Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4197178, 12 pages
https://doi.org/10.1155/2022/4197178

Research Article

Research on Underwater Target Recognition Technology Based on Neural Network

Zhiguang Guan, Chenglong Hou, Siqi Zhou, and Ziyi Guo

Shandong Provincial Engineering Lab of Traffic Construction Equipment and Intelligent Control, Shandong Jiaotong University, Jinan, 250357 Shandong, China

Correspondence should be addressed to Zhiguang Guan; guanzhiguang@sdjtu.edu.cn

Received 9 December 2021; Revised 25 January 2022; Accepted 23 March 2022; Published 14 April 2022

Academic Editor: Shunmei Meng

At present, the underwater environment required by the seafood aquaculture industry is very bad, and the fishing operation is completed artificially. In this environment, the use of machine fishing instead of artificial fishing is the development trend in the future. By comparing the characteristics of different algorithms, the multiscale Retinex algorithm (autoMSRCR) is selected to deal with image color skew, blur, atomization, and other problems. Labelimg software is used to annotate underwater targets in the image and make data sets. Of these, 20% are used as test sets, 70% as training sets, and 10% as verification sets. The target detection network of You Only Look Once Version4 (YOLOv4) based on convolutional neural networks (CNN) is adopted in this paper. The main feature extraction network adopts CSPDarknet53 structure, and the feature fusion network adopts SSP, and PANet network carries out sampling and convolution operations. The prediction output of extracted features is carried out through YoloHead network. After training the recognition model of the training sets, the detection effect is obtained by testing the data of the test sets. The identification accuracy of sea cucumber and sea urchin is 90.8% and 87.76%, respectively. Experiments show that the target detection network model can accurately identify the specified underwater organisms in the underwater environment.

1. Introduction

In China, offshore seafood aquaculture, sea cucumbers, and sea urchins grow at the bottom of the seawater. In particular, sea cucumbers and sea urchins live on reefs of 12-13 meters underwater or artificial reefs. When the temperature is lower than 0°C or higher than 20°C, sea cucumbers will enter the seabed or dormancy. The depth and the presence of rocks make fishing operations extremely difficult. At present, divers can only go into the sea to fish seafood. "Humans cannot work underwater for a long time because of their body structure and the way they breathe. Fishing divers are prone to decompression sickness and rheumatoid arthritis" [1]. Hence, the high fees paid to divers result in huge fishing costs. At present, there is no suitable robot to replace artificial underwater operations in seafood aquaculture. The fishing robot can replace the artificial long-term dangerous operation and reduce the risk and cost of fishing [2]. "With the rapid development of digital image processing and computer technology, neural network technology is becoming more and more mature in the field of computer. The neural network model has the advantages of self-learning ability, strong adaptability, and high robustness and is especially suitable for classification and recognition problems" [3–5].

Currently, popular target detection algorithms based on deep learning can be divided into two categories: one-stage network and two-stage network [6]. One-stage network is much faster in detection speed than two-stage network, but lower in detection accuracy.

Two-stage network firstly carries out region proposal (RP) for the input images and then classifies them through CNN. Representative algorithms include R-CNN, SPP-Net,

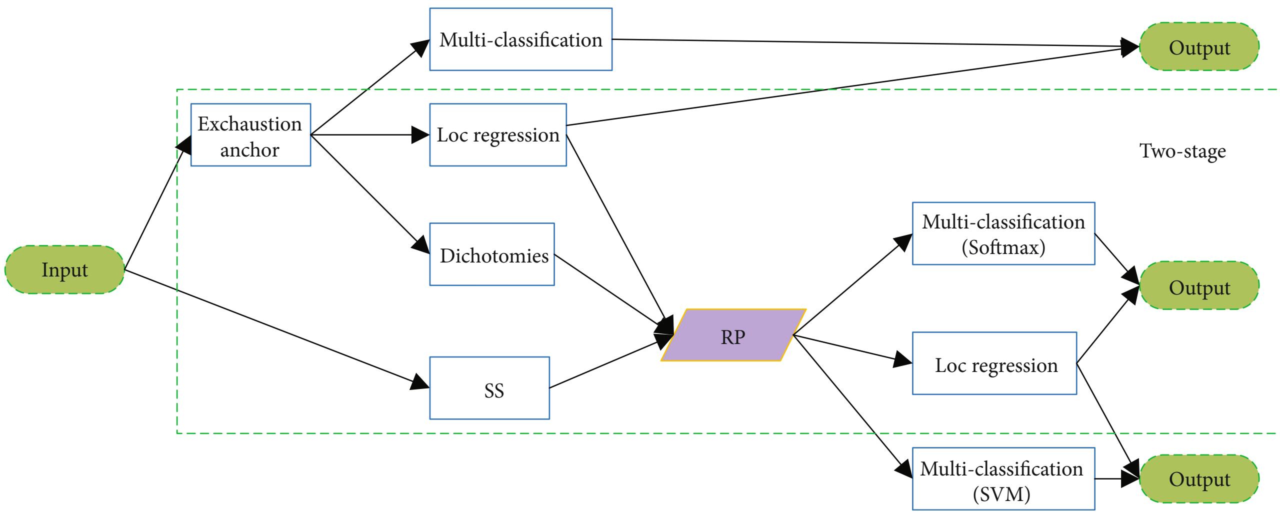


Figure 1: Two-stage network structure.

Fast R-CNN, Faster R-CNN, and R-FCN [7, 8]. The network structure is shown in Figure 1.

One-stage network input does not use the RP generation prior-box but directly extracts features from CNN to predict object classification and location. Representative algorithms include OverFeat, YOLOv1, YOLOv2, YOLOv3, YOLOv4, SSD, and RetinaNet [9]. The network structure is shown in Figure 2.

The target detection network based on deep learning is applied to the underwater robot. The main problem is the accurate recognition of seafood in a shallow sea environment. The specific steps are as follows:

(1) Manufacturing of data sets: according to the changes of underwater environment, the image is preprocessed to enhance the feature information and distinguish the training sets, test sets, and verification sets. Labelimg is used to mark the data for network training

(2) Targeting recognition: building the framework of YOLOv4, inputting the processed training sets, and getting the trained model

(3) Adjusting model parameters: using the test sets to test, then adjusting the learning rate and the network model of data processing way, and letting the model accuracy and speed realize optimality

(4) Realizing recognition: building a platform on the existing underwater fishing robot to realize the combination of algorithm and model and verifying the performance and reliability of the algorithm

2. Data Collection and Production

The data sets and network structure are the main factors influencing on the detection accuracy in the target detection algorithm based on neural network. Having large data sets is the premise of training and optimizing high performance network model.

2.1. Data Collection and Processing. Since it is underwater real-time detection, the authenticity of the image will directly affect the robustness of the training model. Underwater image sets come from visual competitions, most of which are underwater aquaculture environments of sea cucumbers and sea urchins. The original data sets of the competition have four species. Two species of sea cucumbers and sea urchins are adopted and carried out manual labeling.

The complex physical environment changes of ocean make the underwater images by ocean optical and visual imaging system degraded greatly. There are some serious problems such as image color fatigue, low contrast, and blurred details. The severely degraded underwater images lack effective data and information for target recognition, so the recognition difficulty increases [10]. Therefore, it is necessary to preprocess the image using image enhancement technology and carry out feature extraction in CNN.

Retinex is based on the theory that the color of an object is determined by its reflection of light, not by the absolute value of reflected light intensity. The color of an object is not affected by the illumination uniformity and has consistency.

Multiscale Retinex algorithm formula is as follows:

$$\log R_i(x, y) = \log \sum_{k=1}^{k} W_k\{\log I_i(x, y) - \log [F_k(x, y) * I_i(x, y)]\}, \tag{1}$$

where K stands for the number of scales, which is usually 3. When $K = 1$, it is the single-scale Retinex algorithm. W_k stands for weighting coefficient. F_k stands for filter function. I_i stands for original input images. The i stands for RGB color channel.

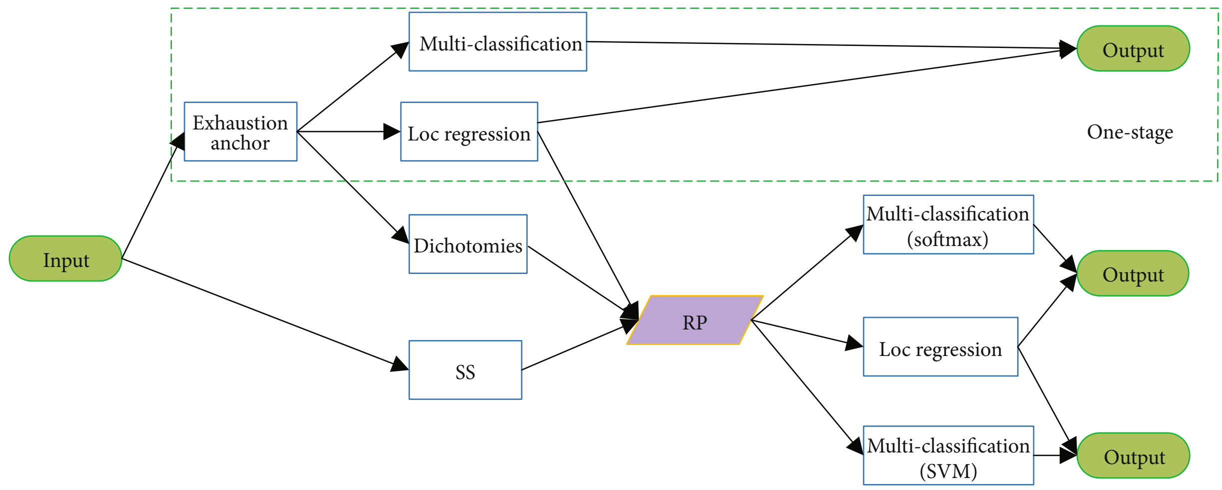


Figure 2: One-stage network structure.

Multiscale Rentinex algorithm with color balance is an improved algorithm of MSR. In MSR image, due to the increase of noise, local detail color distortion will be caused, and the overall visual effect will be worse. To solve this problem, color restoration factor is added to adjust the weight between the three-color channels in the original image. Thus, the information in the relatively dark area can be color adjusted to eliminate the defect of image color distortion [11]. The formula of MSRCR with color balance is as follows [12]:

$$R_{MSRCRi}(x, y) = C_i(x, y) R_{MSRi}(x, y), \tag{2}$$

where $C_i(x, y)$ stands for color recovery factor of channel i.

The MSRCR algorithm with automatic color levels removes the largest and smallest part of the MSRCR processing results according to a certain percentage. Then, the remaining middle part is quantified to 0-255, which can restore the image better than MSRCR [13]. The image preprocessing effect comparison is shown in Figure 3.

The image processed by autoMSRCR has the most obvious contrast, the better defogging effect, the most obvious local details, and the better effect. Therefore, autoMSRCR is selected as the image preprocessing algorithm.

2.2. Data Set Making. The research object are underwater image data, which are difficult to collect. The data sets in the online vision competition are adopted in the paper, which contains four species, namely, sea cucumber, sea urchin, coral reef, and seaweed. Only sea cucumbers and sea urchins are selected in the experiment. 1200 images are manually selected as the original data sets, but such small data sets will cause problems such as low precision and overfitting of the model in the training process. Therefore, data augmentation is used to expand the number of images in the data sets.

Mosaic data augmentation method is used in this paper. Mosaic date augmentation method takes four images and splices them together to form a new image. The process is to read four pictures randomly and then reverse the four pictures, zoom, gamut, and other changes. And according to the position of the top left, top right, bottom left, and bottom right, an image is spliced. And then, combine it into an image, which is shown in Figure 4.

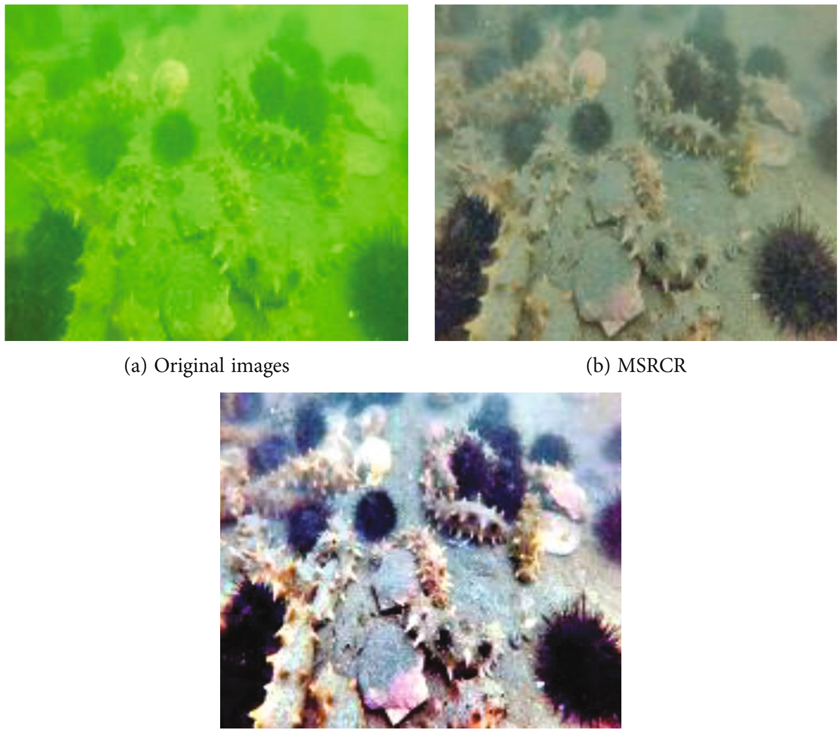
Object detection based on deep learning is a kind of supervised learning [14]. The feature of the target is extracted directly through the convolutional network for learning. Therefore, the position of the target in the image needs to be manually labeled, and the labeled information is converted into VOC2007 format. Labelimg is used in this experiment to select the target. The labeling annotation process is shown in Figure 5.

The annotated data sets are divided into training sets, verification sets, and test sets in proportion. To ensure a wide range of data coverage, the division principle is random. The ratio adopted in this experiment is as follows: training sets : verification sets : test sets is 7 : 1 : 2, that is 840 images of training sets (excluding augmented images), 120 images of verification sets, and 240 images of test sets.

3. Target Detection Algorithm Based on YOLOv4 Network

YOLOv4 network mainly consists of three parts: backbone network, neck network, and head network [15]. CSPDarknet53 is used as the backbone feature extraction network. Mish function is used as the activation function. SSP and PANet are used as the neck network, which can effectively separate the most significant features of context. In the head part, the YOLO Head is adopted as the feature utilization part to extract and convolved. The anchor frame system of RCNN is introduced to greatly improve the map. There is no regional sampling, so it performs well on the global information.

3.1. YOLOv4 Network Structure. The backbone network of YOLOv4 adopts CSPDarknet53 network structure with large residual edges. The image size used in this experiment is 416×416. It is input into the CSPDarknet53 network, and



(a) Original images (b) MSRCR

(c) autoMSRCR

Figure 3: Image preprocessing effect comparison.

channels are added by using a single convolutional layer of Mish function. Among them, the Mish function is the activation function, which has the advantages of smooth gradient descent good effect. Meanwhile, the unboundedness of Mish function can avoid the saturation problem, which is used in this experiment. Then, feature extraction is performed through an 11-layer Resblock network with residual structure to generate 52×52 output I. At the same time, the output continues to extract features from the 8-layer Resblock network to produce output II with a size of 26×26. Output II also extracts features from the 4-layer Resblock network to produce output III with a size of 13×13.

In the neck network, the output III generated by the backbone network enters the SPP network structure. Output III is pooled at four different scales, and the pooled nucleus sizes are 13×13, 9×9, 5×5, and 1×1, respectively. Output II and output I are transmitted into PANet network, and the output through SPP structure is also transmitted into PANet network through a connection layer. Features are repeatedly extracted through up- and downsampling pyramid to achieve the best separation effect.

In the head network, YOLOv4 extracts three feature layers transmitted by the neck network and predicts the output through two-layer convolution. The overall network structure of YOLOv4 is shown in Figure 6.

3.2. Target Loss Function. The loss function of YOLOv4 can be divided into three parts: classification loss, confidence loss, and location loss [16]. The CIoU loss function is used in location loss to reflect the deviation between the real frame and the prediction frame, which is added the coverage area, center distance, and aspect ratio based on IoU loss function. The loss function is only calculated for positive samples. The formula are as follows [17]:

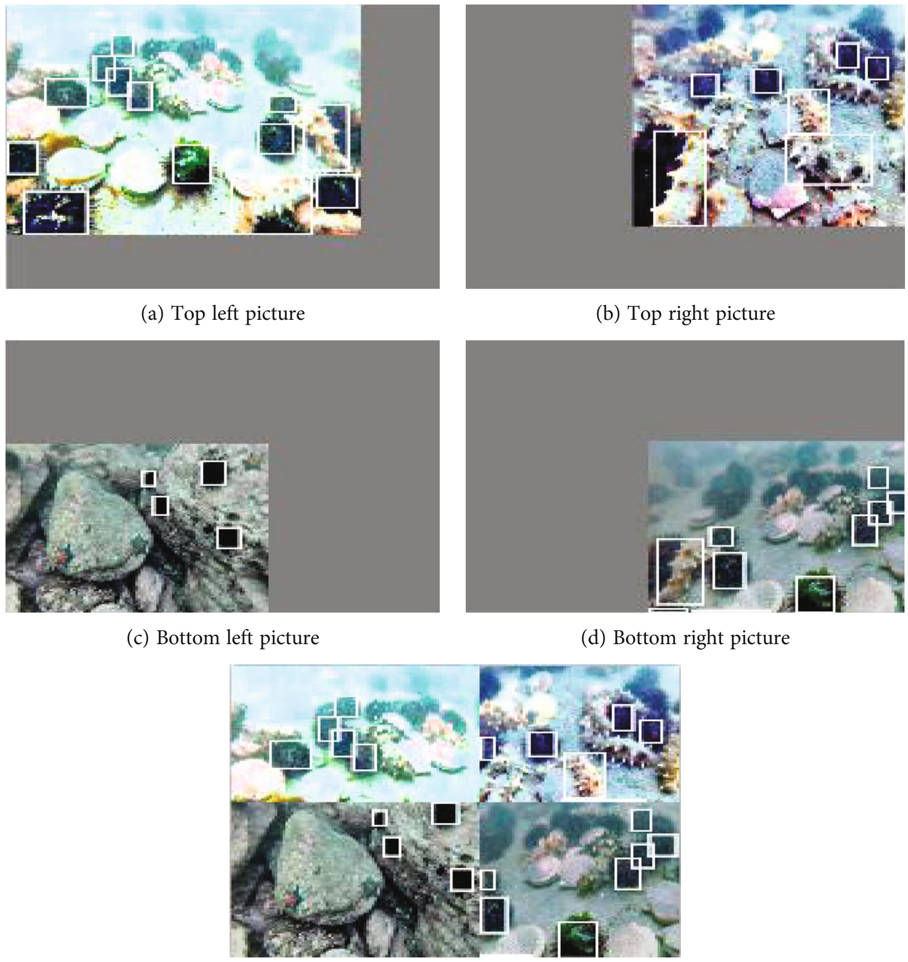
$$l_{CIoU} = 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \partial v,$$
$$v = \frac{4}{\pi^2}\left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h}\right)^2, \tag{3}$$

where $\rho^2(b, b^{gt})$ stands for Euclidean distance between the real frame and the prediction frame, c stands for the diagonal length of the minimum enclosing rectangle between the real frame and the prediction frame, v stands for distance between the width ratio of the real frame and the prediction frame, if the width and height are similar, then v =0. ∂ stands for item weight. w and w^{gt} and h and h^{gt} stands for the width and height of prediction frame and real frame, respectively.

The classification loss adopts binary cross entropy loss, which is calculated only when the sample is positive.

Confidence loss is divided into two parts, target-oriented loss and target-free loss, which are calculated both in positive and negative samples. It is better as positive sample confidence is closer to 1, or negative sample confidence is closer to 0.

In the training process, through the random gradient descent method and back propagation, the loss value of the loss function is continuously reduced in the iterative training, the learning rate is constantly updated according to the loss value, and the model parameters are constantly



(a) Top left picture

(b) Top right picture

(c) Bottom left picture

(d) Bottom right picture

(e) Splice picture

Figure 4: Image enlargement effect.

Figure 5: Data sets annotation process.

adjusted. The learning rate is also constantly updated according to the loss value, which minimizes the deviation between the prediction frame and the real frame. Continuously improve the network category confidence, so as to achieve optimal network performance.

4. Experimental Training and Result Analysis

By identifying sea cucumber and sea urchin, YOLOv4 network parameters are set according to the above methods, and the model is obtained by training on GPU.

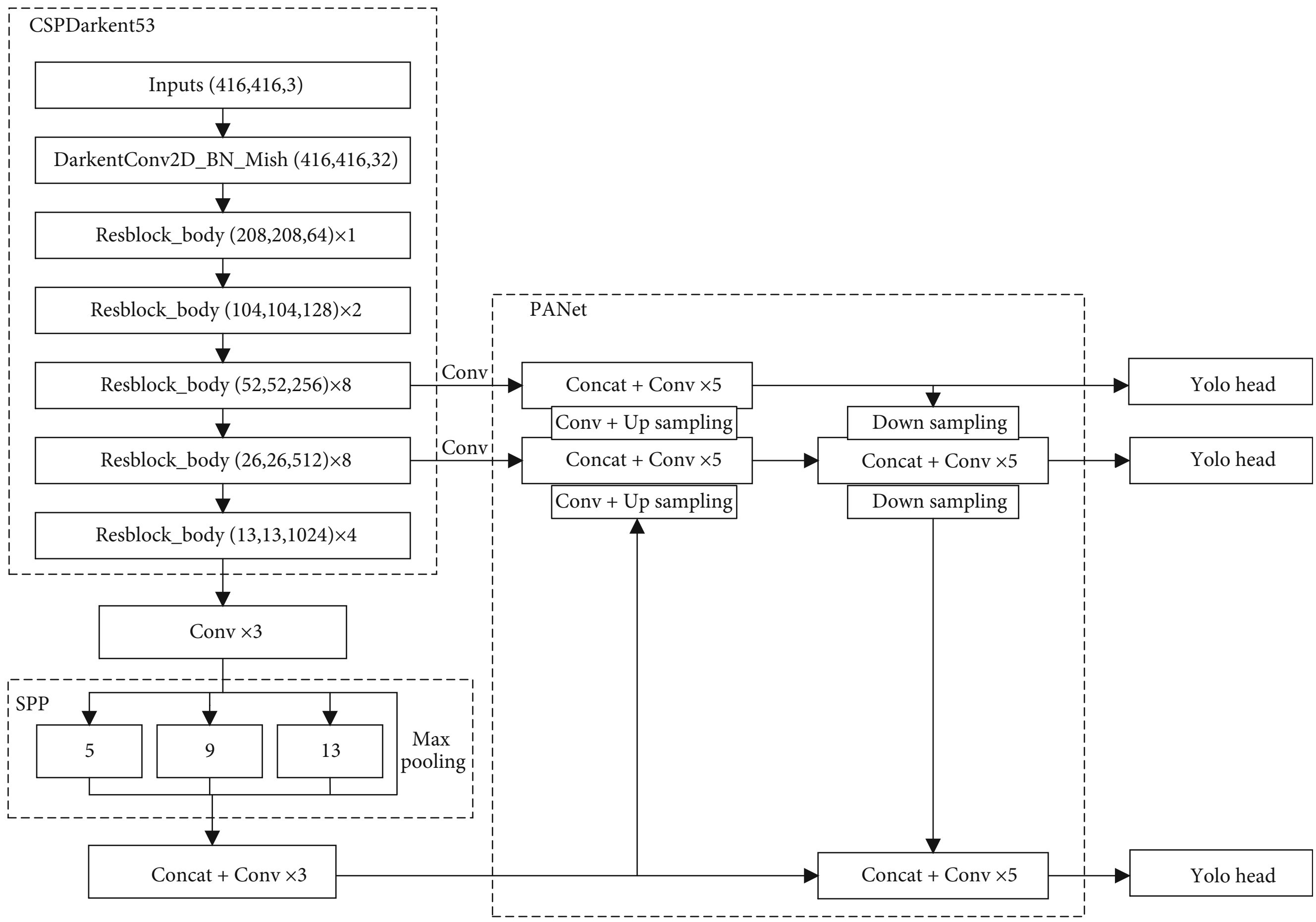


Figure 6: YOLOv4 network structure.

4.1. Experimental Platform and Parameter Design. Pytorch1.7 deep learning tool based on python3.8 is adopted, which supports a variety of classical neural network models. The system environment is Linux Ubuntu18.04. NIVIDIA CUDA11.0 version is adopted in GPU computing framework, and the corresponding neural network acceleration library cudnn is configured. The overall configuration is shown in Table 1.

Table 1: Experimental environment configuration.

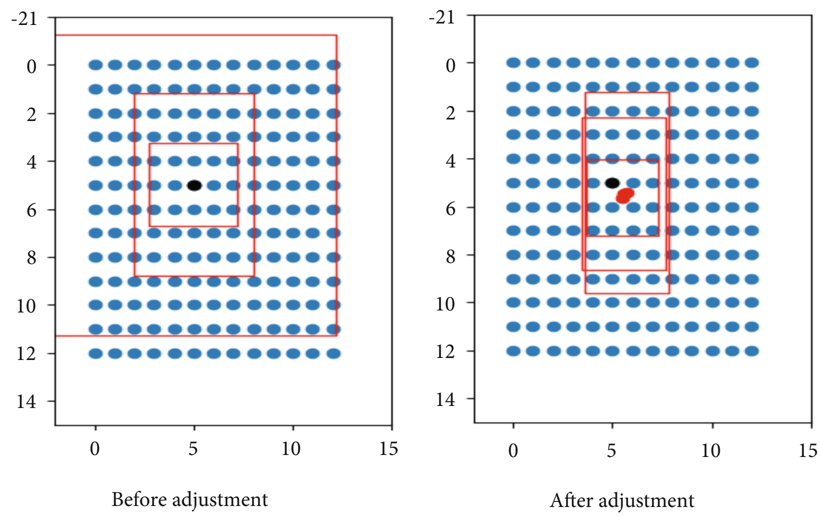
Project	Parameter
Operating system	Linux Ubuntu18.04
CPU	Intel(R) Xeon(R) Gold 6130 CPU
GPU	Tesla V100-32GB
Video driver	CUDA 11.0
Software environment	OpenCV 3.4.1.15 Python3.8
Deep learning framework	Pytorch 1.7

The image autoMSRCR algorithm processing and TensorboardX and Tqdm library network model training process are realized using OpenCV-python library.

Many parameters are involved in the initialization process of network training. The selection of training parameters and training strategies has influence on the convergence result and detection performance of the network. The main parameters are as follows: training batch (Batch_size), total iteration times (Epoch), frozen iteration times (Freeze_epoch), thawed iteration times (Thaw_ epoch), optimizer, initial learning rate (Base_LR), learning rate change strategy (Cosine_lr), and weight attenuation (Weight_decay).

The training batch is the number of samples selected in every training. The Batch_size directly affects the optimization degree and learning speed of the model. By setting Batch_size, GPU utilization is improved, and training time is reduced. The larger the Batch_size is, the more accurate the gradient calculation will be. Meanwhile, the number of iterations should be increased. The smaller Batch_size is, the less accurate the gradient calculation will be and the more obvious the oscillation will be.

In YOLOv4 network, the loss value of the model is calculated in the forward propagation process, and the gradient is calculated in the back propagation process. The selection of optimization algorithm will directly affect the training speed and accuracy of the model. Adaptive moment estimation is adopted to calculate and update the adaptive learning rate of each parameter. The learning rate determines the learning degree of each iteration and the updating speed of weights in



(a) Before adjustment (b) After adjustment

Figure 7: Comparison before and after adjustment of the prior-box.

Table 2: Parameters setting table.

Parameter	Numerical value	Parameter	Numerical value
Batch_size	32	Optimizer	Adam
Epoch	500	Cosine_lr	TRUE
Freeze_epoch	100	lr	0.01
Thaw_epoch	400	Weight_decay	0

the whole training process [18, 19]. When the learning rate setting is too large, it is easy to cause overfitting, while if the learning rate is too small, it is easy to produce slow convergence rate and poor recognition effect after model training.

4.2. Training Methods. The training of neural network is to transmit the image data to the network for calculation and reverse update the weight parameters of each layer of network. The network can accurately extract and detect the target features, calibrate the position of the target object, and output the processed image.

In YOLO Head, there are 13×13, 26×26, and 52×52 different size outputs, and the image is converted into a corresponding number of grids. 3 prior-boxes are generated at each grid point. By sigmoid function, the prediction results are normalized so that the center point of the prior-box is in the grid and the size of the prior-box is adjusted. The comparison before and after adjustment of the prior-box is shown in Figure 7.

In order to quickly get the accurate position of the prior-box, the k-means clustering algorithm is used to precalculate the prior-box. The k-means clustering algorithm is a clustering algorithm based on statistics, which can quickly obtain the size of clustering center and prior-box without machine learning [20]. 9 clustering centers are divided and the clustering standard is IoU [21]. The central coordinates after clustering are as follows:

[20.8 19.41333333]
[31.2 31.89333333]
[34.08888889 59.57530864]
[39.86666667 133.5308642]
[46.8 45.19506173]
[50.26666667 71.90123457]
[62.97777778 97.58024691]
[88.97777778 140.72098765]
[92.44444444 68.81975309]

The method of freezing training is first adopted and then thawing training in model training. The principle is to freeze the weight parameters of common parts (such as backbone network) and train the remaining parameters through the weight files obtained in advance. More resources are allocated to the neck and head network for training, and then after a certain number of iterations, the training time and resource utilization are improved.

In the process of model optimization training, there may be several local optimal solutions besides the global optimal solution. In the training process of gradient descent algorithm, the model may fall into the local minimum and cannot be optimized again. The study rate improvement strategy is cosine annealing algorithm (hot restart algorithm) to further improve the study rate. The principle is that hot restart is turned on after a few iterations, and local minimum value is skipped by increasing the learning rate of the model and learning continues. When the model approaches the global minimum, the control learning rate becomes smaller to avoid overfitting. When the loss value tends to be stable, the position deviation between the prediction frame and the real frame reaches the minimum. Category confidence is the highest, and network performance is the best.

Network training parameters are shown in Table 2:

4.3. Training Results. According to the above parameter setting, VOC2007 data set pretraining model is used to train the labeled data sets. There are 1000 times of

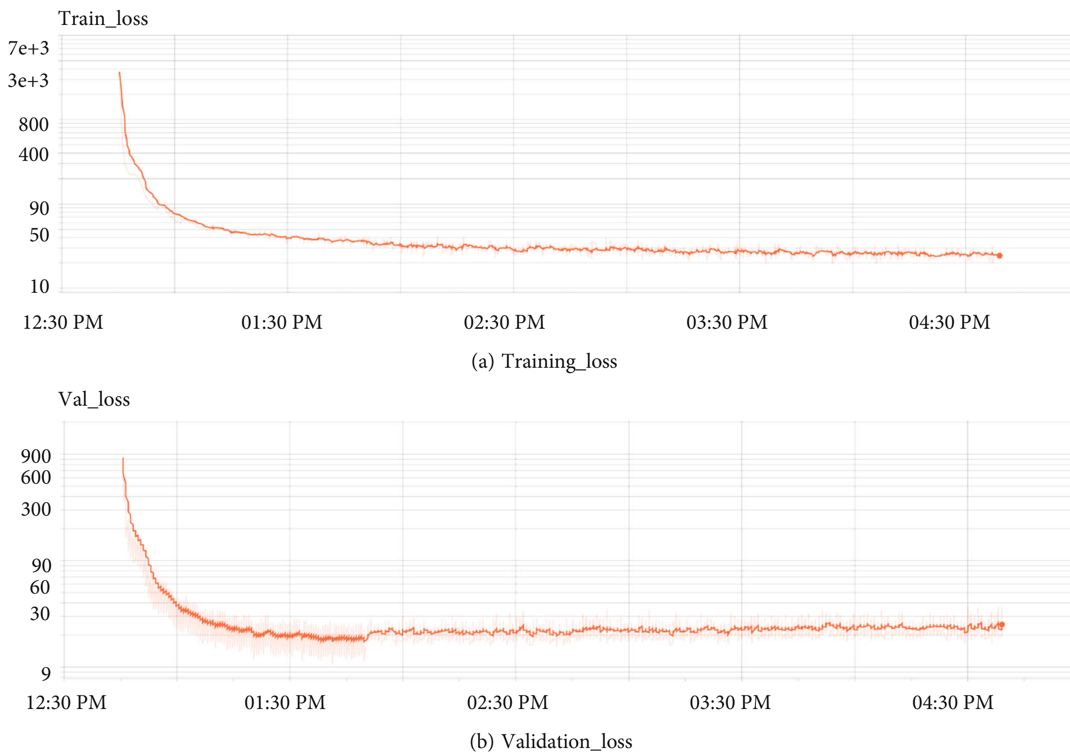


Figure 8: Loss variation diagram of training sets and validation sets.

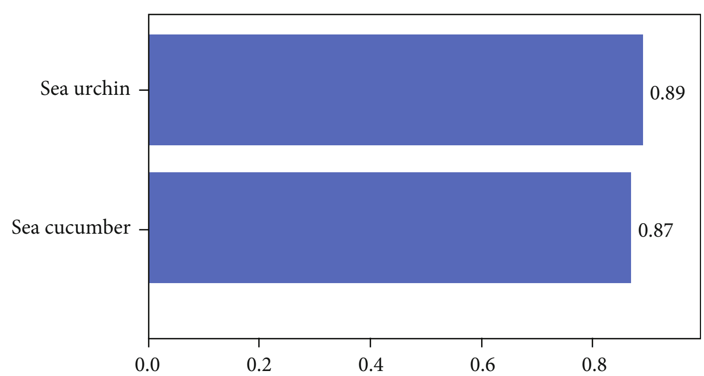


Figure 9: Average precision.

training where 100 iterations are freezing training and 900 iterations are thawing training. TensorboardX is used to record the Train_Loss value and Val_Loss value in the training process, which is shown in Figure 8.

It can be seen from the above figures that the loss value of the model is in a state of oscillation convergence during the training process. The loss value decreases with the increase of training times. The average accuracy of sea cucumber and sea urchin is shown in Figure 9.

The test is carried out on the test sets. After comparing the effects of each model on the test sets, it is found that the thawing training has the highest accuracy when the number of iterations is 805. Therefore, this model is selected as the final underwater sea cucumber and sea urchin recognition model, and the recognition effect is shown in Figure 10, where "green" color represents sea cucumber and "red" color represents sea urchin.

The sea cucumber and sea urchin recognition model is used to test the video at the rate of 11 frames per second on Windows system and 30 frames per second on Linux server. The video viewing rate is 24 frames per second, so running the model on the server can meet the real-time requirement.

The selected sea cucumber and sea urchin recognition model is tested on the test sets. Its accuracy, recall rate, comprehensive index ($F1$), and average accuracy are calculated to evaluate the model. The results are shown in Table 3.

$F1$ is a comprehensive index of precision and recall rate, which can be considered as the average effect. In general, the precision rate and recall rate affect and restrict each other. The calculation formula of precision and recall rate are as

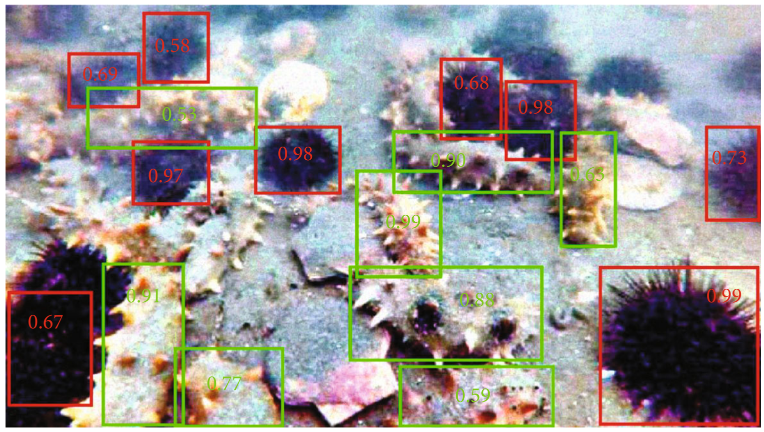


FIGURE 10: Recognition effect on the test sets.

TABLE 3: Parameters calculating.

Species	Identification accuracy (%)	Recall rate (%)	$F1$ (%)	Mean accuracy (%)
Sea cucumber	90.8	58.96	71	86.65
Sea urchin	87.76	76.14	82	89.31

follows [17]:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

where TP means that the actual situation is the positive example, and the predicted result is the number of positive examples. FP means that the actual situation is the number of negative examples, and the predicted result is the number of positive examples. FN means that the actual results are positive examples, and the predicted results are the number of negative examples.

The formula of $F1$ is as follows:

$$F1 = \frac{2}{(1/\text{PRE}) + (1/\text{REC})}, \tag{5}$$

where PRE stands for precision rate and REC stands for recall rate.

Various data show that the sea cucumber and sea urchin recognition model has a good effect on the test sets and can detect the target regardless of whether the target contour is clear or not. However, in general, the robustness of the model needs to be improved. When the image is blurred, the target object cannot be detected and the training times are less.

4.4. Error Analysis. After YOLOv4 model training and detection, there are two types of errors. One is the error in model training, and another is the error of model detection.

The main error of the model in the course of training is overfitting. The model overfitting the characteristics of the training data performed well in the training sets and predicted and distinguished all the targets almost perfectly. But in the validation sets, the performance is average with poor generalization and low robustness. There is no way to accurate judgement if it is a target with a new sample. The main reason for this problem in model training is that the amount of data is too small. In the training, Train_Loss decreases continuously while Val_Loss increases gradually, as shown in Figure 11.

YOLOv4 can detect sea cucumber and sea urchin targets at different scales and different scenarios, but some detection problems may occur in some environments. The experimental results show that there are two kinds of detection problems in the model test sets: missed and false detection.

Missed detection means that the model misses one or several objects in the image during detection, resulting in incomplete detection. False detection refers to the identification of an object in an image that is not a sea cucumber or sea urchin as a sea cucumber or sea urchin [22]. The reasons for this problem on YOLOv4 are two aspects. One is the image preprocessing using automatic color recovery Retinex algorithm. The characteristics of recognized objects are blurred due to distortion and contrast imbalance after image processing. Furthermore, it is lost in the process of convolutional neural network and feature transfer, which leads to missed or false detection. Another is the inaccuracy of artificial data sets. In the manual annotation data sets, some fuzzy objects observed by human eyes are not marked. Therefore, the model does not learn the fuzzy object during learning, also resulting in missed or false detection.

In view of the problems of the above model, a solution is proposed: firstly, manually relabel the data sets, especially the target in the fuzzy region, so that the model can be learned in the training process. Secondly, the value of Batch_size and iteration epoch should be adjusted appropriately during training to make model learning more

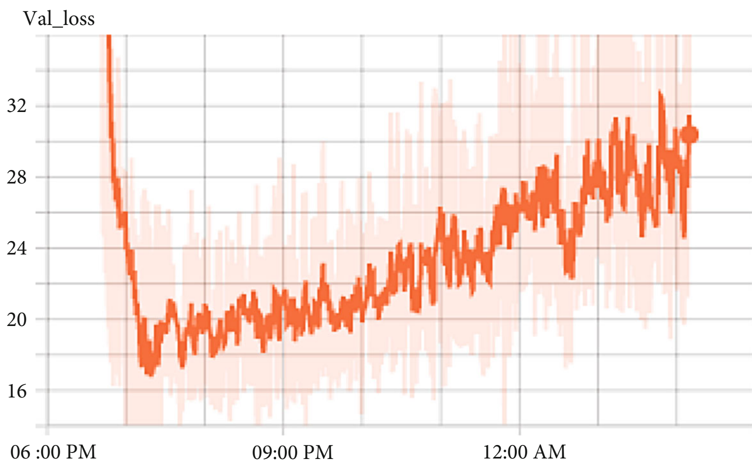
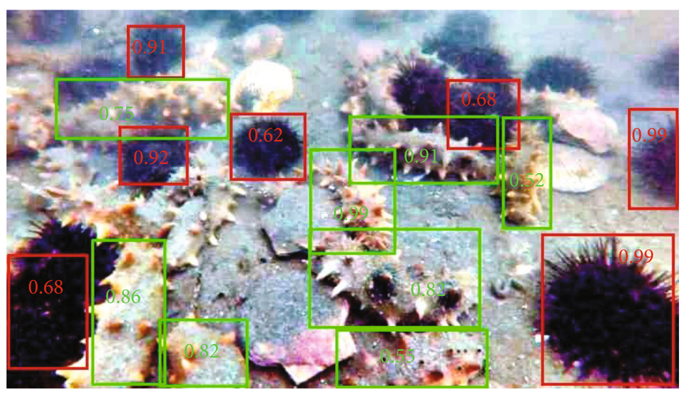
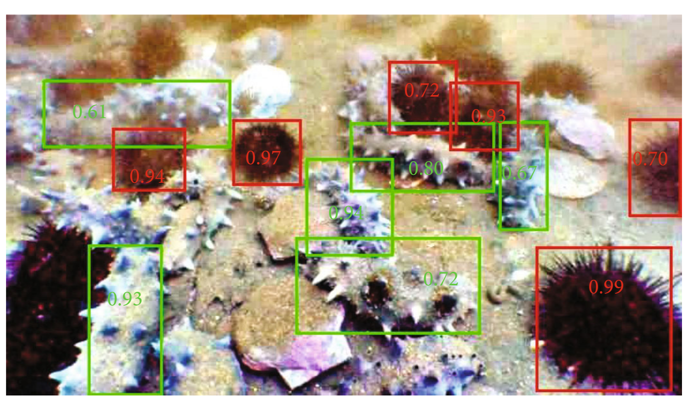


Figure 11: Overfitting loss changes.



(a) Clear water



(b) Muddy water

Figure 12: Recognition effect in clear and muddy environments.

sufficient. Furthermore, optimizing the autoMSRCR algorithm and adding penalty items reduce the distortion after image processing.

The self-developed underwater fishing robot will be arranged for launching experiments. The underwater operation of the robot is controlled by the control panel, and the underwater monitoring image and model detection results are displayed on the screen.

The recognition effect of the underwater robot in clear and muddy environments is shown in Figure 12, where "green" color represents sea cucumber and "red" color represents sea urchin.

The result shows that the detection effect is better in clear water. The image restored by autoMSRCR in the muddy water shows color distortion, which causes poor detection effect. Generally, the model has certain feasibility and reliability. In order to further improve the robustness and application level of the model, muddy water quality and data sets under different illumination conditions can be supplemented to train the model.

5. Conclusion

The YOLOv4 target detection platform is built by Linux Ubuntu 18.04 system, and target detection models of two species of sea cucumber and sea urchin are obtained through training. The main conclusions are as follows:

(1) k-means clustering algorithm is adopted to calculate the size and location coordinates of the prediction frame. The YOLOv4 underwater sea cucumber and sea urchin detection model is trained by using the learning rate optimization strategy of cosine annealing. The results of model training are reliable

(2) The data sets adopted this time is manual annotation data sets, and some fuzzy targets are not marked. The detection accuracy is reduced and there are some cases of missed detection

(3) The model can also be further optimized, such as model lightweight, which can improve the model detection rate per second and make the monitoring effect more smoothness

(4) Experiments show that the target detection network model adopted in the paper can accurately identify the specified underwater organisms, such as sea cucumber and sea urchin. But if the number of samples increases, the identification accuracy will also improve

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Major Science and Technology Innovation Project of Shandong Province (2019JZZY020703), Science and Technology Support Plan for Youth Innovation in Universities of Shandong Province Colleges and Universities (2019KJB014), and Shandong Jiaotong University "Climbing" Research innovation Team Program (SDJTUC1805).

References

[1] Z. Wang, M. Lin, and C. Ban, "Research on hydrodynamics analysis and double loop integral sliding mode control of 4-joint underwater manipulator," in *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 728–733, Jeju, South Korea, 2017.

[2] C. Dai, M. Lin, Z. Guan, and Y. Liu, "Aquatic organism recognition using residual network with inner feature and kernel calibration module," *Computers and Electronics in Agriculture*, vol. 190, article 106366, pp. 1–13, 2021.

[3] S. Herzog, C. Tetzlaff, and F. Wrgtter, "Evolving artificial neural networks with feedback," *Neural Networks*, vol. 123, pp. 153–162, 2020.

[4] X. Cao, J. Yao, Z. Xu, and D. Meng, "Hyperspectral Image Classification with Convolutional Neural Network and Active Learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 7, pp. 4604–4616, 2020.

[5] X. Xiaolong, H. Li, X. Weijie, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: A survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[6] S. K. Patnaik, C. Narendra Babu, and M. Bhave, "Intelligent and adaptive web Data extraction system using convolutional and long short-term memory deep learning networks," *Big Data Mining and Analytics*, vol. 4, no. 4, pp. 279–297, 2021.

[7] T. Zhao, F. Ye, Y. Ming, H. Liu, and S. Basodi, "A Survey on Algorithms for Intelligent Computing and Smart City Applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.

[8] F. Dai, P. Huang, X. Xiaolong, L. Qi, and M. Khosravi, "Spatio-Temporal Deep Learning Framework for Traffic Speed Forecasting in IoT," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 66–69, 2020.

[9] Z. Guan, Z. Dong, M. Lin, and J. Li, "Mechanical Analysis of Remotely Operated Vehicle," in *2018 4th International Conference on Control, Automation and Robotics*, pp. 446–450, Auckland, New Zealand, 2018.

[10] C. Dai, Z. Guan, and M. Lin, "Single low-light image enhancer using Taylor expansion and fully dynamic convolution," *Signal Processing*, vol. 189, article 108280, pp. 1–14, 2021.

[11] G. Hou, *Research on Underwater Image Enhancement and Object Recognition Algorithms*, Ocean University of China, 2015.

[12] X. Zhang, J. Qin, and Z. Jia, "Study on the Application of Multi-scale Retinex to Image Defogging Algorithm," *Journal of Xichang University (Natural Science Edition)*, vol. 35, no. 3, pp. 60–65, 2021.

[13] C. Dai, M. Lin, Z. Wang, D. Zhang, and Z. Guan, "Color Compensation Based on Bright Channel and Fusion for Underwater Image Enhancement," *Acta Optica Sinica*, vol. 38, no. 11, pp. 1–10, 2018.

[14] X. Xu, Z. Fang, J. Zhang et al., "Edge Content Caching with Deep Spatiotemporal Residual Network for IoV in Smart City," *ACM Transactions on Sensor Networks (TOSN)*, vol. 17, no. 3, pp. 1–33, 2021.

[15] A. L. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection. CVPR, Seattle USA," 2020, https://arxiv.org/abs/2004.10934.

[16] G. Ziyan, H. Huiyan, and H. Ligang, "Gesture Recognition Algorithm and Application Based on Improved YOLOV4," *Journal of North University of China (Natural Science Edition)*, vol. 42, no. 3, pp. 223–231, 2021.

[17] R. Shi, D. Jiang, and Q. Fang, "Aircraft target detection in remote sensing image based on YOLOv4," *Bulletin of Surveying and Mapping*, vol. S1, pp. 134–138, 2021.

[18] X. Xiaolong, Q. Huang, X. Yin, M. Abbasi, M. Khosravi, and L. Qi, "Intelligent Offloading for Collaborative Smart City Services in Edge Computing," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7919–7927, 2020.

[19] Z. N. Mohammad, F. Farha, A. O. M. Abuassba, S. Yang, and F. Zhou, "Access Control and Authorization in Smart Homes: A Survey," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 906–917, 2021.

[20] S. Xia, D. Peng, D. Meng et al., "Ball k-Means: Fast Adaptive Clustering with No Bounds," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, Piscataway, NJ, 2020.

[21] Z. H. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: faster and better learning for bounding box regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12993–13000, 2020.

[22] X. Qiao, *Sea Cucumber Identification in real-time Based on Underwater Machine Vision Technique*, China Agricultural University, 2017.