




Research Article

An Improved Ship Classification Method Based on YOLOv7 Model with Attention Mechanism

Jian Cen ^{1,2} Hao Feng ^{1,2} Xi Liu ^{1,2} Yongjian Hu,³ Haoliang Li,⁴ Haisheng Li,^{1,2} and Weisheng Huang⁵

¹School of Automation, Guangdong Polytechnic Normal University, China

²Guangzhou Intelligent Building Equipment Information Integration and Control Key Laboratory, China

³School of Electronic and Information Engineering, South China University of Technology, China

⁴Department of Electrical Engineering, City University of Hong Kong, China

⁵Guangdong Xixun Intelligent Technology Co., Ltd., China

Correspondence should be addressed to Xi Liu; liuxi401402403@163.com

Received 2 November 2022; Revised 29 December 2022; Accepted 17 February 2023; Published 13 April 2023

Academic Editor: M. Hassaballah

Copyright © 2023 Jian Cen 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.

Deep learning (DL) is widely used in ship detection, but there are still some problems in the effective classification, such as inaccurate object feature extraction and inconspicuous feature information in deep layers. To address these problems, we propose a YOLOv7-residual convolutional block attention module (YOLOv7-RCBAM) by combining the convolutional attention mechanism and residual connections to the YOLOv7. First, to accelerate the training speed, the parameters in the backbone network of the pretrained model are frozen by using transfer learning, and the model is fine-tuned for training. Second, to enhance the information relevance of channel dimensional features, an attention mechanism with residual connectivity is adopted. Finally, a feature fusion attention mechanism is introduced to improve the effective feature extraction. The effectiveness of the proposed method is fully validated on the SeaShips dataset. The results show that the YOLOv7-RCBAM model achieves better performance with a 97.59% in mAP and effectively extracts object feature in deep layers. Meanwhile, the YOLOv7-RCBAM model can accurately locate ship in complex environments with darkness and noise with the mAP reaching 96.13% to achieve effective ship classification detection.

1. Introduction

With the development of image recognition technology, video surveillance has been applied in the field of maritime supervision and service. It plays a key role in tasks such as ship traffic flow statistics and ship collision prevention. Real-time detection and intelligent tracking of moving ship in complex environments are the significant basis for promoting the efficiency of maritime supervision. However, traditional methods generally have problems, such as slow training speed and weak interference resistance, which make it hard to detect and track moving ship with high accuracy [1, 2].

Recently, with the rapid development of DL in various fields [3–5], it has made significant breakthroughs in image

detection, gradually solving the problems of slow training speed and low detection accuracy of object detection. Detection algorithms based on DL are divided into two categories: (1) the two-stage algorithm based on the candidate area, that is, the selection of the candidate area, and then the positioning and classification of the object in the candidate area. The typical algorithms are R-CNN [6], Faster RCNN [7], R-FCN [8], Mask R-CNN [9], etc. (2) The one-stage algorithm based on regression classification combines the selection of candidate regions with positioning and classification to improve the training speed. The typical algorithms include SSD [10] and YOLO [11]. Compared with the two-stage algorithm, the one-stage algorithm fuses the detection steps of generating and optimizing the bounding box. It can accelerate the

training speed while maintaining stable detection accuracy. Therefore, we use the one-stage algorithm to achieve ship classification detection.

However, traditional detection algorithms still have many problems. It is difficult for the model to focus on the object information in the interference of complex environment. The object feature information is not obvious after being disturbed [12], leading to inaccurate positioning and classification. And deep feature extraction is easily led to the loss of feature information [13].

With the development of YOLOv7 [14], it has better advantages in faster speed and higher accuracy. It introduced reparameterized [15, 16] module that replaces the original module to reduce parameters and improve inference speed. And it introduced efficient long-range attention network [17] (ELAN) module instead of CSP module for the backbone network, which can enhance the extraction features and improve the use of parameters and calculations. The trainable bag-of-freebies [14] was proposed, so that the detection accuracy can be improved without increasing the inference cost. With the advantages of YOLOv7, it is more suitable for our method.

Based on analyzing the disadvantages of the YOLOv7, we introduce transfer learning [18], residual connection [19], convolutional block attention module (CBAM) [20], and feature fusion [21]. Transfer learning can freeze part of the model parameters to improve the training speed and extract richer features by fine-tuning. CBAM is capable of adaptively weighting feature values to enhance important features of the ship and restrain interference from the background. By combining a reasonable combination of residual connectivity and CBAM [22], multiple feature recalibrations are prevented from leading to reduced deep feature responses. Feature fusion is able to improve the representation ability of the object. Based on the above methods, YOLOv7-RCBAM is proposed, which effectively improves ship detection accuracy. Specifically, the contributions of this article are summarized as follows:

- (1) A convolutional attention mechanism block combined with residual connections is proposed. The improved method effectively extracts object feature information and focuses on foreground information
- (2) An improved YOLOv7 model based on double transfer learning is proposed, which introduces a feature fusion attention mechanism to improve the richness of feature extraction and solve the problem of the feature disappearance caused by too deep network
- (3) Image enhancement is performed to simulate ship detection in the dark night, rainy, and foggy environments. It can still verify that the method has strong interference resistance ability

The experimental results show that RCBAM outperforms the other attention mechanisms, with an average mAP improvement of 0.53% on the dataset. YOLOv7-RCBAM outperforms other classical YOLO methods with a mAP of 97.59%, which improves classification detection. Under the complex environment, the ship detection accu-

racy still reaches 96.13%, which verifies the strong interference resistance ability of YOLOv7-RCBAM.

The rest of this paper is organized as follows. Section 2 presents the current related works on ship detection. Section 3 introduces the design of the model, including CBAM, transfer learning, and the network architecture of YOLOv7-RCBAM. Section 4 conducts related experiments. Section 5 presents the conclusion and future works.

2. Related Works

Most of the traditional ship detection methods are based on synthetic aperture radar (SAR) images [23, 24], which is an active microwave earth observation device with all-weather, all-day operation and a certain penetration capability to the ground. It can obtain images of ship similar to optical photographs. The development of DL has also supplied effective assistance for ship detection in SAR images, so many methods based on SAR images are put forward.

Li et al. [25] proposed a region-based convolutional neural network (CNN) detection method which effectively extracted SAR image features at each scale. It replaced the region of interest pooling layer with RoIAlign to reduce the quantization error. Yang et al. [26] proposed an anchor-free method using rotatable bounding box on SAR ship detection called CPS-Det. It helped improve speed and proposed a scheme for calculating angle loss to improve the accuracy of angle prediction. He et al. [27] proposed a feature distillation framework to enhance mid-low-resolution ship detection. Yue et al. [28] proposed a two-stage SAR ship detection network to generate anchors. It can mainly capture small objects by generating high-quality anchors and improve the feature pyramid network by inserting a receptive field enhancement module, which can help enrich the feature map. However, it is difficult to process the weight of foreground and background information adaptively because of the improper processing of redundant information of image data. And the small ship detection is not easy to be detected based on the SAR images.

To solve the problem of small ship detection, the following papers were introduced. Chen et al. [29] proposed a method that combines a generative adversarial network (GAN) and a CNN-based detection approach, which can solve the problem of a limited number of small ships. In order to improve the detection accuracy of small ship objects, Zhou et al. [30] improved the YOLOv5s algorithm by optimizing the loss function and expanding the receptive field at the spatial pyramid pooling (SPP) layer. Although they proposed the methods against the small-scale ship, they cannot process the weight of background information. By the way, to solve the problem of feature information redundancy, the following papers introduce the attention mechanism. Li et al. [31] proposed an improved YOLOv3-tiny network, aiming at the real-time transport ship classification problem of waterway and river video surveillance images. It introduced CBAM to adjust the feature weights of the channel and space dimensions, so that the model can focus on the image ship object. Han et al. [32] proposed a ShipYOLO model to strengthen the detection speed and accuracy. They designed a new backbone network and amplified receptive field module to improve the

```

Input: Training set  $X = \{x_1, \dots, x_i\}$  and ground truths  $G = \{g_1, \dots, g_i\}$ 
Output: Trained model
Initialization: Learning rate  $Lr$ , batch size  $bs$ , and set each parameter value  $epoch$ ,  $weights$ .
Forepochs = 1 to ido
  //YOLOv7-RCBAM
  Preprocess  $x_i$  by mosaic:  $x_i = \text{ImagePreprocessing}(x_i)$ 
  Extract feature information in backbone:  $Features = \text{backbone}(x_i)$ 
  Fuse  $features$  by feature fusion module:  $Fused = \text{FeatureFusion}(features)$ 
  Attention module extract feature:  $Attention = \text{RCBAM}(fused)$ 
   $Objectness, anchor\_shape, feature\_cls, feature\_loc = \text{YOLOHead}(attention)$ 
  //Calculate loss and gradient descent
   $L = \text{box\_loss}(feature\_loc, g_i) + \text{obj\_loss}(objectness, g_i) + \text{cls\_loss}(feature\_cls, g_i)$ 
  Calculate gradients:  $Weight = \text{backpropagation}(L)$ 
End: save  $weights$  of model

```

ALGORITHM 1: Training of YOLOv7-RCBAM.

acquisition of small-scale ship. And they used the attention mechanism and ResNet's shortcut idea to improve the feature pyramid structure. Liu et al. [33] proposed a YOLOv4 method, which applied the reverse depthwise separable convolution (RDSC) to the backbone network and feature fusion network. It reduced the number of weights to increase the detection speed. They solved the difficulties in low detection accuracy in complex environments.

Based on the above papers, it can be found that the attention mechanism can be embedded in the DL model to effectively extract key information, and the network is lightweight to improve the training speed. However, the above methods still have the problem that the attention features are not obvious, and the lightweight model leads to the problem that the feature extraction is not rich enough. We propose YOLOv7-RCBAM based on video surveillance images to improve the richness of feature extraction information, improve the network training speed, and enhance the anti-interference ability of the model.

3. The Design of YOLOv7-RCBAM

We propose the method as follows. First, RCBAM is introduced to extract important features for model fine-tuning. Second, transfer learning is introduced to freezing parameters after pretraining the backbone network. Finally, feature information is enhanced by feature fusion method instead of feature disappearance in the deep network. The methods proposed are based on YOLOv7. YOLOv7 has better advantages in faster speed and higher accuracy. It introduced reparameterized module to reduce parameters and improve inference speed. And it introduced ELAN module for the backbone network, which can enhance the extraction features and improve the use of parameters and calculations. The trainable bag-of-freebies was proposed, so that the detection accuracy can be improved without increasing the inference cost. The specific process and the diagram of YOLOv7-RCBAM are shown in Algorithm 1 and Table 1.

The overall framework of our method is shown in Figures 1 and 2. YOLOv7-RCBAM network structure mainly includes backbone feature extraction module, SPPCSPC, feature fusion module, RCBAM, reparameteriza-

tion (REP) modules, and YOLO detection modules for regression object information.

As the backbone network, it usually adopts CBS module, ELAN, and max-pooling (MP) block. CBS is composed of convolutional layer, batch normalization (BN), and SiLU layer for feature extraction. The ELAN is composed by stacking a number of CBS modules for changing the channels and extracting feature information. And the MP block is composed of MaxPool layer and CBS module for downsampling. The SPPCSPC module performs downsampling through max-pooling layer and CBS layers of different sizes, effectively increasing the receptive field and separating the most salient contextual features. Then, in the neck part, feature fusion module includes upsample, ELAN-H layers, and MP-2 layers. They change the channels by upsample and MP-2 module, extract the feature by ELAN-H module, and do the job of feature fusion and communication. And it can convey the semantic and strengthen the extraction ability of multiscale targets. The RCBAM is added before the REP module to enhance the feature maps. Finally, in order to extract and smooth the feature, the REP module is composed of conv layers and BN layers before the prediction head.

3.1. Residual Convolutional Block Attention Module. In traditional object detection algorithms, feature extraction is usually performed on global information. The major shortcoming of the methods is the extraction feature information loss in the deep layers, which makes it difficult to focus on key objects. An effective feature map is necessary for the deep network. To improve the focus of the algorithm on the object, we introduce a RCBAM to enhance the foreground response of the ship.

The attention mechanism is a special module used to calculate the weights of input data and has been modified into a variety of attention mechanisms [34–36] used in DL. CBAM includes two independent submodules, namely, the channel attention mechanism (CAM) [37] module and the spatial attention mechanism (SAM) [38] module. The CAM adaptively assigns weights to receive $1*1*C$ feature map in the channel dimension, while the SAM adaptively assigns weights to receive $H*W*1$ feature map in the spatial dimension. We find that repeated feature recalibration makes the

TABLE 1: The diagram of YOLOv7-RCBAM.

Type/stride	Filter shape	Stride	Input size	Output size
Input image	—	—	640 * 640 * 3	640 * 640 * 3
Conv	3 * 3	1	640 * 640 * 3	640 * 640 * 32
Conv	3 * 3	2	640 * 640 * 32	320 * 320 * 64
Conv	3 * 3	1	320 * 320 * 64	320 * 320 * 64
Conv	3 * 3	2	320 * 320 * 64	160 * 160 * 128
Conv	1 * 1	1	160 * 160 * 128	160 * 160 * 64
Route 4	—	—	—	160 * 160 * 128
Conv	1 * 1	1	160 * 160 * 128	160 * 160 * 64
Conv	3 * 3	1	160 * 160 * 64	160 * 160 * 64
Conv	3 * 3	1	160 * 160 * 64	160 * 160 * 64
Conv	3 * 3	1	160 * 160 * 64	160 * 160 * 64
Conv	3 * 3	1	160 * 160 * 64	160 * 160 * 64
Route 5 7 9 11	—	—	—	160 * 160 * 256
Conv	1 * 1	1	160 * 160 * 256	160 * 160 * 256
MaxPool	—	—	160 * 160 * 256	80 * 80 * 256
Conv	1 * 1	1	80 * 80 * 256	80 * 80 * 128
Route 13	—	—	—	160 * 160 * 256
Conv	1 * 1	1	160 * 160 * 256	160 * 160 * 128
Conv	3 * 3	2	160 * 160 * 128	80 * 80 * 128
Route 15 18	—	—	—	80 * 80 * 256
Conv	1 * 1	1	80 * 80 * 256	80 * 80 * 128
Route 19	—	—	—	80 * 80 * 256
Conv	1 * 1	1	80 * 80 * 256	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Route 20 22 24 26	—	—	—	80 * 80 * 512
Conv	1 * 1	1	80 * 80 * 512	80 * 80 * 512
Conv	1 * 1	1	80 * 80 * 512	80 * 80 * 128
Route 28	—	—	—	80 * 80 * 512
MaxPool	—	—	80 * 80 * 512	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Route 28	—	—	—	80 * 80 * 512
Conv	1 * 1	1	80 * 80 * 512	80 * 80 * 256
Conv	3 * 3	2	80 * 80 * 256	40 * 40 * 256
Route 32 35	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Route 36	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Route 37 39 41 43	—	—	—	40 * 40 * 1024
Conv	1 * 1	1	40 * 40 * 1024	40 * 40 * 1024
Conv	1 * 1	1	40 * 40 * 1024	40 * 40 * 256
Route 45	—	—	—	40 * 40 * 1024
MaxPool	—	—	40 * 40 * 1024	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Route 45	—	—	—	20 * 20 * 512
Conv	1 * 1	1	40 * 40 * 1024	40 * 40 * 512
Conv	3 * 3	2	40 * 40 * 512	20 * 20 * 512
Route 49 52	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 256
Route 53	—	—	—	20 * 20 * 256
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Route 54 56 58 60	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Route 62	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Conv	1 * 1	1	20 * 20 * 512	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	20 * 20 * 512
Route 67	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	20 * 20 * 512
Route 67	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	20 * 20 * 512
Route 67 68 70 72	—	—	—	20 * 20 * 2048
Conv	1 * 1	1	20 * 20 * 2048	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Route 63 75	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Conv	1 * 1	1	20 * 20 * 512	20 * 20 * 256
Upsample	—	—	20 * 20 * 256	40 * 40 * 256
Route 46 79	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Route 80	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 128
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128
Route 81 83 85 87	—	—	—	40 * 40 * 1024
Conv	1 * 1	1	40 * 40 * 1024	40 * 40 * 256
Conv	1 * 1	1	40 * 40 * 256	40 * 40 * 128
Upsample	—	—	40 * 40 * 128	80 * 80 * 128
Route 29 91	—	—	—	80 * 80 * 256
Conv	1 * 1	1	80 * 80 * 256	80 * 80 * 128
Route 92	—	—	—	80 * 80 * 256
Conv	1 * 1	1	80 * 80 * 256	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 64
Conv	3 * 3	1	80 * 80 * 64	80 * 80 * 64
Conv	3 * 3	1	80 * 80 * 64	80 * 80 * 64
Conv	3 * 3	1	80 * 80 * 64	80 * 80 * 64
Route 93 95 97 99	—	—	—	80 * 80 * 512
Conv	1 * 1	1	80 * 80 * 512	80 * 80 * 128
MaxPool	—	—	80 * 80 * 128	40 * 40 * 128
Conv	1 * 1	1	40 * 40 * 128	40 * 40 * 128
Route 101	—	—	—	80 * 80 * 128
Conv	1 * 1	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	2	80 * 80 * 128	40 * 40 * 128
Route 103 106	—	—	—	40 * 40 * 256
Route 89 107	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Route 108	—	—	—	40 * 40 * 512
Conv	1 * 1	1	40 * 40 * 512	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 128
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128
Conv	3 * 3	1	40 * 40 * 128	40 * 40 * 128
Route 109 111 113 115	—	—	—	40 * 40 * 1024
Conv	1 * 1	1	40 * 40 * 1024	40 * 40 * 256
MaxPool	—	—	40 * 40 * 256	20 * 20 * 256
Conv	1 * 1	1	20 * 20 * 256	20 * 20 * 256
Route 117	—	—	—	40 * 40 * 256
Conv	1 * 1	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	2	40 * 40 * 256	20 * 20 * 256
Route 119 122	—	—	—	20 * 20 * 512
Route 77 123	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Route 124	—	—	—	20 * 20 * 1024
Conv	1 * 1	1	20 * 20 * 1024	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 256

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Conv	3 * 3	1	20 * 20 * 256	20 * 20 * 256
Route 125 127 129 131	—	—	—	20 * 20 * 2048
Conv	1 * 1	1	20 * 20 * 2048	20 * 20 * 512
Conv	1 * 1	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Route 133	—	—	—	20 * 20 * 512
AvgPool	—	—	20 * 20 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Route 133	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Add 141 145	—	—	—	1 * 1 * 512
Mul 133 146	—	—	—	20 * 20 * 512
Add 137 147	—	—	—	20 * 20 * 512
Mul 133 148	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	20 * 20 * 1
Route 149	—	—	—	20 * 20 * 512
AvgPool	—	—	20 * 20 * 512	20 * 20 * 1
Route 150 152	—	—	—	20 * 20 * 2
Conv	7 * 7	1	20 * 20 * 2	20 * 20 * 1
Mul 149 154	—	—	—	20 * 20 * 512
Add 133 155	—	—	—	20 * 20 * 512
Route 133	—	—	—	20 * 20 * 512
Conv	1 * 1	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Conv	3 * 3	1	20 * 20 * 512	20 * 20 * 512
Route 133	—	—	—	20 * 20 * 512
AvgPool	—	—	20 * 20 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Route 133	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Conv	1 * 1	1	1 * 1 * 512	1 * 1 * 512
Add 165 169	—	—	—	1 * 1 * 512
Mul 133 170	—	—	—	20 * 20 * 512

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Add 161 171	—	—	—	20 * 20 * 512
Mul 133 172	—	—	—	20 * 20 * 512
MaxPool	—	—	20 * 20 * 512	20 * 20 * 1
Route 173	—	—	—	20 * 20 * 512
AvgPool	—	—	20 * 20 * 512	20 * 20 * 1
Route 174 176	—	—	—	20 * 20 * 2
Conv	7 * 7	1	20 * 20 * 2	20 * 20 * 1
Mul 173 178	—	—	—	20 * 20 * 512
Add 133 179	—	—	—	20 * 20 * 512
Add 156 180	—	—	—	20 * 20 * 512
RepConv	3 * 3	1	20 * 20 * 512	20 * 20 * 1024
YOLO				
Route 117	—	—	—	40 * 40 * 256
Conv	1 * 1	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Route 117	—	—	—	40 * 40 * 256
AvgPool	—	—	40 * 40 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Route 117	—	—	—	40 * 40 * 256
MaxPool	—	—	40 * 40 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Add 192 196	—	—	—	1 * 1 * 256
Mul 117 197	—	—	—	40 * 40 * 256
Add 188 198	—	—	—	40 * 40 * 256
Mul 117 199	—	—	—	40 * 40 * 256
MaxPool	—	—	40 * 40 * 256	40 * 40 * 1
Route 200	—	—	—	40 * 40 * 256
AvgPool	—	—	40 * 40 * 256	40 * 40 * 1
Route 201 203	—	—	—	40 * 40 * 2
Conv	7 * 7	1	40 * 40 * 2	40 * 40 * 1
Mul 200 205	—	—	—	40 * 40 * 256
Add 117 206	—	—	—	40 * 40 * 256
Route 117	—	—	—	40 * 40 * 256
Conv	1 * 1	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Conv	3 * 3	1	40 * 40 * 256	40 * 40 * 256
Route 117	—	—	—	40 * 40 * 256
AvgPool	—	—	40 * 40 * 256	1 * 1 * 256

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Route 117	—	—	—	40 * 40 * 256
MaxPool	—	—	40 * 40 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Conv	1 * 1	1	1 * 1 * 256	1 * 1 * 256
Add 216 220	—	—	—	1 * 1 * 256
Mul 117 221	—	—	—	40 * 40 * 256
Add 212 222	—	—	—	40 * 40 * 256
Mul 117 223	—	—	—	40 * 40 * 256
MaxPool	—	—	40 * 40 * 256	40 * 40 * 1
Route 224	—	—	—	40 * 40 * 256
AvgPool	—	—	40 * 40 * 256	40 * 40 * 1
Route 225 227	—	—	—	40 * 40 * 2
Conv	7 * 7	1	40 * 40 * 2	40 * 40 * 1
Mul 224 229	—	—	—	40 * 40 * 256
Add 117 230	—	—	—	40 * 40 * 256
Add 207 231	—	—	—	40 * 40 * 256
RepConv	3 * 3	1	40 * 40 * 256	40 * 40 * 512
YOLO				
Route 101	—	—	—	80 * 80 * 128
Conv	1 * 1	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Route 101	—	—	—	80 * 80 * 128
AvgPool	—	—	80 * 80 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Route 101	—	—	—	80 * 80 * 128
MaxPool	—	—	80 * 80 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Add 243 247	—	—	—	1 * 1 * 128
Mul 101 248	—	—	—	80 * 80 * 128
Add 239 249	—	—	—	80 * 80 * 128
Mul 101 250	—	—	—	80 * 80 * 128
MaxPool	—	—	80 * 80 * 128	80 * 80 * 1
Route 251	—	—	—	80 * 80 * 128
AvgPool	—	—	80 * 80 * 128	80 * 80 * 1
Route 252 254	—	—	—	80 * 80 * 2
Conv	7 * 7	1	80 * 80 * 2	80 * 80 * 1
Mul 251 256	—	—	—	80 * 80 * 128

TABLE 1: Continued.

Type/stride	Filter shape	Stride	Input size	Output size
Add 101 257	—	—	—	80 * 80 * 128
Route 101	—	—	—	80 * 80 * 128
Conv	1 * 1	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Conv	3 * 3	1	80 * 80 * 128	80 * 80 * 128
Route 101	—	—	—	80 * 80 * 128
AvgPool	—	—	80 * 80 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Route 101	—	—	—	80 * 80 * 128
MaxPool	—	—	80 * 80 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Conv	1 * 1	1	1 * 1 * 128	1 * 1 * 128
Add 267 271	—	—	—	1 * 1 * 128
Mul 101 272	—	—	—	80 * 80 * 128
Add 263 273	—	—	—	80 * 80 * 128
Mul 101 274	—	—	—	80 * 80 * 128
MaxPool	—	—	80 * 80 * 128	80 * 80 * 1
Route 275	—	—	—	80 * 80 * 128
AvgPool	—	—	80 * 80 * 128	80 * 80 * 1
Route 276 278	—	—	—	80 * 80 * 2
Conv	7 * 7	1	80 * 80 * 2	80 * 80 * 1
Mul 275 279	—	—	—	80 * 80 * 128
Add 101 280	—	—	—	80 * 80 * 128
Add 258 281	—	—	—	80 * 80 * 128
RepConv	3 * 3	1	80 * 80 * 128	80 * 80 * 256
YOLO				

depth feature response decrease and affects the detection result, so the improvement of the CAM module can help improve the accuracy of ship detection.

The basic structure of residual channel attention mechanism (RCAM) is shown in Figure 3. The CAM adds a parallel maximum pooling layer. The input feature maps compress the global information through the pooling layer and then obtain the characteristic map through two-layer convolution with activation function, respectively:

$$M_{\text{Avg}} = F_1(\text{ReLU}(F(\text{AvgPool}(z))))), \quad (1)$$

$$M_{\text{Max}} = F_1(\text{ReLU}(F(\text{MaxPool}(z))))), \quad (2)$$

$$M_{\text{pool}} = \text{Sigmoid}(M_{\text{Avg}} + M_{\text{Max}}), \quad (3)$$

where z is the input of the CAM. AvgPool and MaxPool are the pooling layers. ReLU and Sigmoid are the activation

function. F is the convolution layer. The output of them are M_{Avg} and M_{Max} .

Then, the channel information is recalibrated by matrix point multiplication. The feature map by the sigmoid activation function is denoted as M_{pool} , and then, after recalibration

$$M = Z \cdot M_{\text{pool}} = [z_1 m_1, z_2 m_2, \dots, z_i m_i]. \quad (4)$$

Finally, we stack number of the convolutional batch-normalization SiLU (CBS) modules. It can help extract effective feature in the deep layers. The CBS₁ is used to smooth feature by 1×1 kernel size convolution layer, and the last three CBS₃ can be used to extract the feature by 3×3 kernel size convolution layers and maintain the original channel dimension:

$$N = \text{CBS}_3(\text{CBS}_3(\text{CBS}_3(\text{CBS}_1(z)))). \quad (5)$$

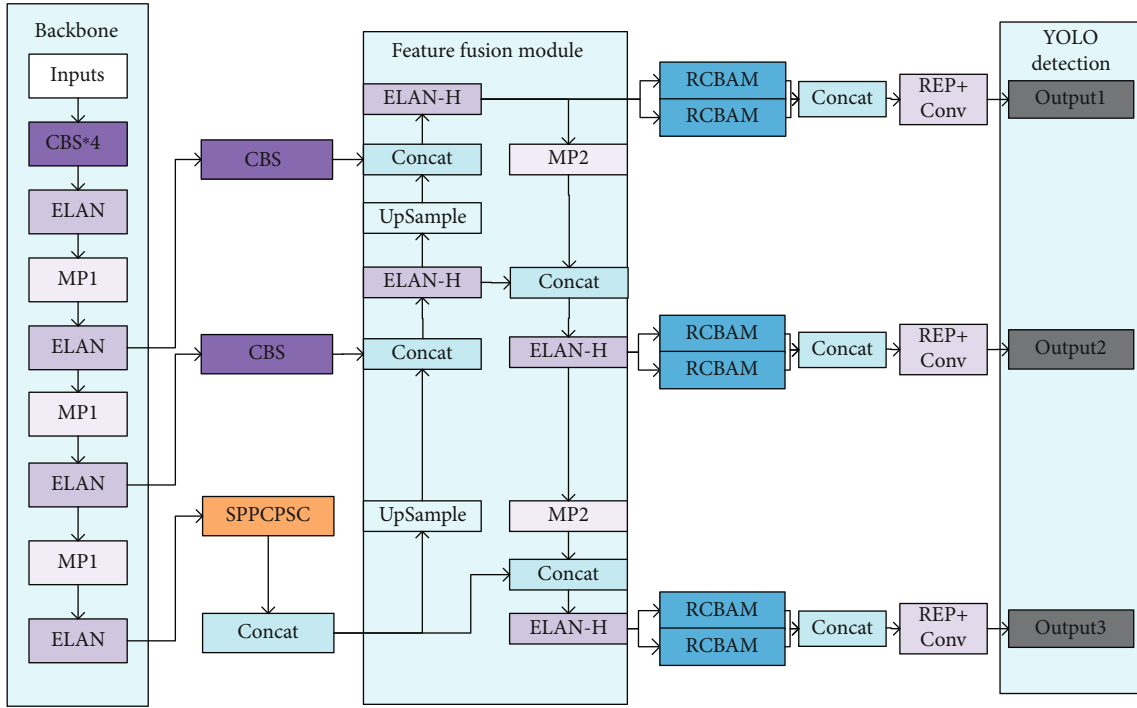


FIGURE 1: The structure of YOLOv7-RCBAM network.

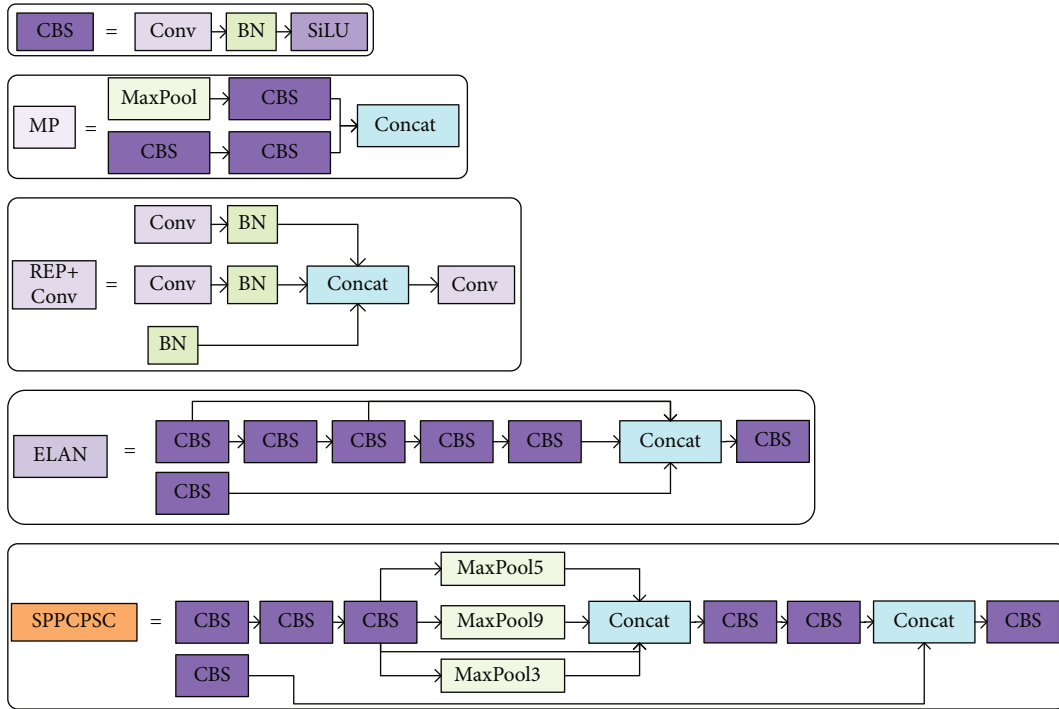


FIGURE 2: The modules of YOLOv7-RCBAM network.

After that, we add the operation of merging and dot-multiplying the feature map by the CAM module with the feature extraction map. Therefore, M fully considers the lead of global information and effectively highlights the discriminative feature information of the ship. And the N is formed after feature extraction by four CBS modules. After the “+” operation, we get

$$Z_1 = N + M, \tag{6}$$

since successive repetitive feature recalibration operations can lead to lower response values of depth features and thus affect the detection effect. Residual connection helps to fuse the extraction information and prevents the loss of

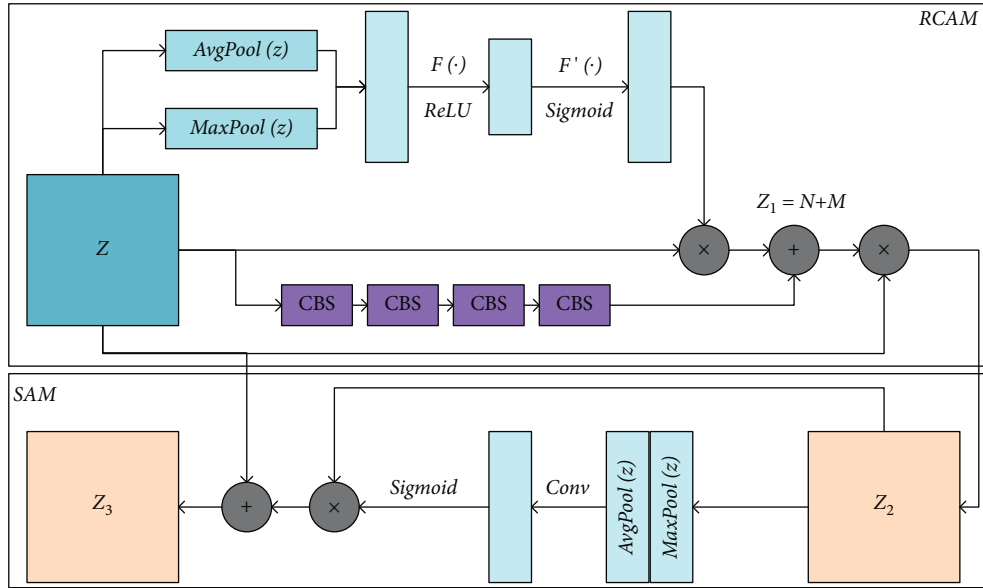


FIGURE 3: The RCBAM.

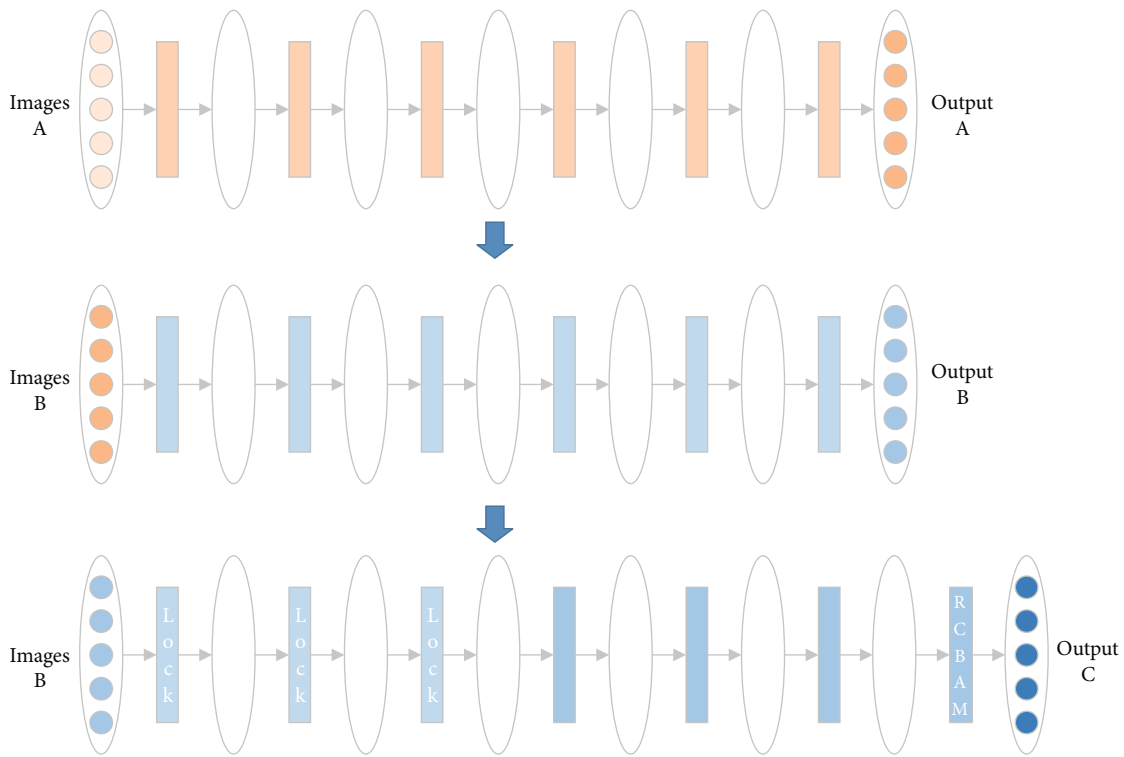


FIGURE 4: The transfer model.

feature information due to recalibration. Finally, the feature map Z_2 is obtained after recalibration. We introduce residual connection between successive feature recalibrations by using the idea of residual learning. It can improve the feasibility of optimization while preserving the original information. In the end, the RCBAM is proposed by connecting the output of the residual channel attention module (RCAM) to the SAM.

3.2. Double Transfer Strategy. In DL, the amount of parameters needs to be trained in the face of a large dataset. Training from scratch easily leads to the problems of slow training speed and poor interference resistance. Therefore, transfer learning is introduced to improve the training process and enhance the interference resistance in this paper.

Model-based transfer learning, also called parameter-based transfer learning, shares some common knowledge

TABLE 2: The comparison of AP in different attention mechanisms in different ship categories.

Attention model	Bulk cargo carrier	Container ship	Fishing boat	General cargo ship	Ore carrier	Passenger ship
YOLOv7	96.89%	99.99%	95.49%	97.48%	96.66%	93.33%
YOLOv7+ECA	96.71%	100%	95.84%	97.81%	97.67%	95.18%
YOLOv7+SE	96.20%	100%	95.89%	97.78%	97.69%	94.16%
YOLOv7+CBAM	96.70%	100%	95.75%	97.76%	97.96%	93.93%
YOLOv7-RCBAM	96.90%	100%	95.93%	98.16%	97.99%	96.56%

TABLE 3: The comparison of indicators in different attention mechanisms.

Attention model	R	P	F_1	mAP
YOLOv7	91.72%	95.86%	93.50%	96.64%
YOLOv7+ECA	93.88%	95.95%	95.00%	97.20%
YOLOv7+SE	94.31%	95.88%	95.00%	96.95%
YOLOv7+CBAM	93.79%	96.40%	95.00%	97.02%
YOLOv7-RCBAM	94.14%	96.00%	95.00%	97.59%

TABLE 4: The comparison of AP values for each model between different ship categories.

Model	Bulk cargo carrier	Container ship	Fishing boat	General cargo ship	Ore carrier	Passenger ship	mAP
YOLOv4	92.79%	98.61%	94.21%	93.83%	94.29%	90.08%	93.97%
YOLOv5	95.80%	99.50%	95.90%	98.50%	95.00%	96.00%	96.80%
YOLOv7	96.89%	99.99%	95.49%	97.48%	96.66%	93.33%	96.64%
YOLOv7-RCBAM	96.90%	100%	95.93%	98.16%	97.99%	96.56%	97.59%

between the original task and the target task at the model level. The transfer learning based on sharing parameters achieves the purpose of knowledge transfer by freezing the common parameters of some models. The premise of the transfer of shared parameters is that there are similar features in the learning task to make the model parameters consistent.

Since the model is not easy to identify ships, it is difficult to do well in the fine-grained detection of ships, so the model fine-tuning method is introduced. As shown in Figure 4, the VOC2007 dataset A is loaded into the original model for training to get the pretraining parameters, based on which the SeaShips dataset B is input for classification experimental training to make the model improve in ship category recognition. To further improve the accuracy of ship classification detection and increase the model training speed, we add an attention mechanism to the tail of the model for ship classification dataset B training by freezing the backbone network.

3.3. Feature Fusion Module. The features extracted by the single attention mechanism are not obvious. To deepen the features that the attention mechanism pays attention to, the output of the attention is fused, which enhances the feature performance [39, 40]. Given the problem of unfocus in different feature fusion modules, we introduce the same attention fusion to enhance the effective features and prevent feature disappearance from the deep network.

We add an improved convolutional attention mechanism to the three-dimensional feature maps of 128, 256, and 512 and fuse two identical feature maps through a merging operation to enhance the effect of the features extracted by the attention mechanism. The resulting features are

$$\text{Out} = Z_3 + Z_3. \quad (7)$$

The feature fusion attention module can effectively prevent the loss of depth information so that the model can learn rich features and pay attention to the target features, instead of superimposing attention feature, which can change the feature dimension and lose the feature information. Fusing attention feature is more beneficial to retain feature dimension and information.

4. Experiments

4.1. Datasets and Evaluation Metrics. At present, there are some datasets with ship categories; these include the COCO dataset, VOC dataset, CIFAR-10 dataset, and SeaShips [41] dataset. COCO and VOC only have one type of ship (boat), so ship classification experiments cannot be performed. There are 7000 open-source data in SeaShips dataset, which contains 948 passenger ships, 4398 mining ships, 3010 general cargo ships, 4380 fishing ships, 1802 container ships, and 3904 bulk carriers in total of six types of ships, so the SeaShips dataset is chosen for our experiment. The dataset is divided into training

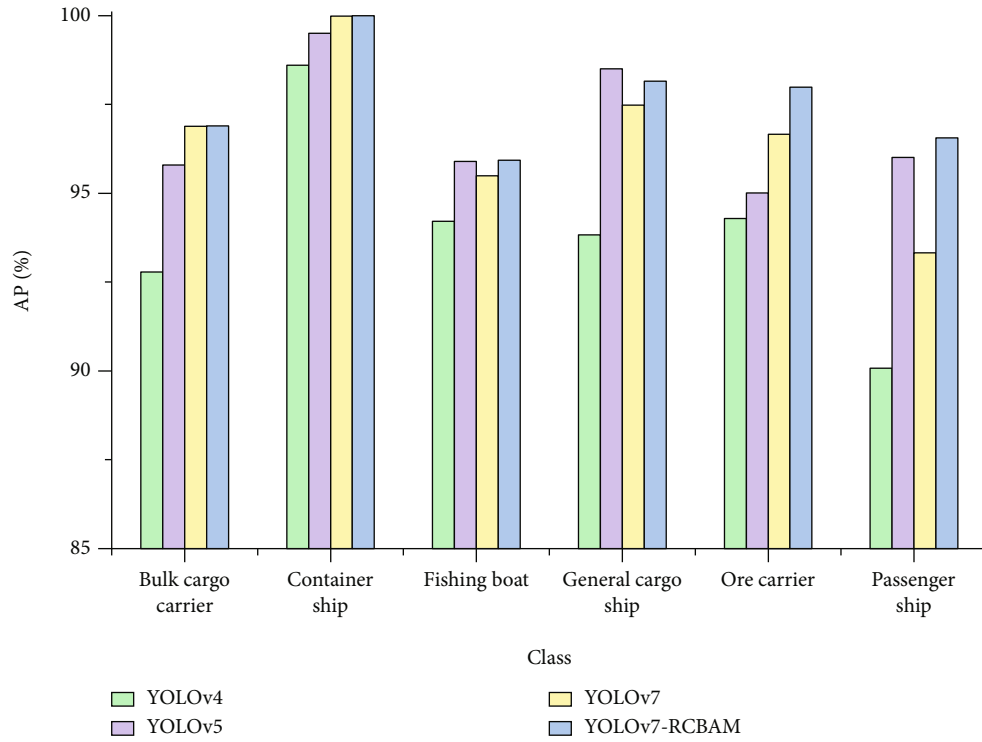


FIGURE 5: The histogram comparison of AP in different models between different ship categories.



FIGURE 6: Different model test renderings.

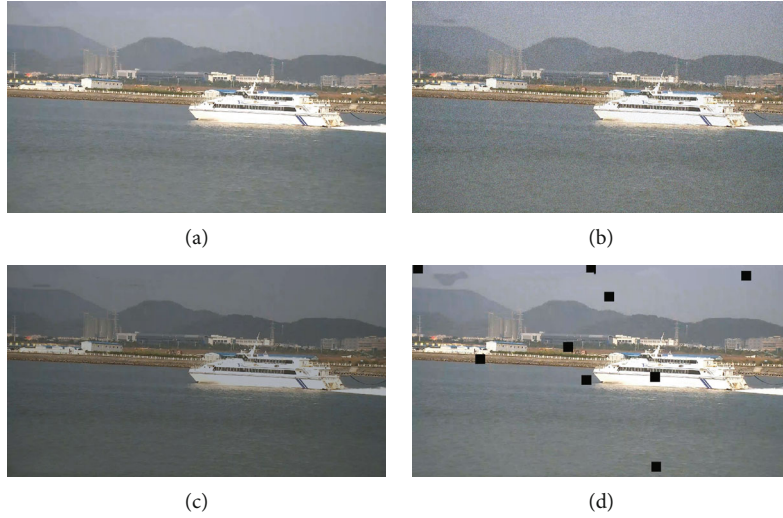


FIGURE 7: (a) Original, (b) add noise, (c) adjust brightness, (d) and cutout.

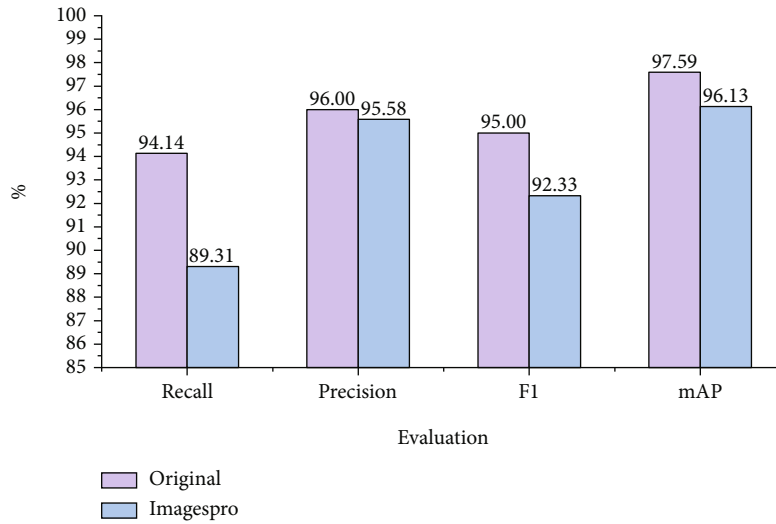


FIGURE 8: The comparison of data enhancement assessment metrics.

set and test set in the ratio of 4:1, and the training set is divided into subtraining set and validation set of 4:1. The subtraining set contains 4480 images, and the validation set contains 1120 images, while the test set contains 1400 images.

We select a variety of evaluation indexes to prove the superiority of the model, including precision (P), recall (R), average precision (AP), and mAP. The main usage indicators are as follows:

- (1) AP refers to the area under the precision-recall (P-R) curve when balancing the P and R to represent the AP of each category in the model detection degree, and calculation formula is

$$AP = \int_0^1 P(R) dR. \quad (8)$$

TABLE 5: The comparison of data enhancement accuracy.

Model	R	P	mAP
YOLOv4	77.89%	90.45%	86.62%
YOLOv5	88.30%	91.10%	92.60%
YOLOv7	72.37%	92.21%	91.19%
YOLOv7-RCBAM	89.31%	95.58%	96.13%

- (2) mAP refers to the mean of the AP of all categories and the calculation formula is

$$mAP = \frac{\sum_{i=1}^N AP_i}{N}. \quad (9)$$

N is the number of all target categories.



FIGURE 9: The image enhancement detection effect.

4.2. *Experiment Analysis of Ship Classification Detection.* In this paper, we set 20 training epochs and mainly use mAP values for model detection accuracy assessment. First, to verify the advantage of our method in comparison with different advanced attention mechanisms, we compare the accuracy of various types of ships. We compare this method with other YOLO methods to verify the superiority of our method. To verify the detection ability of the method to complex environment, the image is enhanced to train the model.

4.2.1. *Comparison of Different Attention Mechanisms.* To verify the effectiveness of the attention mechanism we proposed, we use the YOLOv7 as the base model, and the efficient channel attention (ECA), squeeze-and-excitation attention (SE), and CBAM and RCAM are added to it, respectively.

The experimental results show that our method performs well. As shown in Table 2, compared with the AP of the original YOLOv7 model, there are different degrees of improvement in the AP of all types of ships. The ECA and SE model focus on the channel dimension and ignore the spatial dimension in the ship images, resulting in a low AP for all types of ship detection. Compared with the CBAM, our method improves the channel attention mechanism, which can retain the detailed information of various ships and improve the detection accuracy. However, in the category of ore carriers, large ships are more likely to mistake the background for a ship due to the obstruction of the external environment, resulting in a decrease in ship detection accuracy. As can be seen from Table 3, our method achieves the highest mAP of 97.59% among various attention mechanisms. The R and P are increased to 96.00% and 94.14%, and the F_1 value reaches 95.00%.

4.2.2. *Comparison of Different Models.* To prove that our method has a good detection effect compared with other object detection models, we use the SeaShips dataset for model training and evaluation. The experimental models for comparison are YOLO4, YOLOv5, and YOLOv7. The experimental results are as shown.

The experimental results in Table 4 and Figure 5 show that our method improves the AP of various types of ships by different degrees compared with various models, but in the class general cargo ship, our method is lower than the YOLOv5 by 0.34%. Compared with the YOLOv4, YOLOv5, and YOLOv7, our method adds an improved attention mechanism to reduce the feature extraction of redundant

TABLE 6: The comparison of indicators in different attention mechanisms.

Methods	Small ship detection	ShipYOLO	Enhanced YOLOv3-tiny	Ours
mAP	96.35%	95.50%	97.00%	97.59%

information and focus on the ship target features, which can significantly improve the mAP value and AP values. The improvement over the YOLOv4 networks is particularly significant, with mAP improving by 3.62%. While comparing the YOLOv5, mAP improves by 0.79% to reach a maximum of 97.59%, reflecting the superiority of the model's mAP. Figure 6 shows the effect of six types of ship detection. YOLOv5 networks have located the wrong bounding boxes in bulk cargo boat, indicating that the method have worse detection performance. However, YOLOv4 networks have repeatedly detected the bounding boxes, indicating that the method can correctly locate and classify the ship target, but due to the inaccuracy of the NMS, the wrong bounding boxes cannot be correctly eliminated. YOLOv7 networks have predicted a bigger bounding box in ore carrier, which show that it can correctly locate the target but mismatch the background. Our method focuses on object features and enhances feature performance, so it can correctly classify and locate ships.

4.2.3. *Test of Model Interference Resistance.* As shown in Figure 7, to verify the interference resistance of the model detection, we cutout the images to occlude the ships, reduce the image brightness to simulate the night, and finally increase the image noise to simulate rainy and foggy conditions. We adopt the method of random enhancement of each ship image and randomly select one of the methods of cutout, changing brightness, and increasing noise for random scale enhancement. It can increase the richness of the dataset and enable the model to adapt to multiple ship detections in different scenarios.

The interference resistance of the model is verified after image enhancement of the original data, as shown in the figures.

As shown in Figure 8, after image enhancement, it is difficult for the model to detect the ship objects in environments such as darkness, noisy, and cutout, resulting in a decrease in the model R of 4.83%, and the detection accuracy

and mAP of the model decrease by 0.42% and 1.46%. As shown in Table 5, compared with different models, our method takes the lead mAP by 96.13%, indicating that the model is effective in detecting the recalled ship target features and can correctly classify ships. Facing complex environments, our method focuses on the ship's target itself during detection and can still achieve a high precision. It can be seen that the method is interference resistant and can adapt to various environmental vessel classification detection. From the effect figure in Figure 9, we know that our model can correctly locate ship targets and classify ship types in the environment with weak light and strong noise, while the model can still locate and classify ships without being affected by target occlusion, multiple target overlaps, and small target ships in the image.

4.2.4. Comparison with Previous State-of-the-Art Approaches. To verify that our method has a good detection effect, we compare it with previous SOTA approaches in the SeaShips dataset. The experimental models for comparison are small ship detection method [42], ShipYOLO [32], and enhanced YOLOv3-tiny [31]. The results are as shown.

The experimental results in Table 6 show that our method achieves the highest level of mAP with 97.59% over the other methods. Compared with small ship detection, our method improves the mAP by 1.24%. Small ship detection and ShipYOLO methods ignore the extraction feature in deep layers due to the lower accuracy. Enhanced YOLOv3-tiny method reduces the parameters to improve the detection speed, while the mAP is 0.59% lower than ours. Although the methods add an attention mechanism to focus on the ship objects, our RCBAAM performs better, since the RCBAAM extracts the deep feature by CBS module and fuses the extraction information by residual connection. By the way, feature recalibration operations help improve the feasibility of optimization, and feature fusion attention module effectively captures the deep rich features.

5. Conclusion

In this paper, aiming at the problems of inaccurate object feature extraction and inconspicuous feature information in deep layers, a YOLOv7-RCBAAM for the ship detection method is proposed. In our design, the RCBAAM module is introduced to extract object feature information effectively, and the double transfer learning and the feature fusion attention of the method can fuse the feature information and avoid feature losing in deep layers. The effectiveness of our method based on YOLOv7-RCBAAM is verified by case studies on several experimental data in this paper. Compared with other benchmark models and state-of-the-art methods, our method has better detection accuracy and anti-interference. In future research work, the real-time performance of ship detection is particularly important. We will focus on improving the detection accuracy of the model while reducing the number of model parameters, to improve the training speed of the model and achieve the purpose of lightweight models.

Data Availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors' Contributions

J.C. and H.F. wrote the main manuscript text, X.L., Y.H., and H.L. listed the data, and H.L. and W.H. reviewed and edited the manuscript. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This work was supported by the Guangzhou Science and Technology Key R&D Program (grant number 202206010022), the Innovation Team Project of Ordinary University of Guangdong Province (grant number 2020KCXTD017) the Guangdong Special Project in Key Field of Artificial Intelligence for Ordinary University (grant number 2019KZDZX1004), and the Guangzhou Key Laboratory Construction Project (grant number 202002010003).

References

- [1] Y.-L. Chang, A. Anagaw, L. Chang, Y. Wang, C. Y. Hsiao, and W. H. Lee, "Ship detection based on YOLOv2 for SAR imagery," *Remote Sensing*, vol. 11, no. 7, p. 786, 2019.
- [2] N. Liu, Z. Cao, Z. Cui, Y. Pi, and S. Dang, "Multi-scale proposal generation for ship detection in SAR images," *Remote Sensing*, vol. 11, no. 5, p. 526, 2019.
- [3] F. Wang, J. Cen, Z. Yu, S. Deng, and G. Zhang, "Research on a hybrid model for cooling load prediction based on wavelet threshold denoising and deep learning: a study in China," *Energy Reports*, vol. 8, pp. 10950–10962, 2022.
- [4] J. Zhang, X. Zou, L. D. Kuang, J. Wang, R. S. Sherratt, and X. Yu, "CCTSDB 2021: a more comprehensive traffic sign detection benchmark," *Human-centric Computing Information Sciences*, vol. 12, 2022.
- [5] H. Chen, J. Cen, Z. Yang, W. Si, and H. Cheng, "Fault diagnosis of the dynamic chemical process based on the optimized CNN-LSTM network," *ACS Omega*, vol. 7, no. 38, pp. 34389–34400, 2022.
- [6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus, OH, USA, 2014.
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.
- [8] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, pp. 379–387, 2016.

- [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, Venice, Italy, 2017.
- [10] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," *European conference on computer vision*, vol. 9905, 2016.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, Las Vegas, NV, USA, 2016.
- [12] R. Xia, Y. Chen, and B. Ren, "Improved anti-occlusion object tracking algorithm using unscented Rauch-Tung-Striebel smoother and kernel correlation filter," *Journal of King Saud University-Computer Information Sciences*, vol. 34, no. 8, pp. 6008–6018, 2022.
- [13] Z. Liu, Q. Li, and W. Li, "Deep layer guided network for salient object detection," *Neurocomputing*, vol. 372, pp. 55–63, 2020.
- [14] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, <https://arxiv.org/abs/02696>.
- [15] H. Wang, Z. Guan, S. Yu, J. Cao, and Y. Li, "Infrared and visible image fusion via decoupling network," *IEEE Transactions on Instrumentation Measurement*, vol. 71, pp. 1–13, 2022.
- [16] M. Hu, J. Feng, J. Hua et al., "Online convolutional re-parameterization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 568–577, New Orleans, LA, USA, 2022.
- [17] X. Zhang, H. Zeng, S. Guo, and L. Zhang, "Efficient long-range attention network for image super-resolution," 2022, <https://arxiv.org/abs/06697>.
- [18] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," *Advances in neural information processing systems*, vol. 29, pp. 343–351, 2016.
- [19] 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, Las Vegas, NV, USA, 2016.
- [20] S. Woo, J. Park, J. Lee, and I. Kweon, "Cbam: convolutional block attention module," in *Proceedings of the European conference on computer vision*, pp. 3–19, Munich, Germany, 2018.
- [21] J. Zhang, W. Feng, T. Yuan, J. Wang, and A. K. Sangaiah, "SCSTCF: spatial-channel selection and temporal regularized correlation filters for visual tracking," *Applied Soft Computing*, vol. 118, article 108485, 2022.
- [22] H. Wang, Z. Liu, D. Peng, and Y. Qin, "Understanding and learning discriminant features based on multiattention 1DCNN for wheelset bearing fault diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5735–5745, 2019.
- [23] M. Zhu, G. Hu, S. Li, H. Zhou, and S. Wang, "FSFADet: arbitrary-oriented ship detection for SAR images based on feature separation and feature alignment," *Neural Processing Letters*, vol. 54, no. 3, pp. 1995–2005, 2022.
- [24] Z. Sun, X. Leng, Y. Lei, B. Xiong, K. Ji, and G. Kuang, "BiFA-YOLO: a novel YOLO-based method for arbitrary-oriented ship detection in high-resolution SAR images," *Remote Sensing*, vol. 13, no. 21, p. 4209, 2021.
- [25] Y. Li, S. Zhang, and W. Wang, "A lightweight faster R-CNN for ship detection in SAR images," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
- [26] Y. Yang, Z. Pan, Y. Hu, and C. Ding, "CPS-Det: an anchor-free based rotation detector for ship detection," *Remote Sensing*, vol. 13, no. 11, p. 2208, 2021.
- [27] S. He, H. Zou, Y. Wang et al., "Enhancing mid-low-resolution ship detection with high-resolution feature distillation," *IEEE Geoscience Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.
- [28] T. Yue, Y. Zhang, P. Liu, Y. Xu, and C. Yu, "A generating-anchor network for small ship detection in SAR images," *IEEE Journal of Selected Topics in Applied Earth Observations Remote Sensing*, vol. 15, pp. 7665–7676, 2022.
- [29] Z. Chen, D. Chen, Y. Zhang, X. Cheng, M. Zhang, and C. Wu, "Deep learning for autonomous ship-oriented small ship detection," *Safety Science*, vol. 130, article 104812, 2020.
- [30] J. Zhou, P. Jiang, A. Zou, X. Chen, and W. Hu, "Ship target detection algorithm based on improved YOLOv5," *Journal of Marine Science Engineering*, vol. 9, no. 8, p. 908, 2021.
- [31] H. Li, L. Deng, C. Yang, J. Liu, and Z. Gu, "Enhanced YOLO v3 tiny network for real-time ship detection from visual image," *IEEE Access*, vol. 9, pp. 16692–16706, 2021.
- [32] X. Han, L. Zhao, Y. Ning, and J. Hu, "ShipYolo: an enhanced model for ship detection," *Journal of Advanced Transportation*, vol. 2021, Article ID 1060182, 11 pages, 2021.
- [33] T. Liu, B. Pang, L. Zhang, W. Yang, and X. Sun, "Sea surface object detection algorithm based on YOLO v4 fused with reverse depthwise separable convolution (RDSC) for USV," *Journal of Marine Science Engineering*, vol. 9, no. 7, p. 753, 2021.
- [34] J. Yang, C. Zhang, Y. Tang, and Z. Li, "PAFM: pose-drive attention fusion mechanism for occluded person re-identification," *Neural Computing Applications*, vol. 34, no. 10, pp. 8241–8252, 2022.
- [35] Y. Chen, L. Liu, V. Phonevilay et al., "Image super-resolution reconstruction based on feature map attention mechanism," *Applied Intelligence*, vol. 51, no. 7, pp. 4367–4380, 2021.
- [36] Z. Yang, J. Cen, X. Liu, J. Xiong, and H. Chen, "Research on bearing fault diagnosis method based on transformer neural network," *Measurement Science Technology*, vol. 33, no. 8, p. 085111, 2022.
- [37] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.
- [38] X. Zhu, D. Cheng, Z. Zhang, S. Lin, and J. Dai, "An empirical study of spatial attention mechanisms in deep networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6688–6697, Seoul, South Korea, 2019.
- [39] L. Dai, J. Liu, and Z. Ju, "Binocular feature fusion and spatial attention mechanism based gaze tracking," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 2, pp. 302–311, 2022.
- [40] W. Zhou, Z. Xia, P. Dou, T. Su, and H. Hu, "Attention-augmented memory network for image multi-label classification," *ACM Transactions on Multimedia Computing, Communications, Applications*, vol. 19, no. 1, pp. 1–23, 2022.
- [41] 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.
- [42] J. Zheng and Y. Liu, "A study on small-scale ship detection based on attention mechanism," *IEEE Access*, vol. 10, pp. 77940–77949, 2022.