Two-Stage Underwater Object Detection Network Using Swin Transformer

JIA LIU, SHUANG LIU, SHUJUAN XU, (Member, IEEE), AND CHANGJUN ZHOU

1School of Computer Science and Engineering, Dalian Minzu University, Dalian, Liaoning 116600, China
2College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, Zhejiang 321004, China

Corresponding author: Shuang Liu (dlnuliushuang@qq.com)

This work was supported in part by the Liaoning Province Economic and Social Development Research Project 2023 of Provincial Social Science Association under Grant 2023lslybkt-039, in part by the National Youth Science Foundation of China under Grant 62002046, and in part by the National Natural Science Foundation of China under Grant 61672121.

ABSTRACT Underwater object detection plays an essential role in ocean exploration, and the increasing amount of underwater object image data makes the study of advanced underwater object detection algorithms of great practical significance. However, there are problems with colour offset, low contrast, and target blur in underwater image data. An underwater object detection algorithm based on Faster R-CNN is proposed to solve these problems. First, the Swin Transformer is used as the backbone network of the algorithm. Second, by adding the path aggregation network, the deep feature map and the shallow feature map are superimposed and fused. Third, online hard example mining makes the training process more efficient. Fourth, the ROI pooling is improved to ROI align, eliminating the two quantization errors of ROI pooling and improving the detection performance. Compared with other algorithms, the proposed algorithm’s based on improved Faster-RCNN on URPC2018 dataset is improved to 80.54%, and basically solve the problem of missed detection and false detection of objects of different sizes in a complex environment.

INDEX TERMS Deep learning, underwater target detection, swin transformer, feature fusion, faster R-CNN, OHEM.

I. INTRODUCTION

More than 70 percent of the earth’s surface area is occupied by oceans, which produce almost half of the earth’s oxygen, absorb the most carbon dioxide from the environment, and provide countless marine resources for human beings. The rational development of the ocean is inseparable from acquiring underwater information. There are two main ways to obtain underwater information: underwater sonar technology and underwater optical imaging technology. Compared with underwater sonar technology, underwater optical imaging technology has the advantages of intuitive object detection, high imaging resolution, and a large amount of information. It is more suitable for short-range object detection. In recent years, most underwater exploration has relied on divers, but long-term diving operations and complex underwater environments have significantly burdened their health.

Therefore, the research on underwater object detection is significant.

The traditional underwater object detection mainly relies on extracting the features [1], [2] manually designed in the candidate bounding box, and sending them into the support vector machine [3], AdaBoost [4] and other classifiers for detection. Many researchers utilize these traditional methods for underwater object recognition. Xu et al. [5] proposed an underwater object feature extraction method based on the singular value of a generalized S-Transform module time-frequency matrix. Ma et al. [6] analyzed and extracted polarization features, edge features and line features that are more suitable for object detection in underwater environments, and then used the model to generate feature maps to detect underwater objects. Liu et al. [7] proposed a feature matching algorithm based on Hough transform and geometric features for object detection in special underwater environments. Li et al. [8] proposed an underwater small object recognition algorithm based on shape features. Not only that, but there...
are methods for self-supervised learning using graph neural networks [9].

However, the underwater environment is complex and changeable, and the artificially designed features are not robust enough to satisfy generalization requirements. As deep learning is well known for its powerful automatic feature extraction ability in the field of image recognition, Girshick et al. also proposed R-CNN [10] object detection algorithm for the first time in 2014. It opens the application of deep learning in the field of object detection.

As a milestone in applying convolutional neural networks to object detection, R-CNN had good feature extraction and classification performance at that time, surpassing all traditional object detection algorithms. However, a series of problems such as low efficiency and long time consumption has led to the application of R-CNN not being widely used. While aiming at many R-CNN problems, Ross Girshick proposes an improved algorithm Fast R-CNN [11] with higher practicability and faster speed. Faster R-CNN [12] used the RPN(Region Proposal Network) so that the four steps required for object detection, candidate region generation, feature extraction, classifier classification, and regressor regression are all handed over to the deep neural network and run on the GPU, which greatly improves the efficiency of the operation.

The classification and localization of these algorithms are carried out separately, called two-stage detection algorithms. In addition, other two-stage detection networks that improve on the above two-stage detection algorithm can also produce an excellent performance. Such as Mask R-CNN [13], Sparse R-CNN [14], Dynamic R-CNN [15], Grid R-CNN [16], Cascade R-CNN [17] and R-FCN [18]. In contrast, other algorithms directly classify and locate the object in one step, called single-stage detection algorithms. These algorithms do not require region extraction and therefore detect objects faster. One-stage algorithms mainly include the SSD [19], DSSD [20], RetinaNet [21] and YOLO series [22], [23], [24], [25]. These methods treat object detection as a regression problem and directly use a neural network to detect and locate objects from the whole image. In 2018, Liu et al. proposed CornerNet, which is a typical anchor-free detector. Anchor-free detectors can be roughly divided into anchor-point detection and key point detection. The anchor-point detectors, such as Densebox [26], Unitbox [27], FCOS [28], FSAF [29] or FoveaBox [30], encode the ground-truth boxes as anchor points with corresponding point-to-boundary distances, where anchor points are pixels on the feature pyramid maps and their positions are associated with features. Key-point detectors, such as CornerNet [31], ExtremeNet [32], CenterNet [33], decode the key points into prediction boxes by predicting the positions of several key points of the bounding box.

However, the detection effect is slightly inferior. Researchers began to apply deep learning to underwater object recognition. Li et al. [34] first applied deep CNN to underwater detection and constructed an Image-CLEF dataset. Chen et al. [35] proposed a novel sample-weighted hyper-network to address the blurring of underwater images under severe noise interference. Wei et al. [36] built a generalized model to address the complex environment in underwater object detection by simulating data augmentation strategies for overlapping, occluded, and blurred objects. Zeng et al. [37] proposed that the joint training of Faster R-CNN and adversarial network can effectively prevent the detection of fixed features generated by network overfitting. In [38], YOLOv4 is modified by replacing the upsampling module with a deconvolution module and by incorporating depthwise separable convolution into the network. And use image enhancement during the pre-training stage to obtain better detection performance. Aiming at the underwater dynamic target tracking problem, Cao et al. [39] studied an autonomous underwater vehicle tracking control method based on trajectory prediction. The algorithm part uses the YOLO v3 network to determine the target in a sonar image and obtain the position of the target. Yu et al. [40] integrated the Transformer module with YOLOv5s and introduced an attention mechanism to propose a novel TR-YOLOv5s network to meet the accuracy and efficiency requirements of underwater images. Lei et al. [33] proposed to use the Swin Transformer as the backbone network of YOLOv5, and at the same time adopted a variety of data enhancement methods, which significantly improved the detection accuracy of underwater objects. However, due to the harsh underwater environment, current underwater object detection algorithms still face various challenges in practice, such as poor quality, loss of visibility and weak contrast, etc. These factors may seriously hinder underwater object detection. DETR [41] is a target detection model developed by the Facebook research team by cleverly using the Transformer architecture, which not only simplifies the target detection process but also is an important step in applying Transformer to computer vision. Deformable DETR [42] overcomes the shortcomings of slow convergence of DETR and poor detection of small objects and becomes a new detection paradigm.

However, the accuracy of the above algorithm is not good enough, especially in complex underwater scenes, and the image degradation will lose many features. For example, the colour information of sea urchins, scallops and other creatures are relatively stable, but the texture information is easily destroyed. The texture information of creatures such as sea cucumbers has the strong resolution, but the color information will be destroyed due to insufficient lighting and other reasons. At the same time, the ocean space is huge, and the scale of objects is often tiny. Subjects may be blurred or incomplete due to underwater light scattering and sediment, resulting in loss of features. In response to the above challenges, this paper proposes a new object detection algorithm based on the Faster R-CNN algorithm.

(1) Given the low quality of underwater imaging and the low detection accuracy caused by the complex underwater environment, the Swin Transformer [43] containing the multi-head attention mechanism is used as the backbone
network for image feature extraction to enhance the ability of the network to acquire features.

(2) In view of the problem of different sizes and shapes of underwater objects, which leads to the low detection accuracy of the network model, PAN(Path Aggregation Network) [44] is used to more fully combine the deep features rich in semantic information and the shallow features rich in location information and detailed information, thereby improve the multi-scale feature fusion ability of the model.

(3) To solve the region mismatch problem caused by the quantization rounding operation in ROI(Region of Interest) pooling, ROI align is used to generate a fixed size output so that the model can obtain more accurate candidate regions and also improve the ability of the network to detect small object defects.

(4) For the problem of class imbalance between simple samples and difficult samples, the OHEM(Online Hard Example Mining) [45] algorithm can automatically select difficult samples for training, improving difficult samples’ detection performance.

The rest of this paper is organized as follows. Section 2 introduces the architecture of the Faster R-CNN model and the improved approach adopted in this paper. Section 3 introduces the data set, experimental environment, experimental methods and experimental results adopted in this paper. In Section 4, the specific ways of improving the different modules are introduced. A discussion is conducted in Section 4 on the experimental results and the limitations of the proposed method. Finally, in section 5, we concluded this paper.

II. IMPROVED FASTER R-CNN NETWORK

A. OVERVIEW OF FASTER R-CNN

Faster RCNN is a typical representative of the two-stage detection model, which integrates the region generation network RPN and the Fast R-CNN network model. They are in a parallel relationship, and each can be trained end-to-end so that the classification confidence and localization regression box can be detected.

The basic structure of Faster R-CNN is shown in Figure 1. First, the model extracts features from the input image and generates feature maps through a series of convolutional layers such as VGG [46], ResNet [47] and other feature extraction networks [48], and then inputs the feature maps into the region proposal network to generate candidate regions. The most significant difference between Faster R-CNN and the previous two-stage detection algorithm is that it uses the RPN network. In RPN, the input feature map is traversed through a set of $3 \times 3$ convolution kernels, k anchor boxes of different scales are used at each position on the feature map, and the fully connected layer classifies the anchor box to determine the anchor. The probability that the box belongs to the target and the frame regression is used to correct the anchor box to make it more in line with the target scale. Then, the candidate region generated by the RPN layer and the last layer feature map generated by the convolution layer is input to the region of interest pooling layer, and the feature map of the candidate region is normalized to a fixed size. Finally, the feature maps of these candidate regions are again passed through the fully connected layer to achieve category classification and bounding box regression to obtain more accurate bounding box prediction positions.

The network structure of RPN is a typical fully convolutional structure. The convolutional layer and the activation layer constitute the entire RPN network model, its input can be a feature map of any size, and finally, a series of rectangular boxes are output, which is candidate boxes. In these large number of candidate box, there is often an overlap of candidate frames, so non-maximum suppression (NMS) is used to remove redundant candidates’ areas. Therefore, the fundamental role of the RPN network is to locate the target to be detected initially. Compared with the previous selective search [49] method, it takes about two seconds to four seconds to detect the candidate region of the picture. RPN can generate candidate regions faster, dramatically reduces the time for detecting areas, and improves efficiency.

B. PROPOSED MODEL

1) BACKBONE NETWORK BASED ON SWIN TRANSFORMER

The underwater image will affect the quality of the image due to insufficient light and more suspended matter in the water,
making it difficult for the general CNN feature extraction backbone network to extract image features effectively. The Transformer [50], using the self-attention mechanism, can highlight the features of the detected target and weaken the background features. Originally, Transformers were widely used in the field of natural language processing. ViT [51] uses it creatively in the field of computer vision. However, there is a natural difference between natural language and images, and the application of Transformers in the image field faces two problems. First, when Transformer is applied to natural language, the basic element of the input is a fixed-size token. At the same time, in computer vision, objects may vary greatly in scale, and the performance of the Visual Transformer may not be excellent in different scenarios. Second, when applying Transformer in natural language, the computational complexity is related to the square of the token. At the same time, in computer vision, if the input feature map is a \(56 \times 56\) feature map, it will involve more than 3000 length and width matrix operations and computing. The amount has become very large, which is unacceptable. The above reasons make it difficult for ViT to become a general backbone network.

As a pure transformer architecture, Swin Transformer has the most significant contribution of proposing a backbone that can be widely applied to all computer vision fields. Most of the hyperparameters common in CNN networks can also be manually adjusted in Swin Transformer. For example, the number of network blocks and layers per block can be adjusted, the size of the input image, etc. While introducing the idea of locality for self-attention calculation in a single window, the idea of shifted windows is also proposed to communicate information between different windows. Through the above method, the computational complexity is linearly related to the size of the input image.

With the deepening of the network, the feature map generated by maintaining the same downsampling operation in the previous ViT is an undivided whole (Figure 2a). On the contrary, the Swin Transformer imitates CNN and adopts the method of hierarchical architecture. During the initialization stages, the input image is segmented into non-overlapping patches, and adjacent patches are gradually merged into deeper transformer layers. By computing self-attention using non-overlapping windows, the computational complexity changes from quadratic to linear. However, this division of pictures will reduce the global information connection. To solve this problem, Swin Transformer proposed the shifted windows method, as shown in Figure 2b. The shifted windows can fuse the information between different windows, which significantly enhances the ability of global modelling (Figure 2c). This is also the main difference from the original transformer architecture.

This architecture has four stages of getting feature maps (Figure 3a), and each stage contains Swin Transformer blocks (Figure 3b). The Swin Transformer Block is the algorithm’s core point, consisting of a window multi-head self-attention (W-MSA) and a shifted-window multi-head self-attention (SW-MSA) layer in Figure 3b. For this reason, the number of layers of the Swin Transformer should be an integer multiple of 2, one for W-MSA and one for SW-MSA. It also can be seen that each Swin Transformer block consists of a LayerNorm (LN) layer, a multi-head self-attention module, residual connections, and a multilayer perceptron (MLP) with two fully connected layers with GELU nonlinearity. The window-based multi-head self-attention (WMSA) module and the shifted window-based multi-head self-attention (SW-MSA) module are applied in the two successive transformer blocks, respectively. Based on such window partitioning mechanism, the process of calculating the feature map in...
FIGURE 3. (a) The architecture of a Swin Transformer. (b) Two successive Swin transformer blocks. The first is the W-MSA block, and the second is the SW-MSA block.

the continuous Swin Transformer blocks is shown below:

\[
\hat{z}^{l} = W - \text{MSA}(LN(z^{l-1})) + z^{l-1},
\]

(1)

\[
z^{l} = \text{MLP}(LN(\hat{z}^{l})) + z^{l-1},
\]

(2)

\[
\hat{z}^{l+1} = \text{SW} - \text{MSA}(LN(z^{l})) + z^{l},
\]

(3)

\[
z^{l+1} = \text{MLP}(LN(\hat{z}^{l+1})) + z^{l+1},
\]

(4)

where \(z^{l+1}\) and \(z^{l}\) denote the outputs of the (S)W-MSA model and the MLP module of the \(l\)th block, respectively.

Compared with the Multi-Head Self Attention (MSA) mechanism in the traditional ViT, the W-MSA in the Swin Transformer uses a window as a unit to perform calculations in it, which greatly reduces the amount of calculation. At the same time, MSA has no cross-window connection, so SW-MSA needs to provide different window segmentation methods after W-MSA to realize cross-window information exchange.

Let’s take Figure 4a as an example, assuming that the input feature map size is 56 \times 56 and the window size \(W\) is set to 7. The W-MSA block divides the feature map into 49 patches in the first Swin Transformer block, as shown in Figure 4b. The multi-head self-attention calculation is limited to 49 pixels in red windows, but the relationship between these red windows is not considered. However, there is a lack of connections across non-overlapping windows, resulting in patches in different windows having no interaction with each other and significantly limiting the model performance.

To solve this problem, an SW-MSA is added. As shown in Figure 4c, the top 3 \times 56 pixel bar is moved to the bottom, and the left 3 \times 56 pixel bar is transferred to the right. The window dividing line is moved down and to the right by round down of \((W/2)\) pixels, respectively, as shown by the green grids in Figure 4d. In this way, the pixels that belonged to different patches before can communicate with each other to achieve the ability of global modelling.

2) IMPROVEMENT OF MULTI-SCALE FEATURE FUSION

The low-level feature map extracted by the backbone network contains more localization details, while the top-level feature map contains more feature information. The original Faster R-CNN algorithm uses the top-level features extracted by the backbone network for prediction, which makes Faster R-CNN unable to use the underlying information for accurate positioning. The FPN [52] algorithm proposes to use both the high resolution of the low-level features and the high-semantic information of the high-level features and achieve the prediction effect by fusing the features of these different layers. However, in the FPN algorithm (Figure 5a), because of the bottom-up process, the shallow features need to go through dozens or even more than one hundred network layers to the top layer (red arrow). Obviously, after so many transmission layers, the shallow feature information is seriously lost. The feature layer path from the bottom layer to the topmost layer is too long, which increases the difficulty of accurately locating the information. On this basis, PAN (Figure 5b) proposes to add a bottom-up path augmentation (green arrow) so that shallow features are connected to \(P_2\) through the bottom layer of FPN and then from \(P_2\) along the bottom-up path augmentation passed to the top layer, which can better retain the shallow feature information. Here \(N_2\) and \(P_2\) represent the same feature map. But \(N_3, N_4\), and \(N_5\) differ from \(P_3, P_4\), and \(P_5\). \(N_3, N_4,\) and \(N_5\) are the result of the fusion of \(P_3, P_4,\) and \(P_5\).

The detailed structure of bottom-up path augmentation is shown in Figure 5c, a conventional feature fusion operation. Here, \(Ni\) is after convolution with a convolution kernel size...
of $3 \times 3$ and a stride of 2, the size of the feature map is reduced to half of the original. Then do the add operation with the feature map of $P_{i+1}$, and the result obtained is passed through a convolutional layer with a convolution kernel size of $3 \times 3$ and stride of 1 to obtain $N_{i+1}$.

3) IMPROVEMENT OF ROI POOLING
ROI (Region of interesting) refers to the candidate block on the feature diagram. In the Faster R-CNN algorithm, candidate blocks are generated through RPN, which are mapped on the feature diagram to get ROI. ROI pooling is an operation to extract small feature maps from ROI. Its processing steps are as follows:

- ROI is mapped to the corresponding region position on the feature map.
- Because the ROI of different sizes needs to be changed to a fixed size of $N \times N$ in the end, the ROI is divided equally into $N \times N$ regions.
- Taking the maximum pixel value of each divided area is equivalent to performing a max pooling operation on each area as a “representative” of each area so that each ROI becomes $N \times N$ in size.

However, this method will bring about a loss of accuracy due to quantization errors. Let us illustrate with an example as follows. Assume that the feat stride of the backbone network used in the model is 16 (extraction through the backbone network, the image is reduced to 1/16 of the original image), the original image is 400 x 400, and the feature map size of the last layer is 25 x 25, after ROI pooling, the size of feature map is fixed to 5 x 5.

- There is a region proposal in the original image; the size is 200 x 200, so the size mapped to the feature map is 12.5 x 12.5 (200/16 x 200.16). Because of the rounding operation, the size of the mapped feature map becomes 12 x 12, which is called the first quantization operation.
- The final feature maps need to be fixed to 5 x 5; therefore, the previously obtained 12 x 12 region proposal is divided into 25 small regions of the same size, and the size of each small region is 2.4 x 2.4 (12/5 x 12.5). At this time, a rounding operation is also performed, which is called the second quantization operation, so the small area becomes 2 x 2.

In fact, after these processes, the candidate box obtained has a specific deviation from the original position returned.
from RPN, which will affect the accuracy of detection, especially the detection performance of small objects.

In this paper, we use ROI align to avoid that problem instead of rough ROI pooling. It differs from ROI pooling not by simply quantifying and then pooling but by using a regional feature aggregation approach to transform it into a continuous operation.

Cancel the two quantization operations in the ROI pooling (Figure 6) and use floating point calculations directly (the size obtained in the first time is $12.5 \times 12.5$, and the size obtained in the second time is $2.4 \times 2.4$). At the same time, a hyperparameter is set to indicate the number of sampling points in each area, that is, how many points are taken from each area to calculate the value “representing” this area, which is usually 4. The candidate region is divided into $z \times z$ (2 $\times$ 2 in the figure) cells, and each cell is also not quantified. Determine four positions of a sample point in each cell. The floating-point coordinates of the sampled points are calculated using a bilinear interpolation method to find the value of 4 positions. Then, the ROI output in a fixed dimension can be gotten. In each divided area, they take the maximum value of the four centre point pixel values as their “representatives”, changing the ROI to a $5 \times 5$ size.

4) ONLINE HARD EXAMPLE MINING

Many object detection ideas in computer vision are derived from the concept of image classification. Still, there is a natural gap between the images classification dataset and the object detection dataset, which is a severe imbalance between the samples of object detection. In the object detection task, there are the following sample categories (Figure 7):

- Positive sample: the image area within the ground truth, that is, the image background block.
- Negative sample: the image area other than the ground truth, that is, the image background area.
- Easy-to-classify positive sample: positive samples that are easy to classify correctly.
- Easy-to-classify negative sample: negative samples that are easy to classify correctly.
- Hard-to-classify positive sample: positive samples that are easily misclassified as negative samples.
- Hard-to-classify negative sample: negative samples that are easily misclassified as positive samples.

In the whole training process, easy-to-classify positive samples and easy-to-classify negative samples account for a very high proportion of the total samples, and the loss function value is relatively small. Still, the accumulated loss function value will dominate the entire model. However, the loss function value of a single sample in the training process of hard positive samples and hard negative samples is higher, but the total number of samples is small.

Taking Faster R-CNN as an example, about 20,000 anchors will be generated in the RPN part, but usually, there are about ten objects in a picture, so only about 100 anchors are positive samples in the end. As a result, the ratio of positive and negative samples is around 1:200, so there is a severe imbalance in the samples. The target detection algorithm mainly considers the positive samples corresponding to the real target, and adjusts the network parameters according to its loss during the training process. Suppose there are a large number of negative samples involved in training. In that case, the loss of positive samples will be overwhelmed, thereby reducing the convergence efficiency and detection accuracy of the network.

The OHEM (Figure 8) algorithm is mainly for automatically selecting difficult samples in the training process. In practice, the original ROI Network is expanded into two ROI Networks, and the two ROI Networks share parameters. The former ROI Network only has forward operations, mainly used to calculate the loss; the latter ROI Network includes forward and backward operations, using hard example as input, calculating the loss and returning the gradient.
The problem of class imbalance of data does not need to be solved by setting the ratio of positive and negative samples. With the increase of the data set, the improvement of the algorithm is more obvious. Finally, this paper uses the model of Swin Transformer on the large ImageNet dataset as the pre-training model, and adopts the incremental learning rate warm-up strategy (Figure 9).

III. EXPERIMENTS

A. DATA SET

The experimental dataset uses the underwater optical image dataset provided by the URPC official website, which includes underwater images of sea cucumbers, sea urchins, starfish, and scallops and annotations of corresponding images. The URPC dataset (Figure 10) has a total of 5543 images, of which 41,441 target labels are annotated, 22,343 echinus category targets, 6841 starfish category targets, and 5537 holothurian category targets, and 6720 scallop category targets.

To keep the consistency of the data distribution, the dataset is randomly divided into training set and test set with a ratio of 8:2. The training set contains 4434 images, and the test set has 1109 images.

B. MODEL EVALUATION METRICS

In this paper, the common precision (Pr), recall (Re), average precision (AP), and mean average precision (mAP) in target detection are used as the performance indicators of the evaluation algorithm, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) is used in the definition of these four criteria in Eqs. Of which the area can intuitively reflect the mAP under the P-R curve. To obtain mAP, it is first necessary to calculate the AP value of each type of underwater target under a fixed IOU threshold and then calculate the average value of all types of AP values.

\[
Pr = \frac{TP}{FP + TP},
\]

\[
Re = \frac{TP}{FN + TP},
\]

\[
AP = \frac{1}{I} \int Re(dRe),
\]

\[
mAP = \frac{1}{C} \sum_{c=1}^{C} AP(c),
\]

C. MODEL EVALUATION METRICS

In this paper, the common precision (Pr), recall (Re), average precision (AP) and mean average precision (mAP) in target detection are used as the performance indicators of the evaluation algorithm, true positive (TP), true negative (TN), false positive (FP) and false negative (FN) is used in the definition of these four criteria in Eqs. Of which the area can intuitively reflect the mAP under the P-R curve. To obtain mAP, it is first necessary to calculate the AP value of each type of underwater target under a fixed IOU threshold and then calculate the average value of all types of AP values.

Specifically, the Stochastic Gradient Descent (SDG) optimization algorithm is used to solve the model. The training epochs were set to 100, the batch size was set to 10, the initial learning rate was set to 0.01, the weight decay was set to 0.0005, and the SGD momentum was set to 0.9 (Table 2).

D. EXPERIMENT RESULT

The Swin Transformer is divided into four different models according to the depth of the model: Swin-T, Swin-S, Swin-B, Swin-L. The parameter settings are shown in Table 4. As the depth and the number of channels of the first-stage hidden layer increase, the number of model parameters and model size also increase linearly. We experiment with Swin Transformer models of different depths as the backbone network of Faster R-CNN. As can be seen from the table, the
The dataset contains four biological categories, which are (a) echinus, (b) starfish, (c) holothurian, and (d) scallop. (e) bar chart of the number of targets in each class.

**TABLE 1.** Hardware configuration parameters.

| CPU  | GPU  | RAM | CUDA | CUDNN | Python | Pytorch |
|------|------|-----|------|-------|--------|---------|
| RTX  | 3090 | 64G | 11.3 | 8.2 | 3.713 | 1.110 |

**TABLE 2.** Hyperparameter settings of network training.

| Optimizer | Training Epochs | Batch size | Learning Rate | Weight Decay | Momentum |
|-----------|-----------------|------------|---------------|--------------|----------|
| SGD       | 100             | 10         | 0.01          | 0.0005       | 0.9      |

**TABLE 3.** Experimental results of different Swin transformer models.

| Backbone | Layer Numbers | Channel | Number of Parameters (M) | Size of Model (MB) | mAP (%) | FPS |
|----------|---------------|---------|--------------------------|-------------------|---------|-----|
| Swin-T   | (2,2,6,2)     | 96      | 48.3                     | 368               | 74.2    | 25.3 |
| Swin-S   | (2,2,18,2)    | 96      | 69.6                     | 531               | 74.9    | 18.1 |
| Swin-B   | (2,2,18,2)    | 128     | 107.6                    | 822               | 75.4    | 17.0 |
| Swin-L   | (2,2,18,2)    | 192     | 216.2                    | 1650              | 76.8    | 12.8 |

The mean AP (mAP) of the model increases with the increase of model depth and width. There is a 2.6% improvement between the largest model and the smallest model, Swin-L has a 1.2% performance improvement over Swin-B, and the FPS is only 4.2 lower than that of Swin-L. Compared with the performance gain, the speed reduction is acceptable to us, so we choose Swin-T as the improved benchmark and conduct further experiments. Therefore, we choose Swin-L as the backbone network of the algorithm (Table 3).

Figure 10a shows the change curve of the loss value, including RPN classification loss, RPN bbox loss, classification loss, bbox loss. It shows that with the increase in the number of iterations, all the losses are steadily decreasing, the accuracy rate and mAP(Figure 11b) are steadily increasing.

Figure 11c shows the row-normalized confusion matrix. To more intuitively display each category’s recognition rate and false positive rate, we normalize the values in each row of the confusion matrix by dividing the total number of corresponding categories.

Let’s take the result of the starfish row as an example. The row direction represents the true label, and the column direction represents the predicted category. The probability of starfish being falsely detected as echinus is 2%, and the probability of being missed is 6%. The category with the highest missed detection rate is holothurian, reaching 18%. Because holothurian is very similar to the environment, which leads to missed detection, looking at the bottom row of the confusion matrix alone, we can see that the false positive rate of echinus is the highest, reaching 36%. Because there are many marine plants on the seabed, which are very similar to the shape of echinus, the algorithm identifies the aquatic plants as echinus.

In this paper, the performance of the Faster-RCNN network under various improvements is tested by ablation experiments. The Swin Transformer is used as the backbone network of our algorithm to extract feature, improve the multi-scale fusion network, use ROI align to eliminate quantization errors, and finally use the OHEM algorithm to enhance the training effect of the network. When Swin Transformer is used as the backbone network model, mAP is improved by 1.99%. Through multi-scale feature fusion, mAP is improved by 0.81%. With the addition of ROI align and OHEM, mAP is
FIGURE 11. The variation curves of the loss values: (a) RPN classification loss; RPN bbox loss; classification loss; bbox loss. (b) The curve of mAP. (c) Confusion matrix.

| Method          | Backbone    | mAP (%) | Size of Model (MB) | FPS |
|-----------------|-------------|---------|--------------------|-----|
| Cascade R-CNN   | ResNet50    | 70.39   | 46                 |     |
| Sparse R-CNN    | ResNet50    | 68.04   | 23                 | 9   |
| Grid R-CNN      | ResNet50    | 71.97   | 515                | 6   |
| Deformable-DETR | ResNet50    | 72.10   | 482                | 8.9 |
| YOLO V4         | CSPDarknet53| 64.21   | 123                | 9   |
| YOLO V5         | Focus+CRF+CSP+5P | 65.32 | 135                | 8.1 |
| RetinaNet       | ResNet50    | 66.19   | 267                | 6.8 |
| Ours            | Swin-L      | **80.54** | **1650**          | **8** |

The experimental results of different algorithms.

IV. DISCUSSION

The experimental results show that the algorithm based on this paper presents high detection accuracy and acceptable detection speed in harsh underwater scenes. In Figure 13a, our algorithm detected not only all objects but also unlabeled objects (blue arrows) in the ground truth farther away in the scene. In Figure 13b, it can see that the algorithm can detect different objects. In Figure 13c, both the tiny target increased by 0.11% and 0.81%. Excellent backbone network can extract richer features, ROI align reduces quantization loss, and OHEM makes training more effective. Each module we improve has a significant gain on the overall detection performance, demonstrating the effectiveness of these methods.

To demonstrate the superiority of the improved method based on Faster Cascade R-CNN [16], Sparse R-CNN [13], Grid R-CNN [15], Deformable-DETR [30], YOLO V4 [23], YOLO V5, RetinaNet [21] and are used as other models to conduct comparative experiments. The experimental results are shown in Table 5. Compared with other models, the improved Faster R-CNN model has the highest mAP. The mAP of the improved Faster R-CNN model (80.54% mAP) exceeds other two-stage algorithms and one-stages.
FIGURE 12. The dataset contains four biological categories, which are (a) echinus, (b) starfish, (c) holothurian, and (d) scallop. (e) bar chart of the number of targets in each class.

in the long-range and the large target in the close range is accurately detected, and the unmarked target is also detected. But in group d, because the target and the environment are too similar, the algorithm does not detect all the targets (green arrows).

To better demonstrate our proposed algorithm, class activation maps are visualized by Grad-CAM [53], which uses the back-propagated gradient information to generate a rough localization map highlighting the most sensitive regions in the image for detection (Figure 14). The brightest area in the figure represents that the network is most sensitive to this area. These areas are distributed on our detection target, which proves that the Swin Transformer has strong feature extraction ability and can improve the detection performance of our algorithm.

The experimental results show that Swin Transformer has obvious advantages as a backbone network for feature extraction. When the complexity of the network is increased, the detection performance will be significantly improved. At the same time, our experiments also show that Swin Transformer has some obvious shortcomings as a backbone network,
such as the obtained weight file is relatively large, and when a complex model is used, the inference speed will be reduced.

V. CONCLUSION

For the complex underwater environment, we propose an improved Faster R-CNN to enhance the accuracy of underwater target recognition. We have improved four of these sections. First, the Swin Transformer is used as the backbone network of Faster R-CNN to obtain better feature information. Second, the multi-resolution feature fusion method is improved, which can more effectively fuse images of different resolutions. Third, use ROI align to replace ROI pooling to eliminate quantization errors. Fourth, adopt OHEM to solve the problem of sample imbalance. The detection effect of the Faster R-CNN model when using Swin Transformers of different sizes as the backbone network is compared through experiments, and the improvement of the model by different improvement strategies is also compared through ablation experiments. Finally, the algorithm in this paper is compared with other algorithms, which proves that the algorithm is advanced. The experimental results show that the detection results of the improved Faster R-CNN model in complex underwater environments are improved with the above improvements.

However, it should be noted that the detection speed of our model is not fast enough relative to the single-stage detection algorithm, and the resulting model size is relatively large. And there are many blurry images in the URPC dataset, and we do not specifically design these images to improve the detection performance. In the following work, our research not only focuses on compressing our model, speeding up detection, collecting other underwater target data to expand the dataset, and using data augmentation techniques to improve robustness, but also need design a special module to deal with the effects of blurred pictures. We also can use a single-stage detection model to reduce the complexity of the network. And use model compression, network pruning and weights quantization param to lighten our network. Otherwise, the complexity of the network will not be increased. And we noticed that many of the data sets are small targets, and our next work will also be carried out for small objects.

REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, Nov. 2004.
[2] T. Watanabe, S. Ito, and K. Yokoi, “Co-occurrence histograms of oriented gradients for human detection,” IPSJ Trans. Comput. Vis. Appl., vol. 2, pp. 39–47, Dec. 2010.
[3] J. C. Platt, “A fast algorithm for training support vector machines,” J. Inf. Technol., vol. 2, no. 5, pp. 1–28, 1998.
[4] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of online learning and an application to boosting,” J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119–139, Aug. 1997.
[5] X. U. X. Shi and J. L. Zhang, “Feature extraction of underwater targets using generalized S-transform,” J. Comput. Appl., vol. 32, pp. 280–282, Dec. 2012.
[6] W. Ma, “Underwater target detection based on the polarization features,” Electron. Meas. Technol., vol. 37, pp. 90–92, Jan. 2014.
[7] L. X. H. L. Jiao and T. Chen, “Detection and recognition of underwater target based on feature matching,” Mod. Electron. Techn., vol. 34, no. 4, pp. 73–76, 2011.
[8] L. I. K. Liu and M. Dun, “Algorithm for recognition of underwater small target based on shape characteristic,” Ship Sci. Technol., vol. 34, no. 1, Jan. 2012.
[9] R. Zhang, S. Yang, Q. Zhang, L. Xu, Y. He, and F. Zhang, “Graph-based few-shot learning with transformed feature propagation and optimal class allocation,” Neurocomputing, vol. 470, pp. 247–256, Jan. 2022.
[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.
[11] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2015, pp. 1440–1448.
[12] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 1–9.
[13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. ICCV, 2017, pp. 2961–2969.
[14] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, “Sparse R-CNN: End-to-end object detection with learnable proposals,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jan. 2021, pp. 14454–14463.
[15] H. Zhang, H. Chang, B. Ma, N. Wang, and X. Chen, “Dynamic R-CNN: Towards high quality object detection via dynamic training,” in Proc. Eur. Conf. Comput. Vis., 2020, pp. 260–275.
[16] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, “Grid R-CNN,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7363–7372.
[17] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 6154–6162.
[18] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 1–9.
[19] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot MultiBox detector,” in Proc. ECCV, 2016, pp. 21–37.
[20] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, “DSSD: Deconvolutional single shot detector,” 2017, arXiv:1701.06659.
[21] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Comput. Vis. Pattern Recognit., Jun. 2016, pp. 779–788.
[23] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” 2020, arXiv:2004.10934.
[24] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.
[25] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 7263–7271.
[26] L. Huang, Y. Yang, Y. Deng, and Y. Yu, “DenseBox: Unifying landmark localization with end to end object detection,” 2015, arXiv:1509.04874.
[27] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, “UnitBox: An advanced object detection network,” in Proc. 24th ACM Int. Conf. Multimedia, Oct. 2016, pp. 516–520.
[28] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9627–9636.
[29] C. Zhu, Y. He, and M. Savvides, “Feature selective anchor-free module for single-shot object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 840–849.
[30] T. Kong, F. Sun, H. Liu, Y. Jiang, L. Li, and J. Shi, “FoveaBox: Beyond anchor-based object detection,” IEEE Trans. Image Process., vol. 29, pp. 7389–7398, 2020.
[31] H. Law and J. Deng, “CornerNet: Detecting objects as paired keypoints,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 734–750.
[32] X. Zhou, J. Zhuo, and P. Krahenbuhl, “Bottom-up object detection by grouping extreme and center points,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2019, pp. 850–859.
[33] X. Zhou, D. Wang, and P. Krahenbuhl, “Objects as points,” 2019, arXiv:1904.07850.
[34] X. Li, M. Shang, J. Hao, and Z. Yang, “Accelerating fish detection and recognition by sharing CNNs with objectness learning,” in Proc. OCEANS, Shanghai, China, Apr. 2016, pp. 1–5.

[35] L. Chen, F. Zhou, S. Wang, J. Dong, N. Li, H. Ma, X. Wang, and H. Zhou, “SWIPENET: Object detection in noisy underwater images,” 2020, arXiv:2010.10006.

[36] X. Wei, L. Yu, S. Tian, P. Feng, and X. Ning, “Underwater target detection with an attention mechanism and improved scale,” Multimedia Tools Appl., vol. 80, no. 25, pp. 33747–33761, Oct. 2021, doi: 10.1007/s11042-021-11230-2.

[37] L. Zeng, B. Sun, and D. Zhu, “Underwater target detection based on faster R-CNN and adversarial occlusion network,” Eng. Appl. Artif. Intell., vol. 100, Apr. 2021, Art no. 104190, doi: 10.1016/j.engappai.2021.104190.

[38] L. Chen, M. Zheng, S. Duan, W. Luo, and L. Yao, “Underwater target recognition based on improved YOLOv4 neural network,” Electronics, vol. 10, no. 14, p. 1634, Jul. 2021, doi: 10.3390/electronics10141634.

[39] X. Cao, L. Ren, and C. Sun, “Dynamic target tracking control of autonomous underwater vehicle based on trajectory prediction,” IEEE Trans. Cybern., early access, Aug. 1, 2022, doi: 10.1109/TTCYB.2022.3189688.

[40] Y. Yu, J. Zhao, Q. Gong, C. Huang, G. Zheng, and J. Ma, “Real-time underwater maritime object detection in side-scan sonar images based on transformer-YOLOv5,” Remote Sens., vol. 13, no. 18, p. 3555, Sep. 2021, doi: 10.3390/rs13183555.

[41] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in Proc. Eur. Conf. Comput. Vis., Jun. 2020, pp. 213–229.

[42] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable DETR: Deformable transformers for end-to-end object detection,” 2020, arXiv:2010.04459.

[43] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted Windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10012–10022.

[44] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 8759–8768.

[45] A. Shrivastava, A. Gupta, and R. Girshick, “Training region-based object detectors with online hard example mining,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 761–769.

[46] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[47] X. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 770–778.

[48] R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu, “Deep-IRTarget: An automatic target detector in infrared imagery using dual-domain feature extraction and allocation,” IEEE Trans. Multimedia, vol. 24, pp. 1735–1749, 2022.

[49] J. R. R. Uijlings, E. A. Van De Sande, Koen, G. Theo, and W. M. S. Arnold, “Selective search for object recognition,” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, Sep. 2013.

[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–11.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, R. Zhang, L. Xu, Z. Yu, Y. Shi, C. Mu, and M. Xu, “Deformable DETR: Deformable transformers for end-to-end object detection,” 2020, arXiv:2010.04459.

[52] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2117–2125.

[53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.