water

MDPI

Article

Underwater Biological Detection Algorithm Based on Improved Faster-RCNN

Pengfei Shi [1,2], Xiwang Xu [2], Jianjun Ni [2,*], Yuanxue Xin [2,*], Weisheng Huang [2] and Song Han [2]

[1] Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University, Changzhou 213022, China; shipf@hhu.edu.cn
[2] College of Internet of Things Engineering, Hohai University, Changzhou 213022, China; 211620010057@hhu.edu.cn (X.X.); wilson@hhu.edu.cn (W.H.); 18014754795@163.com (S.H.)
* Correspondence: njjhhuc@gmail.com (J.N.); xinyx@hhu.edu.cn (Y.X.)

Abstract: Underwater organisms are an important part of the underwater ecological environment. More and more attention has been paid to the perception of underwater ecological environment by intelligent means, such as machine vision. However, many objective reasons affect the accuracy of underwater biological detection, such as the low-quality image, different sizes or shapes, and overlapping or occlusion of underwater organisms. Therefore, this paper proposes an underwater biological detection algorithm based on improved Faster-RCNN. Firstly, the ResNet is used as the backbone feature extraction network of Faster-RCNN. Then, BiFPN (Bidirectional Feature Pyramid Network) is used to build a ResNet–BiFPN structure which can improve the capability of feature extraction and multi-scale feature fusion. Additionally, EIoU (Effective IoU) is used to replace IoU to reduce the proportion of redundant bounding boxes in the training data. Moreover, K-means++ clustering is used to generate more suitable anchor boxes to improve detection accuracy. Finally, the experimental results show that the detection accuracy of underwater biological detection algorithm based on improved Faster-RCNN on URPC2018 dataset is improved to 88.94%, which is 8.26% higher than Faster-RCNN. The results fully prove the effectiveness of the proposed algorithm.

Keywords: deep learning; object detection; underwater detection

Citation: Shi, P.; Xu, X.; Ni, J.; Xin, Y.; Huang, W.; Han, S. Underwater Biological Detection Algorithm Based on Improved Faster-RCNN. *Water* **2021**, *13*, 2420. https://doi.org/10.3390/w13172420

Academic Editor: Gwo-Fong Lin

Received: 22 July 2021
Accepted: 1 September 2021
Published: 3 September 2021

1. Introduction

Underwater organisms are an important part of the underwater ecological environment and have received widespread attention. The underwater ecological protection organization conducts research on the distribution and living habits of underwater organisms through artificial diving and underwater robot shooting. However, low-quality underwater imaging makes it difficult for researchers to accurately discover underwater organisms. Therefore, there is an urgent need for an effective target detection algorithm to replace the eyes to detect underwater life.

Target detection is one of the important tasks in computer vision. It is a computer technology related to computer vision and image processing. It deals with specific types of semantic objects (such as people, cars, or animals) in digital images or videos. The research fields of target detection include edge detection [1], multi-target detection [2–4], salient target detection [5,6], and so on.

Traditional target detection methods have many shortcomings, such as poor recognition effect, low accuracy, slow recognition speed, etc. These problems make it difficult to perform effective underwater biological detection. In recent years, the rapid development of deep learning has brought huge breakthroughs in the field of target detection. The target detection algorithm based on deep learning has the advantages of high detection accuracy and strong robustness. It is widely used in environmental monitoring [7], autonomous driving [8], UAV scene analysis [9] and other scenarios.

However, due to the low quality of underwater imaging, complex underwater environment, the different sizes or shapes and overlapping or occlusion of underwater organisms, the general target detection algorithm based on deep learning does not have a good detection effect on underwater organisms. Therefore, this article will improve the target detection model based on deep learning so that it can effectively detect underwater organisms.

The R-CNN first generates candidate regions through selective search [10]. It uses CNN to extract features for the candidate regions. Thus, the accuracy of target detection is improved by replacing the traditional sliding window method. However, R-CNN has a large number of repeated calculations, which seriously affects the detection performance. Faster-RCNN aims at the problem of computational redundancy [11]. It chooses to extract features from the input image through CNN and extract candidate regions through selective search. In this way, all candidate regions can be obtained only through one CNN. Candidate areas reduce repeated calculations. In order to further improve detection speed, Faster-RCNN is proposed. Faster-RCNN [12] uses a regional suggestion network (RPN) instead of a selective search algorithm to filter out candidate regions. Therefore, the detection speed is further improved. The core process of Faster-RCNN detection is that the backbone feature extraction network is used to extract target features and generate candidate regions through the region suggestion network. Faster-RCNN will determine whether the candidate region contains a target and correct the size of the candidate region. Finally, the overall structure of the process Faster-RCNN, using ROI Pooling to classify candidate regions, is shown in Figure 1.

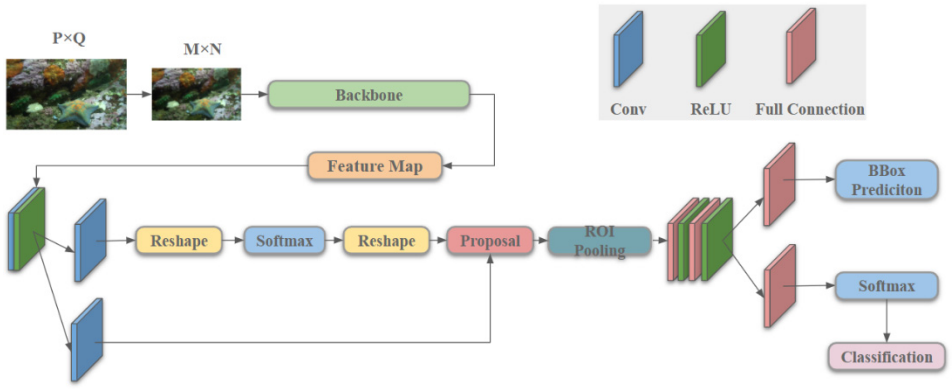

Figure 1. Overall structure of Faster-RCNN.

Based on the traditional Faster-RCNN, this article makes the following three improvements for the underwater biological detection scene.

1. Aiming at the problem of low-quality underwater imaging and low detection accuracy caused by different sizes and shapes of underwater organisms. ResNet is used to replace Faster-RCNN's VGG backbone feature extraction network. Then, BiFPN is added after ResNet to form a ResNet–BiFPN structure to improve the network model feature extraction ability and multi-scale feature fusion ability.
2. EIoU is instead of IoU in Faster-RCNN to reduce the proportion of redundant bounding boxes in the training data by adding centrality weights. Thus, the quality of bounding boxes is improved.
3. The K-means++ clustering is used to generate a more suitable anchor box for the underwater biological data set to improve the detection accuracy.

In this paper, Section 2 describes the materials and methods, mainly including the detail detection algorithm of the improved Faster-RCNN. Section 3 entails experimental results and discussion. Section 4 gives conclusions.

2. Materials and Methods

2.1. ResNet–BiFPN

Optical scattering leads to low underwater imaging quality, and underwater scenes are often very complex. Underwater rocks, water plants, etc., interfere with the extraction of target features. Underwater organisms of different sizes and shapes are also a test of the multi-scale feature fusion ability of the network model. Therefore, this paper selects ResNet [13] with strong feature extraction ability as the backbone feature extraction network of Faster-RCNN. BiFPN [14] is added after ResNet to enhance the multi-scale feature fusion ability of the network model.

ResNet designs a residual structure of identity mapping so that the gradient can be smoothly transmitted from the shallow layer to the deep layer. Through the structure, very deep neural networks can be trained to improve feature extraction capabilities. Compared with the VGG network model, the ResNet network model with a residual mechanism can better retain the shallow features and pass them to the deeper layers to participate in training.

For any input, the processed features are obtained through 5 stages (Stage0–Stage4) of ResNet. Among them, Stage0 can be regarded as the preprocessing stage of the input, and the following four stages are all composed of the bottleneck layer (BTNK, Bottleneck). The structure is relatively similar. Stage1 contains two bottleneck layers, and the remaining three stages contain 4, 6, and 3 bottleneck layers, respectively. (3, 224, 224) refers to the number of input channels, height, and width. In Stage0, the input of the form (3, 224, 224) passes through the convolution layer, the BN layer, the ReLU activation function, and the Max Pooling layer to obtain the output of the form (64, 56, 56). The two bottleneck layers correspond to two cases: the case where the number of input and output channels is the same corresponds to BTNK2, and the case where the number of input and output channels is different corresponds to BTNK1. As shown in the right part of Figure 2, firstly, the input of (C, W, W) passes through the 3 convolution blocks, and the result is added to the original input. Then, the ReLU activation function is used to get the output of BTNK2. Compared with BTNK2, BTNK1 has one more convolutional layer which can be used to match the different dimensions between the input and output.

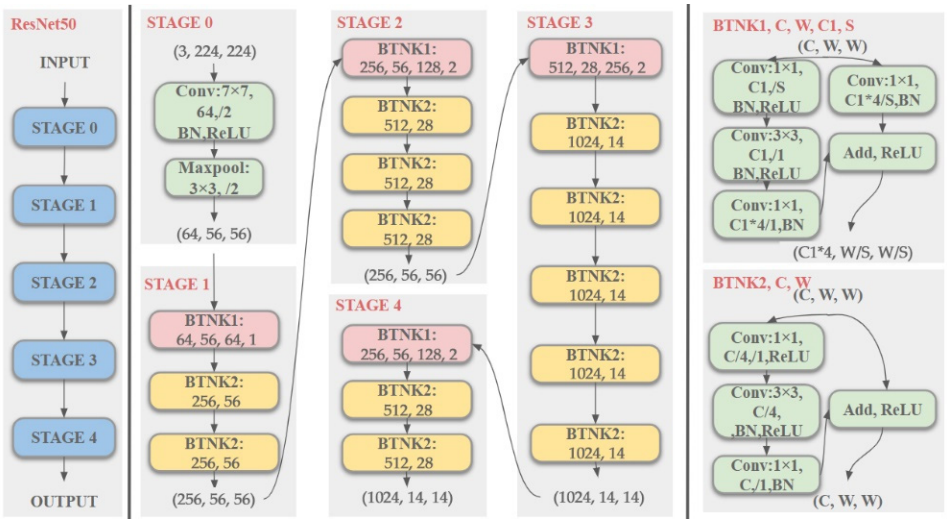

Figure 2. Overall structure of ResNet.

In 2020, the Google Brain team proposed the BiFPN structure in EfficientDet. Figure 3 shows the formation process of BiFPN; the author was inspired by the PANet structure. As shown in Figure 3, each layer is marked with a different color. The gray background part is the main part of the structure. The PANet structure is simplified to remove redundant nodes, as shown in Figure 3a,b. Then, the short-circuit structure is added based on

Simplified PANet, which is called BiFPN, as shown in Figure 3c. The BiFPN introduces learnable weights to learn the importance of different input features while repeatedly applying top-down and bottom-up multi-scale feature fusion.

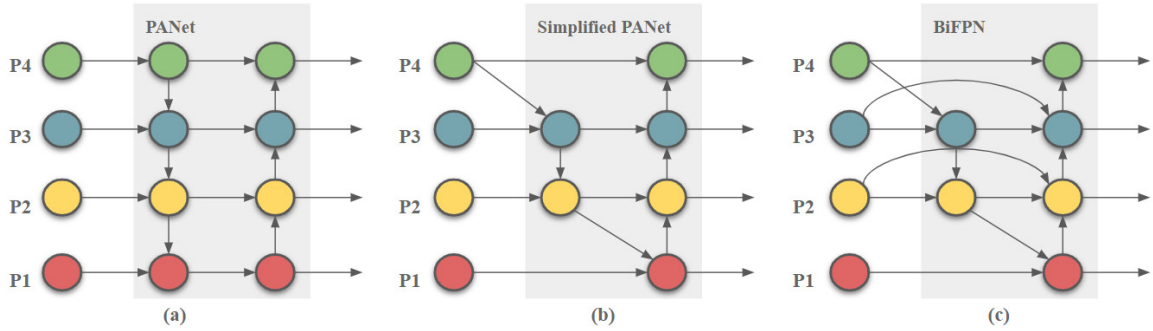

Figure 3. Formation process of BiFPN: (**a**) PANet; (**b**) simplified PANet; and (**c**) BiFPN.

To introduce learnable weights to learn the importance of different input features, it is only needed to multiply the features by a learnable weight

$$Y = \sum_i w_i \cdot I_i, \tag{1}$$

where w_i can be a scalar (for each feature), a vector (for each channel), or a multi-dimensional tensor (for each pixel); I_i is the input feature; and Y is the output feature. However, it is easy to cause training instability if w_i is not restricted, so use Softmax for each weight

$$Y = \sum_j \frac{e^{w_i}}{e^{w_j}}. \tag{2}$$

However, the actual calculation speed of Softmax is slower, so change it to

$$Y = \sum_j \frac{w_i}{\varepsilon + \sum_j w_j}. \tag{3}$$

Among them, $\varepsilon = 0.0001$ to avoid numerical instability. To ensure that the weight is greater than 0, the ReLU activation function is used before the weight. This article adopts Formula (3) to realize the weighting mechanism in BiFPN.

The specific implementation of ResNet–BiFPN is shown in Figure 4. The four groups of features P1–P4 with different scales are obtained through STAGE1–STAGE4 of the backbone feature extraction network ResNet50. The number of channels is adjusted by 1×1 convolution. P4 is transformed into feature ①, which is the same scale as P3 through up-sampling operation. And the result is stacked and convolved with the feature ②, which convolutes with P3. In the weight module, all stacking and convolution operations use Formula (3) for their input to learn the importance of different input features. Taking the features of P4 and P3 as an example, the stacking and convolution operations in the upper left corner will use the weight mechanism to determine whether to pay more attention to ① or pay more attention to ②. After the weighting mechanism is screened, the remaining features continue to perform the same stacking, convolution, and weight screening operations with the features after P2 convolution and the features after P3 convolution and jumper. This operation forms a BiFPN network.

In order to obtain better detection results, EfficientDet repeatedly stacks the BiFPN structure. Experiments have proved that to a certain extent, the stacking of BiFPN can bring about an increase in accuracy, but it will also lead to an increase in the number of parameters. Thus, the performance of the model is affected. This paper tests the stacking times of BiFPN to select the optimal stacking times under the Faster-RCNN model.

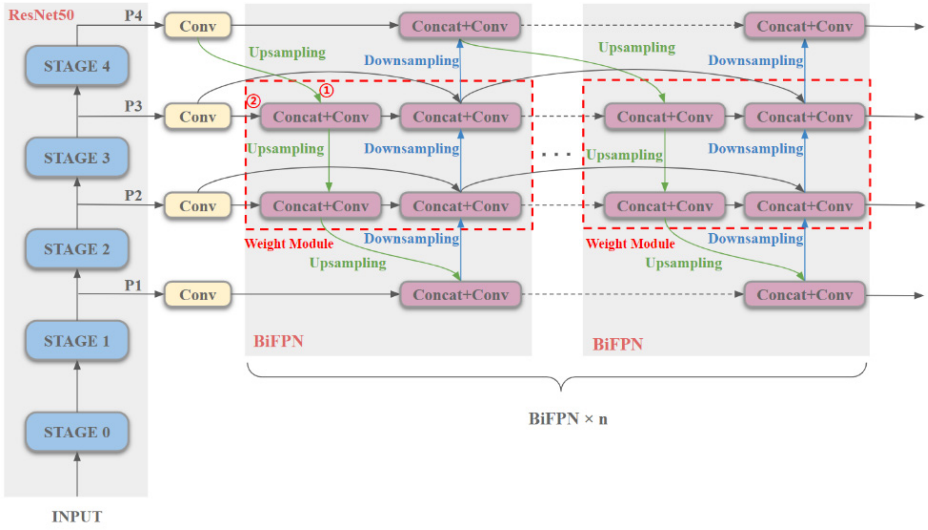

Figure 4. Implementation of ResNet-BiFPN.

2.2. EIoU (Effective Intersection over Union)

In the Faster-RCNN target detection algorithm, we let the model generate a large number of candidate boxes at a time. The boxes are sorted according to the confidence of each box. Then, the IoU is calculated between the boxes. Non-maximum suppression is used to determine which objects need to be found and which should be deleted.

However, the intersection over union design is not perfect. In fact, because the shape of the detection target is different, in addition to the characteristic information of the detection target, there are certain background information in the anchor box. Non-detection target information causes some interference to the model. The following Figure 5 is an example; the white box is the ground truth box (GT), and the red box and the blue box are the two prediction boxes Box A and Box B. According to the calculation formula of the intersection over union, the intersection over the union of Box A and GT is 0.71. The intersection over the union of Box B and GT is 0.65. If the prediction box is only selected based on IoU, Box A will naturally be selected here. However, it is apparent that Box A selects a lot of useless background information, and Box B selects more detection target information. Box B is a more effective sample.

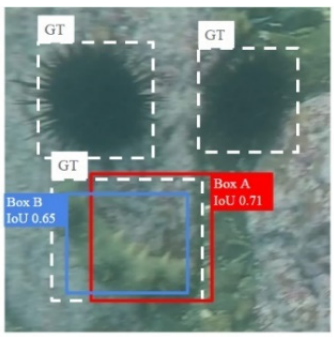

Figure 5. Example of IoU.

Therefore, this paper uses EIoU [15] (Effective Intersection over Union) instead of IoU in Faster-RCNN as the criterion for identifying positive and negative samples in the regional proposal network. EIoU uses "centrality" to measure the degree of a prediction box within the target, and centrality is used to indicate the standardized distance from the center of the prediction box to the center of the label box. EIoU believes that the prediction frame closer to the center of the label frame may contain more effective information about the detection target and should be considered more important.

Assume a bounding box $A = (\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2)$, where $\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2$ represent the horizontal and vertical coordinates of the upper left corner and the lower right corner of the bounding box, so the center point coordinates of the predicted box A are: $C_A = (\hat{x}_c, \hat{y}_c) = \left(\frac{\hat{x}_1+\hat{x}_2}{2}, \frac{\hat{y}_1+\hat{y}_2}{2}\right)$. The label box $GT = (x_1, y_1, x_2, y_2)$ corresponds to the prediction box A, where $\hat{x}_1, \hat{y}_1, \hat{x}_2, \hat{y}_2$ are the horizontal and vertical coordinates of the upper left corner and the lower right corner of the label box. Next, calculate the distance between C_A and the label box boundary while defining d_l, d_r, d_t, d_b as the distance from the center point C_A of the prediction box A to the four sides of the label box

$$\begin{aligned} d^l &= |\hat{x}_c - x_1| \\ d^r &= |x_2 - \hat{x}_c| \\ d^t &= |\hat{y}_c - y_1| \\ d^b &= |y_2 - \hat{y}_c|. \end{aligned} \tag{4}$$

Define the bounding box centrality weight W_A as

$$W_A = \sqrt{\frac{\min\left(d^l, d^r\right)}{\max\left(d^l, d^r\right)} \frac{\min\left(d^t, d^b\right)}{\max\left(d^t, d^b\right)}}. \tag{5}$$

Finally, the definition of $EIoU$ is obtained

$$EIoU = W_A \cdot IoU. \tag{6}$$

Among them, IoU is the intersection over union between the prediction box A and the label box.

In Figure 6, we can see that after adding the centrality weight to the IoU, the prediction box closer to the inside of the target label box has a higher score. The result is more suitable for our detection purpose.

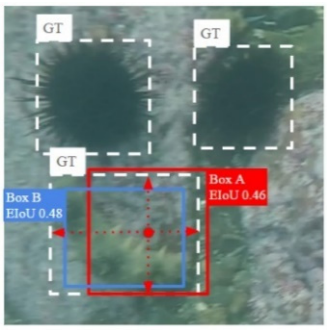

Figure 6. Example of EIoU.

2.3. K-Means++ Clustering

The K-means [16] algorithm is a commonly used clustering method to put similar items together. First, several center points are randomly selected. Then, the distance between the remaining points and these center points are calculated. It is classified as represented by the nearest center point to complete the clustering of all data.

The anchor box of Faster-RCNN is manually set with nine kinds of sizes, which are combined with three aspect ratios and three different areas. This artificially set anchor box has a certain universality in general data sets, but it cannot achieve the best detection effect on the URPC2018 underwater biology data set used in this article. Moreover, in the clustering algorithm, the choice of the initial clustering center is very important. It can often determine the quality of the clustering result of the algorithm. The selection of cluster centers in the K-means algorithm is random. Therefore, it is impossible to determine how to choose cluster centers to get better results.

Therefore, this article uses K-means++ [17] to perform clustering analysis on the URPC2018 data set to generate a set of anchor boxes. In the actual underwater biological data set, a suitable anchor box size can improve the detection accuracy of the model.

Figure 7 shows the size distribution of the label box of the underwater biological data set used in this article. It can be seen from the figure that most of the points are clustered in the lower left corner since the K-means++ clustering algorithm selects the points as far away as possible from each other as the clustering center to avoid the influence of some special points on the clustering effect. In this paper, about 5% of the data is discarded, and all label boxes with a length not greater than 225 and a width not greater than 125 are clustered using the K-means++ clustering algorithm.

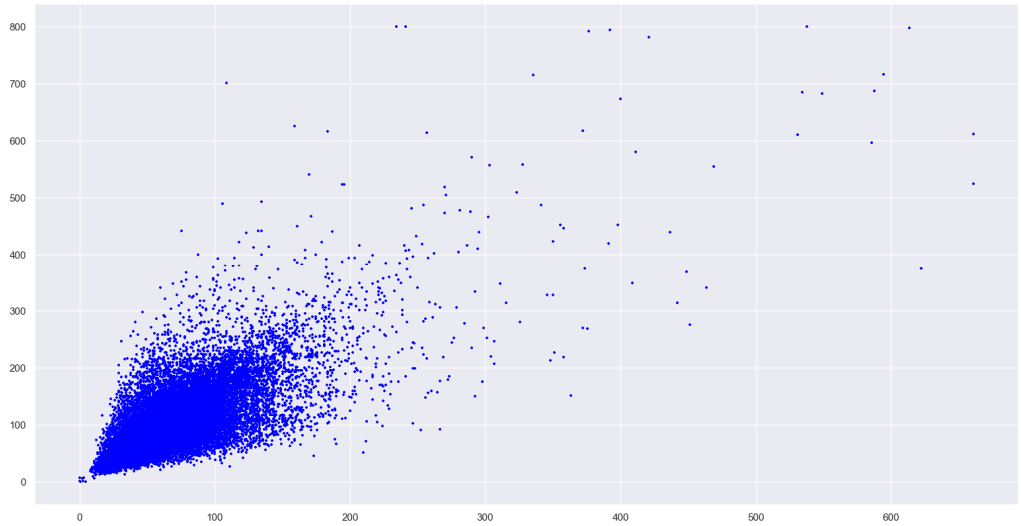

Figure 7. Size distribution of labels in URPC2018 dataset.

The clustering results are shown in Figure 8. This article will use the coordinate values of these nine center points as the length and width of the anchor box to train the Faster-RCNN network model.

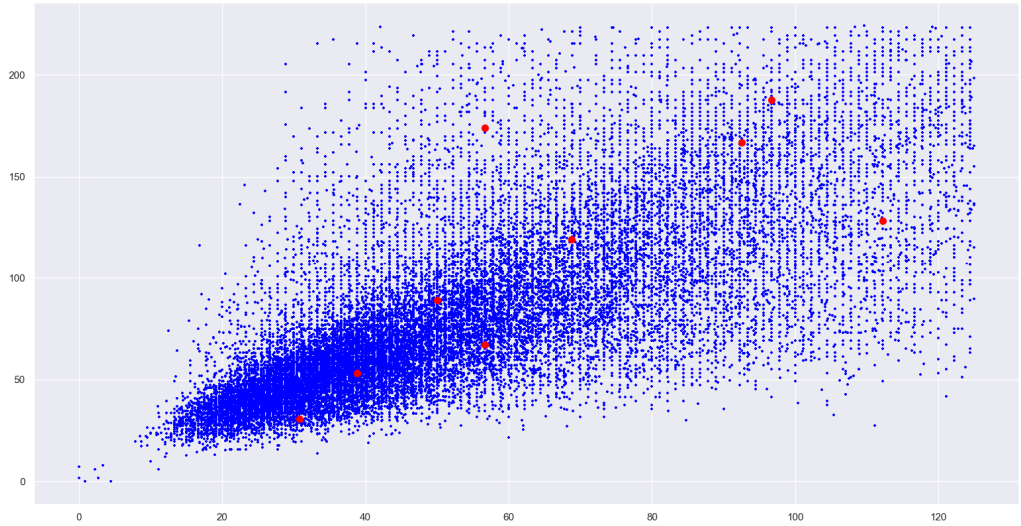

Figure 8. Using k-means ++ clustering algorithm to generate anchors.

2.4. Experimental Configuration and Dataset

Network training is carried out on a workstation equipped with Intel Xeon(R) CPU E5-2620 v4 @ 2.10 GHz processor (Intel, Silicon Valley, CA, USA), 32 GB memory, NVIDIA

GeForce RTX2080 graphics card (NVIDIA, Santa Clara, CA, USA). The Pytorch version is 1.7.0, and the programing language is python.

This article uses the official data set provided by the URPC2018 Underwater Target Detection Competition [18]. The data set provides underwater images and box level annotations. The images are taken by underwater robots, which can reflect the real underwater situation. The URPC2018 data set has a total of 5543 images in the training set, including four categories of sea cucumber, sea urchin, scallop, and starfish. The data set is divided into 8:1:1 in the order of training set, validation set, and test set. The input image size is set to 800 × 800.

In this paper, the weight file which has been pre-trained by Faster-RCNN on the VOC data set is selected as the pre-training weight. The learning rate setting adopts the cosine annealing decay adjustment strategy. The initial learning rate is set to 1×10^{-4}, while the minimum learning rate is 1×10^{-5}. The number of iterations of a learning rate cycle is 5. The entire model training process is divided into two steps. Firstly, the ResNet parameter training is frozen for 100 epochs to avoid damage to the initial weight of the training. The batch size is set to 16. Secondly, the ResNet parameter is unfrozen and then trained for 100 epochs. The batch size is set to 4. All the experimental network models reached convergence before 200 epochs.

2.5. Evaluation Index

In order to quantitatively analyze the target detection effect of the algorithm in this paper, this paper uses Mean Average Precision (mAP) as the evaluation index.

mAP depends on precision and recall. The precision rate represents the ratio of the number of correctly identified category C on a picture for a certain category C (True Positives) to the total number of category C identified on the picture:

$$p = \frac{TP}{TP + FP} \tag{7}$$

The recall r represents the number TP of correctly recognized category C on a picture and the total number of category C on the picture (including the correct recognition number TP and the recognition number of category C but divided into other categories FN (False Negatives)):

$$r = \frac{TP}{TP + FN} \tag{8}$$

The average accuracy AP (Average Precision) of a single category C is based on r as the x-axis and p as the y-axis, draw a P–R curve, and AP value is calculated by the area under the curve:

$$AP = \int_0^1 p(r)dr \tag{9}$$

For a test set with N categories, the mAP calculation formula is as follows:

$$mAP = \frac{\sum_{k=1}^{N} AP}{N} \tag{10}$$

3. Results and Discussion

This paper compares the three improvements with the Faster-RCNN network in detail under the condition that the control training parameters are consistent during the experiment. Under the premise that the default IoU is 0.5 and the confidence is 0.5, the mAP of the Faster-RCNN model test result is 80.68%. After using the ResNet–BiFPN feature extraction network, mAP increased by 4.61%, and the detection effect of the four types was improved to a certain extent. Among them, scallops have the largest increase in AP, which is 7.66%, and the least increase is starfish, which is 1.96%. After using $EIoU$, mAP increased by 0.36%, and the detection effect of all three categories except starfish was also improved. Among them, sea cucumbers had the highest AP increase, which was

0.92%. The detection effect of starfish is slightly reduced, and the AP reduction rate is 0.27%. After using k-means++, mAP increased by 1.11%. Among the detection effects of the four categories, the AP increased by 1.44% for sea cucumbers.

This experiment also verified that the superposition of any two improvements could continue to improve the detection effect on the basis of the original single improvement. The mixture of ResNet–BiFPN and $EIoU$ can increase by 5.66%. Among the four types of detection, scallops have increased the most, and the AP has increased by 8.30%; the mixture of $EIoU$ and k-means++ can increase mAP by 1.19%. Among the four categories of detection, sea cucumber has the most improvement, which is 2.15%; the mixture of ResNet–BiFPN and k-means++ improves mAP the most, which is 6.24%. Among the four types of detection, scallops have the most improvement, which is 8.74%.

Finally, compared with the traditional Faster-RCNN algorithm, the improved Faster-RCNN algorithm in this paper increases mAP by 8.26%. The increase in AP detection for the four categories is 4.97–10.49%, and the experimental results can fully prove the effectiveness of the improved structure in this paper.

Table 1 is the detailed comparison of experimental results, where ① represents the use of ResNet–BiFPN improved feature extraction network on Faster-RCNN alone. ② represents the use of EIoU alone on Faster-RCNN, and ③ represents the use of K-means++ on Faster-RCNN. ① + ②, ② + ③, and ① + ③ represent the combination of two improvements corresponding to numbers, and the proposed is the improved Faster-RCNN underwater organism detection algorithm proposed in this paper.

Table 1. Test results of URPC2018 dataset.

Algorithm	*mAP* (%)	Sea Urchin (%)	Starfish (%)	Scallop (%)	Sea Cucumber (%)
Baseline	80.68	87.39	86.62	75.55	73.17
①	85.29	90.03	88.58	83.21	79.36
②	81.04	87.48	86.35	76.23	74.09
③	81.79	88.80	87.39	76.34	74.61
① + ②	86.34	91.36	89.02	83.85	81.14
② + ③	81.87	88.14	87.20	76.81	75.32
① + ③	86.92	91.93	90.25	84.29	81.24
Proposed	88.94	92.36	90.78	86.04	82.93

Figure 9 shows the detection results of Faster-RCNN on underwater organisms before and after some improvements. It can be seen that the improved Faster-RCNN has excellent detection effects on blurred targets, dense targets, multi-scale targets, and occluded targets.

The underwater biological detection method based on the improved Faster-RCNN has a better detection effect on the incomplete underwater organisms, such as the sea cucumber in the lower right corner of column (a); the improved Faster-RCNN can find this sea cucumber with an incomplete display with a higher degree of confidence. The algorithm in this paper is also more able to "draw the boundaries" when detecting dense organisms, as shown in column (b). The Faster-RCNN before the improvement has a lot of mislabeling and missing labels: the sea cucumber in the lower left corner has not been detected. There are many sea cucumbers that were identified as scallops and sea urchins, and the selection of targets was not precise enough. When the creatures are occluded and overlap each other, the improved Faster-RCNN can accurately distinguish, locate, and identify each creature in the picture. Finally, it can give very high confidence to most of the detected object's degree. The improved Faster-RCNN's ability to detect multi-scale targets is also very powerful. As shown in column (c), after adding the BiFPN bidirectional feature pyramid structure, Faster-RCNN's ability to detect small-scale targets such as scallops has also been enhanced compared with before the improvement. The improved Faster-RCNN can not only find more scallops but also give a high degree of confidence. The improved Faster-RCNN's ability to detect fuzzy targets has also been greatly improved. As shown in column (d), due to serious image distortion and extremely low definition, the improved Faster-RCNN

missed two sea cucumbers. Faster-RCNN improved by using ResNet–BiFPN has a more powerful feature extraction capability and can achieve higher precision detection even in such a fuzzy situation. EIoU can make the prediction box more closely fit the target organism, such as the sea urchin prediction box in column (a). The sea urchin prediction box of the improved Faster-RCNN is obviously closer to the sea urchin.

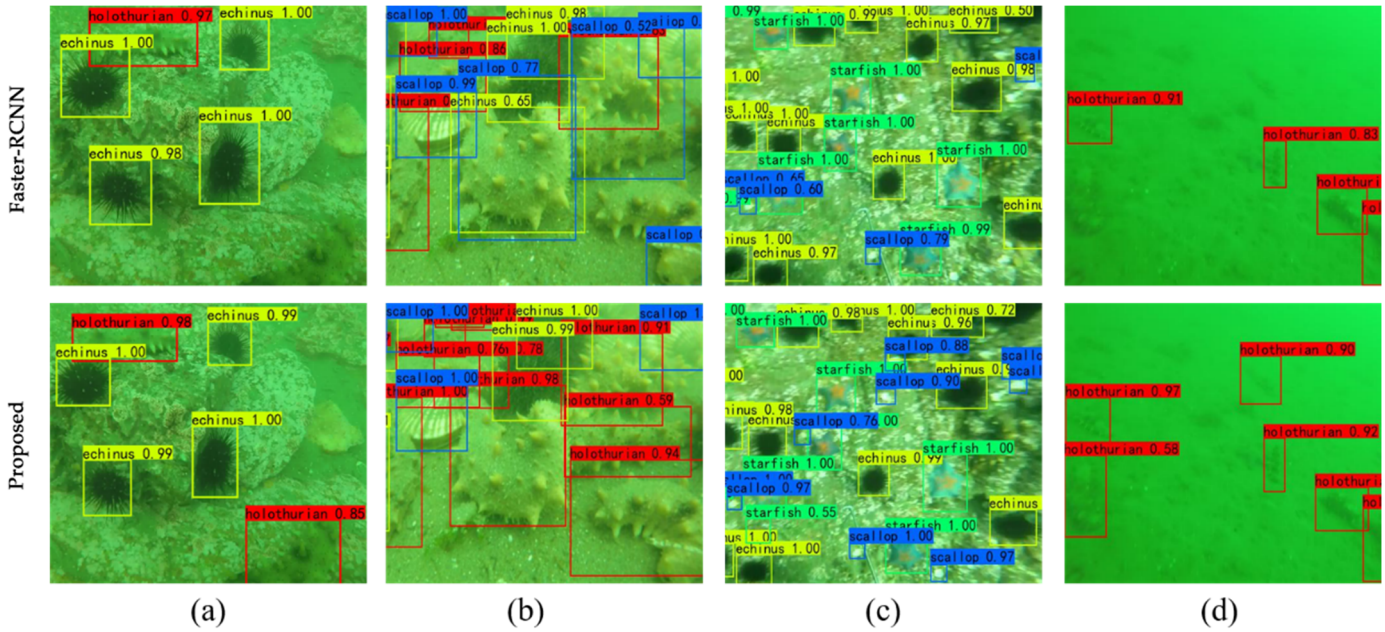

Figure 9. Comparison of test results: (**a**) detection of incomplete targets; (**b**) detection when targets are occluded or overlapped with each other; (**c**) detection of multi-scale targets; and (**d**) detection of fuzzy targets.

In general, ResNet–BiFPN is used to enhance the ability of model feature extraction, multi-scale feature fusion EIoU is used to improve the quality of the prediction frame, and K-means++ clustering is used to generate a more appropriate anchor frame. The proposed algorithm can achieve higher detection accuracy in underwater biological detection.

In addition, this article includes experimental statistics on the number of BiFPN stacks in ResNet-BiFPN. After performing different times of BiFPN stacking, the specific data obtained are shown in Table 2. With the increase of BiFPN stacking times, the feature fusion ability is strengthened. Thus, more effective features can be extracted, and the detection accuracy is improved. However, the network structure becomes more complex; the number of model parameters increases, which leads to reducing the detection efficiency. Considering the balance between detection accuracy and detection efficiency, the number of stacking times of BiFPN is selected as three.

Table 2. Test on different numbers of BiFPN.

	0 (Times)	1 (Times)	2 (Times)	3 (Times)	4 (Times)	5 (Times)
mAP (%)	83.47	84.95	85.18	85.29	85.32	85.30
FPS	5.31	4.63	4.50	4.39	4.22	4.10

In this paper, the performance of the Faster-RCNN network under various improvements is tested experimentally. Since the size of the anchor box does not affect the operating performance of the network, additional performance tests are not performed on the network model with improved K-means++. The parameter quantity of ResNet–BiFPN as the backbone feature extraction network is much larger than that of VGG as the backbone feature extraction network, so it has the greatest impact on the performance of the entire network. EIoU adds a centrality weight on the basis of IoU, which only adds a small

amount of calculation. Thus, it has little effect on the operating performance of the entire network. The specific test results are shown in Table 3.

Table 3. Impact of improvements on network performance.

Algorithm	*FPS*
Baseline	5.31
ResNet–BiFPN	4.39
EIoU	5.18
K-means++	/
ResNet–BiFPN + EIoU	4.30
EIoU + K-means++	/
ResNet–BiFPN + K-means++	/
Proposed	4.30

In order to further verify the performance of the proposed algorithm, we compare it with the YOLOv4 [19] object detection model. The results are shown in Table 4. The comparison includes *mAP* and the detection accuracy of each creature. It can be seen that the proposed algorithm has relatively high accuracy compared to YOLOv4 and Faster-RCNN. Compared with YOLOv4, *mAP* increased by 17.58%, and the detection AP for the four categories increased by 8.96–24.58%.

Table 4. Comparison of different detection models on the URPC2018 dataset.

Algorithm	*mAP* (%)	Sea Urchin (%)	Starfish (%)	Scallop (%)	Sea Cucumber (%)
YOLOv4	71.36	83.40	79.66	61.46	60.93
Faster-RCNN	80.68	87.39	86.62	75.55	73.17
Proposed	88.94	92.36	90.78	86.04	82.93

4. Conclusions

This paper proposes an improved Faster-RCNN underwater organism detection algorithm to solve the problems of low detection accuracy. As we can see from the improved algorithm and experimental results, the improved network structure ResNet–BiFPN has better capability in feature extraction and multi-scale feature fusion. Additionally, the EIoU can reduce the proportion of redundant bounding boxes in the training data. Moreover, the k-means++ clustering generates suitable target anchor frames to improve detection accuracy. Compared with the YOLOv4 and Faster-RCNN method, the accuracy of underwater biological detection has been significantly improved, which fully demonstrates the effectiveness of this algorithm. However, the detection speed still needs to be improved to meet the needs of real-time detection. Therefore, the lightweight design of the proposed algorithm and its embedded transplantation should be the focus of attention and research in the future.

Author Contributions: Conceptualization, P.S.; methodology, J.N.; validation, J.N.; formal analysis, X.X.; investigation, W.H.; resources, S.H.; data curation, S.H.; writing—original draft preparation, X.X.; writing—review and editing, Y.X.; visualization, P.S.; supervision, P.S.; project administration, J.N.; funding acquisition, P.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant No. 61801169, No. 61873086, in part by the Fundamental Research Funds for the Central Universities (B210202087), and in part by the free exploration research fund of Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University (2021JSSPD03).

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

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

References

1. Wang, T.; Chen, Y.; Qiao, M.; Snoussi, H. A fast and robust convolutional neural network-based defect detection model in product quality control. *Int. J. Adv. Manuf. Technol.* **2018**, *94*, 3465–3471. [CrossRef]
2. Wang, R.; Zhang, Y.; Tian, W.; Cai, J.; Hu, C.; Zhang, T. Fast Implementation of Insect Multi-Target Detection Based on Multimodal Optimization. *Remote Sens.* **2021**, *13*, 594. [CrossRef]
3. Xu, X.; Li, X.; Zhao, H.; Liu, M.; Xu, A.; Ma, Y. A real-time, continuous pedestrian tracking and positioning method with multiple coordinated overhead-view cameras. *Measurement* **2021**, *178*, 109386. [CrossRef]
4. Brys, T.; Harutyunyan, A.; Vrancx, P.; Nowé, A.; Taylor, M.E. Multi-objectivization and ensembles of shapings in reinforcement learning. *Neurocomputing* **2017**, *263*, 48–59. [CrossRef]
5. Gao, S.H.; Tan, Y.Q.; Cheng, M.M.; Lu, C.; Chen, Y.; Yan, S. Highly efficient salient object detection with 100k parameters. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 702–721.
6. Fan, D.P.; Zhai, Y.; Borji, A.; Yang, J.; Shao, L. BBS-Net: RGB-D salient object detection with a bifurcated backbone strategy network. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 275–292.
7. Shi, P.; Fang, X.; Ni, J.; Zhu, J. An Improved Attention-Based Integrated Deep Neural Network for PM2. 5 Concentration Prediction. *Appl. Sci.* **2021**, *11*, 4001. [CrossRef]
8. Ni, J.; Chen, Y.; Chen, Y.; Zhu, J.; Ali, D.; Cao, W. A survey on theories and applications for self-driving cars based on deep learning methods. *Appl. Sci.* **2020**, *10*, 2749. [CrossRef]
9. Ni, J.; Gong, T.; Gu, Y.; Zhu, J.; Fan, X. An improved deep residual network-based semantic simultaneous localization and mapping method for monocular vision robot. *Comput. Intell. Neurosci.* **2020**, *2020*, 7490840. [CrossRef] [PubMed]
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Las Condes, Chile, 11–18 December 2015; pp. 1440–1448.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]
13. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
14. Tan, M.; Pang, R.; Le, Q. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
15. Ma, J.; Chen, B.; Sun, X. General object detection framework based on improved Faster R-CNN. *J. Comput. Appl.* **2021**, 1–9. Available online: https://kns.cnki.net/KCMS/DETAIL/51.1307.TP.20210205.1531.023.HTML (accessed on 1 September 2021).
16. Arunkumar, N.; Mohammed, M.A.; Ghani, M.K.A.; Ibrahim, D.A.; Abdulhay, E.; Ramirez-Gonzalez, G.; Albuquerque, V.H.C. K-means clustering and neural network for object detecting and identifying abnormality of brain tumor. *Soft Comput.* **2019**, *23*, 9083–9096. [CrossRef]
17. Chakraborty, N.; Ray, A.; Mollah, A.F.; Basu, S.; Sarkar, R. A Framework for Multi-lingual Scene Text Detection Using K-means++ and Memetic Algorithms. In *Machine Learning for Intelligent Multimedia Analytics: Techniques and Applications*; Springer: Singapore, 2021; pp. 167–187.
18. Chen, L.; Liu, Z.; Tong, L.; Jiang, Z.; Wang, S.; Dong, J.; Zhou, H. Underwater object detection using Invert Multi-Class Adaboost with deep learning. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
19. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal Speed and Accuracy of Object Detection. 2020. Available online: https://arxiv.org/abs/2004.10934 (accessed on 23 April 2020).