parameters are set as follows: the number of small batch is 64 and divided into 8 sub-batches, the iteration number is 8000, the momentum parameter is 0.9, the weight attenuation is 0.0005, and the learning rate is 0.001, which are shown in the following Table 3:

5. The Training and Detection Based on YOLOv2/v3

5.1. The Iterative Convergence Training. Generally, whether a network meets the training requirements is judged by the

convergence of the loss function. In this experiment, due to the small size of the dataset and sufficient computing ability, the convergence with only 8000 times of iterations is achieved, which takes about only 1 hour and 40 minutes. The *Loss* and *AIU* curves of the feedforward training process are shown in Figures 6 and 7, respectively. It can be seen from Figures 6 and 7 that the training has converged steadily when the number of the network training reaches 8000 times.

The training time of YOLOv3 is relatively long, and it takes about 3 hours and 40 minutes for 8000 times of iteration convergence process. The *Loss* and *AIU* curves of the feedforward training process are shown in Figures 8 and 9, respectively. It can also be seen from Figures 8 and 9 that after 8000 times of iterative training, the *Loss* and *AIU* of the network also have converged steadily.

Finally, the weight parameters obtained through 8000 network iterations in the feedforward training are saved in the experiment.

5.2. The Detection Performance Testing. After the network training is stable, it is necessary to verify its detection effect on the testing-set, especially to avoid a decline of the detection effect caused by overfitting. First, the network indexes obtained with the weights of No. 8000 iteration under the testing-set are taken as the evaluation criteria. The specific values are shown in Table 4.

TABLE 3: The same training parameters for our improved RDCNN and YOLOv2/v3.

Batch number	Down sampling size	Image size	Momentum parameter	Weight attenuation	Learning rate	Iteration number
64	8	$416 \times 416 \times 3$	0.9	0.0005	0.001	8000

FIGURE 6: The iterative convergence process of the *Loss* curve for YOLOv2 network training. FIGURE 9: The iterative convergence process of the *AIOU* curve for YOLOv3 network training. FIGURE 7: The iterative convergence process of the *AIOU* curve for YOLOv2 network training. FIGURE 8: The iterative convergence process of the *Loss* curve for YOLOv3 network training.

As the network cannot measure its weight parameters in real time under the training-set during its feedforward running, the network parameters generated in the Nos. 400, 600, 800, 1000, 2000, 3000, 4000, 5000, 6000, 7000, and 8000 training iterations are also taken here to load into the network for a later test and verification. In order to better analyze the detection effect of YOLOv2/v3 in this task, the *AIOU* and *mAP* parameters of the network are compared in different testing iterations under the testing-set, which are shown in Figures 10 and 11.

From the *AIOU* and *mAP* curves on the testing-set, it can be seen that the performance indexes of the network on the testing-set have been stable. There is also no overfitting

TABLE 4: The comparison of performance indexes between YOLOv2/v3 after 8000th iteration.

Network	AIOU	mAP	Recall	FPS	Training time
YOLOv2	0.7838	0.9165	0.9425	94–95	1 h 40 min
YOLOv3	0.7895	0.9280	0.9674	45–46	3 h 40 min

phenomenon caused by too many training times. Through comparison, we can see that YOLOv3, as an improved version of YOLOv2, has advantages in the *AIOU* and *mAP* performance indexes. That is, it has 0.0057 higher in the *AIOU* and 0.0115 higher in the *mAP* than that of YOLOv2. However, as the advantages of YOLOv3 are obtained by deepening and improving its network structure, its detection speed is 49 FPS lower than that of YOLOv2.

6. The Experiment and Analysis of the Improved RDCNN

6.1. The Network Performance Experiment. The improved RDCNN takes 20 more minutes to complete the 8000 training iterations convergence process compared with YOLOv2. However, the training time is much lower than that of YOLOv3. The *Loss* and *AIOU* curves of the feed-forward training process are shown in Figures 12 and 13, respectively. It can also be seen from Figures 12 and 13 that after 8000 times of iterative training, the *Loss* and *AIOU* of the improved network have been converged steadily.

In order to verify the detection effect of the RDCNN on the testing-set, the network weight parameters generated in the Nos. 400, 600, 800, 1000, 2000, 3000, 4000, 5000, 6000, 7000, and 8000 training iterations are taken here to load into the improved network for a later test and verification. Then, the *AIOU* and *mAP* parameter curves under the testing-set are tested in different testing iterations of the network, which are shown in Figures 14 and 15. This paper applies the two editions of YOLO networks, as well as the presented improved RDCNN based on YOLO, into the ship image/video detection and classification. Thus, the comparing

FIGURE 10: The comparison of $AIOU$ index between YOLOv2/v3 under the testing-set.

FIGURE 11: The comparison of mAP index between YOLOv2/v3 under the testing-set.

performance of the $AIOU$ and mAP are also show in Figures 14 and 15 for YOLOv2/v3 and the improved RDCNN network structure. The comparison of the evaluation indexes of each network is also shown in Figure 16.

According to the comparisons, it can be seen that the improved RDCNN network has surpassed YOLOv2 and YOLOv3 in the $AIOU$ detection of positioning accuracy. That is, it is 0.0153 higher than that of YOLOv2 and 0.0096 higher than that of YOLOv3, respectively, in $AIOU$. In addition, the improved network is 0.0044 higher than YOLOv2 in the mAP index. Due to the simplified network structure, the mAP index of the improved network is 0.0071 lower than that of YOLOv3, but the detecting FPS index is 33 higher than that of YOLOv3. Therefore, it can be concluded that the overall effect of the improved network is better than that of YOLOv2/v3 in the collected dataset of this experiment.

FIGURE 12: The iterative convergence process of the $Loss$ curve for the improved RDCNN training.

FIGURE 13: The iterative convergence process of the $AIOU$ curve for the improved RDCNN training.

FIGURE 14: The comparison of $AIOU$ among YOLOv2/v3 and the improved RDCNN under the testing-set.

Therefore, the experimental results show that the improved RDCNN network structure designed in this paper surpasses the two YOLO networks in three evaluation indexes.

6.2. The Effect Demonstration of the Improved Network. For the testing-set, the representative detection results of the improved RDCNN network are shown in Figure 15. In order to achieve a better network effect, the weight parameters of the feature extraction layer extracted in the ImageNet [33] pretraining are loaded to train the improved RDCNN of this

FIGURE 15: The comparison of mAP among YOLOv2/v3 and the improved RDCNN under the testing-set.

FIGURE 16: The comparison of the network performance indexes among YOLOv2/v3 and the improved RDCNN.

TABLE 5: The loading test results of the pretraining weights for the improved RDCNN.

Network	AIOU	mAP	Recall	FPS	Training time
Proposed method	0.7991	0.9209	0.9818	78–80	2 h 0 min

paper. Through the test on the testing-set and the video, the final results are shown in Table 5.

It can be seen that the mAP index of the improved RDCNN is slightly lower than that of YOLOv3 when using the pretraining weights. However, the other indicators are all better than that of YOLOv3, especially in the video detection speed of FPS .

In order to better display the comparison of the network effects, the YOLOv2/v3 and improved RDCNN are used to detect a image with multiple fishing boats. The representative results of the detection effect of the three networks are shown in Figure 16. In this paper, the improved network accurately detects more ships. Obviously, the presented

network in this paper has achieved a better result, which fully proves the effectiveness of the improved RDCNN network.

6.3. Comparison with Other Intelligent Detection and Classification Methods. The proposed method is also compared with other intelligent methods, such as Fast R-CNN, Faster R-CNN, and SSD, or compared with YOLOv2 under different dataset image and hardware configuration. The work in the early published IEEE Trans paper [32] is very similar to this paper, then its experiment results can be used for the comparison. The comparing results are shown in Table 6. The proposed method has advantage over other intelligent methods in precision and

TABLE 6: The comparison with other intelligent detection and classification methods.

Methods	Ship category	Dataset image	Hardware Configuration	IOU threshold	mAP	FPS
Fast R-CNN (VGG)	6	11126	Four titan xp	>0.5	0.710	0.5
Faster R-CNN (ZF)	6	11126	Four titan xp	>0.5	0.892	15
Faster R-CNN (VGG)	6	11126	Four titan xp	>0.5	0.901	6
SSD	6	11126	Four titan xp	>0.5	0.794	7
YOLOv2	6	11126	Four titan xp	>0.5	0.830	83
Shao's method	6	11126	Four titan xp	>0.5	0.874	49
Proposed method	7	4200	Two GTX1080 Ti	>0.5	0.9209	78–80

speed, that is, mAP and FPS, and it can also satisfy the detection and classification requirement in video scene. However, our dataset size is smaller than that of Shao's work, and our hardware configuration is also weaker than that of Shao's work.

7. Discussion and Conclusions

In this paper, the improved RDCNN network is presented to achieve the ship image/video detection and classification task. This network does not need to extract features manually, which improves the regressive CNN from four aspects based on the advantages of the current popular regression deep convolution networks, especially YOLOv2/v3. Thus, this network only needs the dataset of the ship images and a successful training.

This paper makes a self-built dataset for the ship image/video detection and classification, and the method based on an improved regressive deep CNN is researched. The feature extraction layer is lightweighted. A new FPN layer is redesigned. A proper anchor frame and size suitable for the ships are redesigned, which reduces the number of anchors by 60% compared with YOLOv3. The activation function is also optimized with the Leaky-Relu. After a successful training, the method can complete the ship image detection task and can also be applied to the video detection. After 8000 times of iterative training, the *Loss* and *AIIOU* of the improved RDCNN network have been converged steadily.

The experiment on 7 types of ships shows that the proposed method is better in the ship image/video detection and classification compared with the YOLO series networks. The improved RDCNN network has surpassed YOLOv2/v3 in the *AIIOU* detection of positioning accuracy. That is, it's 0.0153 higher than that of YOLOv2 and 0.0096 higher than that of YOLOv3, respectively, in *AIIOU*. In addition, the improved network is 0.0044 higher than YOLOv2 in the *mAP* index. Due to the simplified network structure, the *mAP* index of the improved network is 0.0071 lower than that of YOLOv3, but the detecting *FPS* index is 33 higher than that of YOLOv3. Therefore, it can be concluded that the overall effect of the improved network is better than that of YOLOv2/v3 in the collected dataset of this experiment.

Then, this method can solve the problem of low recognition rate and real-time performance for ship image/video detection and classification. Thus, this method provides a highly accurate and real-time ship detection method for the intelligent port management and visual processing of the USV. In addition, the proposed regressive deep

convolutional network also has a better comprehensive performance than YOLOv2/v3.

The proposed method is also compared with Fast R-CNN, Faster R-CNN, SSD, or YOLOv2 etc. under different datasets and hardware configurations. The results show that the method has advantage in precision and speed, and it can also satisfy the video scene. However, our dataset size is smaller. Thus, the detection in a much larger dataset can be the future work.

Data Availability

The [SELF-BUILT SHIP DATASET and SIMULATION] data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the NSFC Projects of China under Grant No. 61403250, No. 51779136, No. 51509151.

References

- [1] S. Fefilat'yev, D. Goldgof, M. Shreve, and C. Lembke, "Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system," *Ocean Engineering*, vol. 54, no. 1, pp. 1–12, 2012.
- [2] W. T. Chen, K. F. Ji, X. W. Xing, H. X. Zou, and H. Sun, "Ship recognition in high resolution SAR imagery based on feature selection," in *Proceedings of the IEEE International Conference on Computer Vision in Remote Sensing*, pp. 301–305, Xiamen, China, December 2013.
- [3] G. K. Yüksel, B. Yalıtuna, F. Tartar Ö, F. C. Adlı, K. Eker, and O. Yörük, "Ship recognition and classification using silhouettes extracted from optical images," in *Proceedings of the IEEE Signal Processing and Communication Application Conference*, pp. 1617–1620, Zonguldak, Turkey, May 2016.
- [4] S. Li, Z. Zhou, B. Wang, and F. Wu, "A novel inshore ship detection via ship head classification and body boundary determination," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 12, pp. 1920–1924, 2016.
- [5] Y. Zhang, Q.-Z. Li, and F.-N. Zang, "Ship detection for visual maritime surveillance from non-stationary platforms," *Ocean Engineering*, vol. 141, pp. 53–63, 2017.
- [6] K. Eldhuset, "An automatic ship and ship wake detection system for spaceborne SAR images in coastal regions," *IEEE*

- Transactions on Geoscience and Remote Sensing*, vol. 34, no. 4, pp. 1010–1019, 1996.
- [7] H. J. Zhou, X. Y. Li, X. H. Peng, Z. Z. Wang, and T. Bo, “Detect ship targets from satellite SAR imagery,” *Journal of National University of Defense Technology*, vol. 21, no. 1, pp. 67–70, 1999.
 - [8] M. T. Rey, A. Drosopoulos, and D. Petrovic, “A search procedure for ships in radarsat imagery,” Defence Research Establishment Ottawa, Ottawa, ON, Canada, Report No. 1305, 2013.
 - [9] X. Li and S. Li, “The ship edge feature detection based on high and low threshold for remote sensing image,” in *Proceedings of the 6th International Conference on Computer-Aided Design, Manufacturing, Modeling and Simulation*, Busan, South Korea, May 2018.
 - [10] L. C. Yann, B. Léon, B. Yoshua, and H. Patrick, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
 - [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proceedings of the International Conference on Neural Information Processing Systems*, Curran Associates Inc., Lake Tahoe, NV, USA, pp. 1097–1105, 2012.
 - [12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015, <https://arxiv.org/abs/1409.1556>.
 - [13] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” pp. 1–9, 2014, <https://arxiv.org/abs/1409.4842>.
 - [14] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, IEEE Computer Society, Las Vegas, NV, USA, June 2016.
 - [15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” pp. 2261–2269, 2017, <https://arxiv.org/abs/1608.06993>.
 - [16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” pp. 580–587, 2013, <https://arxiv.org/abs/1311.2524>.
 - [17] R. Girshick, “Fast R-CNN,” 2015, <https://arxiv.org/abs/1504.08083>.
 - [18] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.
 - [19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, “SSD: single shot MultiBox detector,” pp. 21–37, 2016, <https://arxiv.org/abs/1512.02325>.
 - [20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, IEEE, Las Vegas, NV, USA, June 2016.
 - [21] J. Redmon and A. Farhadi, “YOLO9000: better, faster, stronger,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6517–6525, IEEE, Honolulu, HI, USA, July 2017.
 - [22] J. Redmon and A. Farhadi, “YOLOv3: an incremental improvement,” 2018, <https://arxiv.org/abs/1804.02767>.
 - [23] M. Kang, K. Ji, X. Leng, and Z. Lin, “Contextual region-based convolutional neural network with multilayer fusion for sar ship detection,” *Remote Sensing*, vol. 9, no. 860, pp. 1–14, 2017.
 - [24] R. Wang, J. Li, Y. Duan, H. Cao, and Y. Zhao, “Study on the combined application of CFAR and deep learning in ship detection,” *Journal of the Indian Society of Remote Sensing*, vol. 4, pp. 1–9, 2018.
 - [25] Q. Li, L. Mou, Q. Liu, Y. Wang, and X. X. Zhu, “HSF-net: multiscale deep feature embedding for ship detection in optical remote sensing imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 12, pp. 7147–7161, 2018.
 - [26] X. Yang, H. Sun, K. Fu et al., “Automatic ship detection in remote sensing images from Google Earth of complex scenes based on multiscale rotation dense feature pyramid networks,” *Remote Sensing*, vol. 10, no. 1, pp. 132–145, 2018.
 - [27] L. Gao, Y. He, X. Sun, Xi Jia, and B. Zhang, “Incorporating negative sample training for ship detection based on deep learning,” *Sensors*, vol. 19, no. 684, pp. 1–20, 2019.
 - [28] Z. Lin, K. Ji, X. Leng, and G. Kuang, “Squeeze and excitation rank faster R-CNN for ship detection in SAR images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 751–755, 2019.
 - [29] L. Zhao, X. F. Wang, and Y. T. Yuan, “Research on ship recognition method based on deep convolutional neural network,” *Ship Science and Technology*, vol. 38, no. 8, pp. 119–123, 2016.
 - [30] M. Yang, Y. D. Ruan, L. K. Chen, P. Zhang, and Q. M. Chen, “New video recognition algorithms for inland river ships based on faster R-CNN,” *Journal of Beijing University of Posts and Telecommunications*, vol. 40, no. S1, pp. 130–134, 2017.
 - [31] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, “SeaShips: a large-scale precisely annotated dataset for ship detection,” *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.
 - [32] Z. Shao, L. Wang, Z. Wang, W. Du, and W. Wu, “Saliency-aware convolution neural network for ship detection in surveillance video,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 3, pp. 781–794, 2019.
 - [33] O. Russakovsky, J. Deng, H. Su et al., “ImageNet large scale visual recognition challenge,” *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.