Journal of
Marine Science and Engineering

MDPI

Article

An Improved YOLOv5s-Based Scheme for Target Detection in a Complex Underwater Environment

Chenglong Hou [1], Zhiguang Guan [1,*], Ziyi Guo [1], Siqi Zhou [1] and Mingxing Lin [2]

[1] Shandong Provincial Engineering Lab of Traffic Construction Equipment and Intelligent Control, Shandong Jiaotong University, Jinan 250357, China; 21106009@stu.sdjtu.edu.cn (C.H.); zyg_sddy@163.com (Z.G.); 21106011@stu.sdjtu.edu.cn (S.Z.)
[2] School of Mechanical Engineering, Shandong University, Jinan 250061, China; mxlin@sdu.edu.cn
* Correspondence: guanzhiguang@sdjtu.edu.cn

Abstract: At present, sea cucumbers, sea urchins, and other seafood products have become increasingly significant in the seafood aquaculture industry. In traditional fishing operations, divers go underwater for fishing, and the complex underwater environment can cause harm to the divers' bodies. Therefore, the use of underwater robots for seafood fishing has become a current trend. During the fishing process, underwater fishing robots rely on vision to accurately detect sea cucumbers and sea urchins. In this paper, an algorithm for the target detection of sea cucumbers and sea urchins in complex underwater environments is proposed based on the improved YOLOv5s. The following improvements are mainly carried out in YOLOv5s: (1) To enhance the feature extraction ability of the model, the g^nConv-based self-attentive sublayer HorBlock module is proposed to be added to the backbone network. (2) To obtain the optimal hyperparameters of the model for underwater datasets, hyperparameter evolution based on the genetic algorithm is proposed. (3) The underwater dataset is extended using offline data augmentation. The dataset used in the experiment is created in a real underwater environment. The total number of created datasets is 1536, and the training, validation, and test sets are randomly divided according to the ratio of 7:2:1. The divided dataset is input to the improved YOLOv5s network for training. The experiment shows that the mean average precision (mAP) of the algorithm is 94%, and the mAP of the improved YOLOv5s model rises by 4.5% compared to the original YOLOv5s. The detection speed increases by 4.09 ms, which is in the acceptable range compared to the accuracy improvement. Therefore, the improved YOLOv5s has better detection accuracy and speed in complex underwater environments, and can provide theoretical support for the underwater operations of underwater fishing robots.

Keywords: underwater target detection; improved YOLOv5; g^nConv; HorBlock; hyperparameter evolution; data augmentation

Citation: Hou, C.; Guan, Z.; Guo, Z.; Zhou, S.; Lin, M. An Improved YOLOv5s-Based Scheme for Target Detection in a Complex Underwater Environment. *J. Mar. Sci. Eng.* **2023**, *11*, 1041. https://doi.org/10.3390/jmse11051041

Academic Editor: Sergei Chernyi

Received: 14 April 2023
Revised: 1 May 2023
Accepted: 11 May 2023
Published: 13 May 2023

1. Introduction

In traditional aquaculture, high nutritional seafood, such as sea cucumbers and sea urchins, has been loved by fishers. Unlike land plants, sea cucumbers and sea urchins live on underwater reefs at depths of 12–13 m. The traditional fishing method mainly relies on two methods, netting and underwater fishing by divers [1]. Although netting can greatly reduce the cost of fishing, it will cause serious damage to the ecological environment of the seafloor in the long term. Manual fishing can avoid damage to the seafloor ecological environment, but the cost of manual fishing is relatively high. Therefore, the research on underwater fishing robots has greatly promoted the development of underwater aquaculture industry [2,3]. In order to realize the automation of underwater robots for sea cucumber and sea urchin fishing operations, the target detection of sea cucumber and sea urchin is especially important.

Underwater imaging quality is poor due to the complex underwater environment that are susceptible to light and other factors [4]. The current detection for sea cucumbers

and sea urchins is mainly optimized in two aspects: on the one hand, the target detection network is optimized to enhance the extraction of underwater image features; on the other hand, image enhancement algorithms are used to improve the quality of underwater images and make the features of underwater images more obvious [5]. In this paper, the target detection network is optimized by enhancing the model's ability to extract features from underwater images, so as to achieve the accurate detection of sea cucumber and sea urchin in complex underwater environments. Currently, some researchers are applying the target detection framework based on convolutional neural networks (CNNs) to the underwater aquaculture industry, which plays an important role in the identification, monitoring, and fishing of underwater seafood [6]. Algorithms on underwater target detection have also been developed; Chen et al. [7] proposed a RetinaNet-based target detection algorithm for underwater robot, which uses Dense Net instead of ResNet to build a backbone network, and uses convolutional layer stacking instead of the original single convolutional operation to reduce the weight of the network. Qiang et al. [8] proposed an improved single shot multibox detector (SSD) target detection algorithm using ResNet instead of the VGG convolutional neural network of SSD, which led to a large improvement in the accuracy of the algorithm. Deng et al. [9] proposed a marine organism detection algorithm based on the improved SSD algorithm, in which a feature fusion module and a feature augmentation module were designed. The algorithm improves the overall detection accuracy of the algorithm for marine organism targets with only a small increase in computation and number of parameters. Wang et al. [10] proposed an underwater target image edge detection algorithm based on ant colony optimization and reinforcement learning, which can effectively extract underwater contour information, better maintain image texture, and has ideal anti-interference performance. Guo et al. [11] proposed an underwater target detection and localization method using feature maps and CNN-based classification, which is superior in underwater signal classification and target localization. Zeng et al. [12] proposed an underwater target detection based on Faster R-CNN and adversarial occlusion network, which can obtain better robustness for underwater seafood.

Currently, deep learning-based targets detection algorithms are mainly divided into one-stage algorithm and two-stage algorithm. The representative algorithms of one-stage are: OverFeat, YOLO series algorithm, SSD and RetinaNet, etc. The images inputting to the network enters directly into the convolutional neural network, and the features can be extracted to predict the classification and location of objects. Two-stage algorithms include: R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN and R-FCN, etc. Two-stage algorithms are characterized by taking the input image and performing candidate region proposal (RP) before classifying it by convolutional neural network. This feature can make the accuracy of the network improve, but due to the feature extraction of the candidate region, it will make the detection speed of the network drop greatly and cannot meet the real-time detection requirements. In underwater target detection, one-stage algorithms are widely used mainly because of its fast detection speed and relatively high detection accuracy. In the YOLO series algorithm, Li et al. [13] proposed an improved YOLOv3ST model, embedded two attention mechanisms, and proposed a K-means clustering algorithm to adapt to multi-scale feature prediction to improve underwater small target detection accuracy. Guo et al. [14] proposed a fast detection scheme for marine organisms based on an improved MSRCP image augmentation algorithm. Zhang et al. [15] proposed a lightweight underwater target detection method based on MobileNetV2, YOLOv4 [16] algorithm and attentional feature fusion. Chen et al. [17] proposed an improved YOLOv4 for detecting underwater targets by replacing the up-sampling module with a deconvolution module and incorporating depth-separable convolution into the network. Li et al. [18] proposed an improved YOLOv5s underwater scallop recognition algorithm, mainly using the group convolution and inverse residual modules instead of the backbone network. The mAP was able to reach 88.2%. Li et al. [19] proposed an improved YOLOv5 based on adding triplet attention and multi-scale detection to meet the shortcomings existing for small target detection. Li et al. [20]

proposed an improved CME-YOLOv5 network to detect fish and small targets in dense groups, and the algorithm showed good detection performance when applied to densely distributed fish and small targets. Currently, the YOLO series algorithms have been applied to the detection of underwater targets. YOLOv5 has smaller trained weight files, faster detection, and higher mAP compared to YOLOv3, YOLOv4 and other models [21]. However, due to the poor imaging quality of the underwater dataset, the original YOLOv5 model did not achieve the ideal results in the target detection process [22].

Therefore, in order to accurately and quickly identify sea cucumbers and sea urchins in the complex environment underwater, an improved YOLOv5s underwater target detection algorithm is proposed to complete underwater fishing operations more effectively. The research content and innovation are mainly in the following aspects: (1) For the underwater dataset with inconspicuous image features, the backbone network of YOLOv5s is improved so that its ability to extract image features is enhanced, thus improving the recall of sea cucumbers and sea urchins. (2) For the accuracy problem of model training, the hyperparameters controlling the model training are optimized by using genetic algorithm, so as to improve the accuracy of model training. (3) For the overfitting phenomenon of model training caused by the lack of underwater datasets, offline data augmentation is performed on the datasets, which can effectively improve the generalization ability of the model.

2. Materials and Methods

2.1. Dataset Creation

The dataset used in the experiment is from the Zhanjiang Underwater Robotics Competition URPC2021 Underwater Optics Competition dataset. The dataset uses two of the categories: sea cucumbers and sea urchins. The images of the dataset are obtained from videos taken by underwater robots in real underwater environments. The image resolutions are 720 × 405 and 400 × 300 pixels. The selected dataset includes 1536 images in total, and the training set, validation set and test set are randomly divided at the ratio of 7:2:1. After the division is completed, the annotation information, category ratio and size distribution are counted again to ensure that the distributions of training set, validation set and test set are similar. In the process of dataset labelling, the Labelimg software with version number 1.8.6 was used to manually label sea cucumbers and sea urchins with rectangular boxes, and finally the labelling file in YOLO format was obtained. The labeling process of the Labelimg software is shown in Figure 1.

Figure 1. The Labelimg software labeling process.

There are two types of data augmentation: one is offline data augmentation and the other is online data augmentation. In order to improve the generalization ability of the model, online data augmentation and offline data augmentation are used in the dataset. The online data augmentation used in this paper is Mosaic data augmentation, which is mainly used to generate a new image by stitching four randomly selected images into one image with a random size ratio. The advantage of this operation is not only to increase the richness of the image itself, but also to increase the number of samples. Mosaic data augmentation is shown in Figure 2.

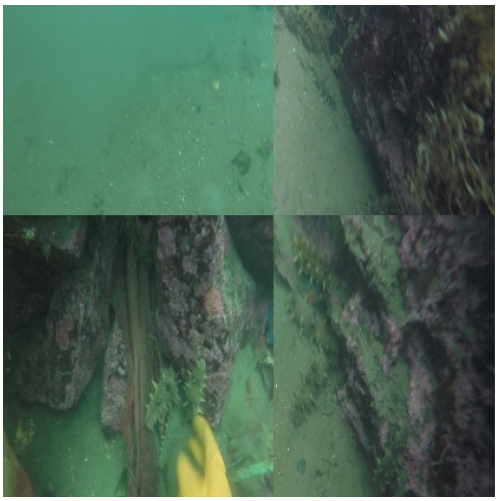

Figure 2. Mosaic data augmentation.

The offline data augmentation is performed by randomly translating, flipping, cropping and rotating each image in several ways at random combinations. An image generated after offline augmentation is shown in Figure 3. After the offline data augmentation, the training set, the validation set and test set are respectively increased by 4196, 1245 and 623 images. Meanwhile, the number of datasets samples before and after offline data augmentation and the number of category labels are shown in Table 1. After the dataset augmentation is completed, the dataset needs to be manually labelled.

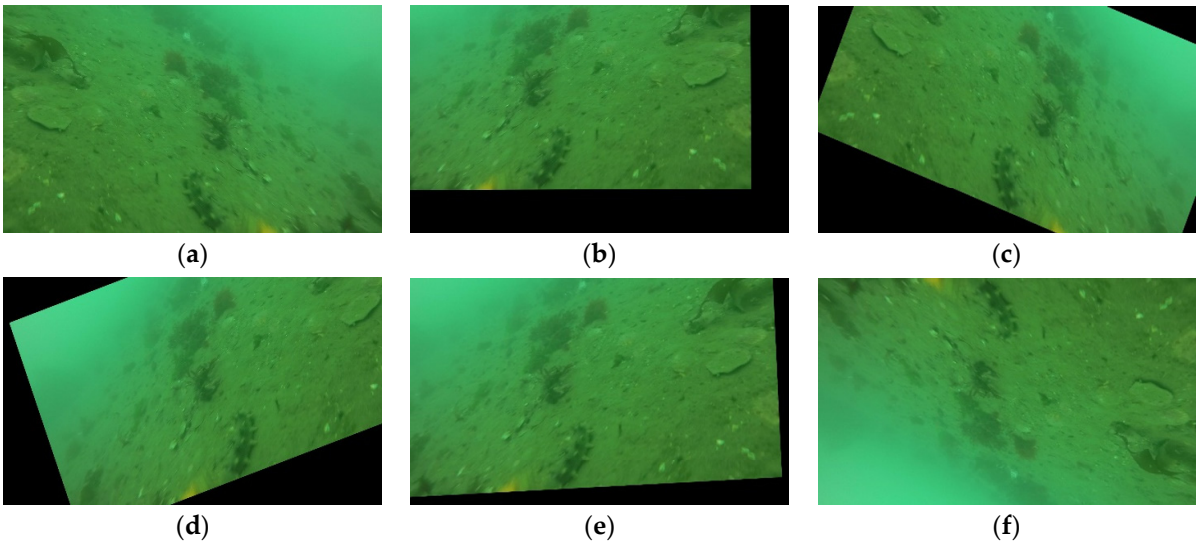

Figure 3. Image offline data augmentation: (**a**) Original image; (**b**) Translate + horizontal flip; (**c**) Translate + rotate; (**d**) Horizontal flip + rotate + translate; (**e**) Rotate (**f**) Vertical flip.

Table 1. Number of data set labels.

Category	Dataset		
	Training set (Original/Augmentation)	Validation set (Original/Augmentation)	Test set (Original/Augmentation)
Sea Cucumber	2771/15774	801/4482	446/2276
Sea Urchin	4403/22374	1339/6918	644/3016

2.2. YOLOv5 Target Detection Algorithm

YOLOv5 target detection is a regression-based algorithm in the one-stage algorithm. When the image is input into the network, it goes directly into the convolutional neural network for feature extraction, and in the output layer, the object classification and location regression prediction is carried out. In this paper, the newer version 6.1 of YOLOv5 is used. For ensuring the detection accuracy, the lightweight version of the model is also fully considered; YOLOv5s is selected for training in the experiment. The YOLOv5s network structure is shown in Figure 4.

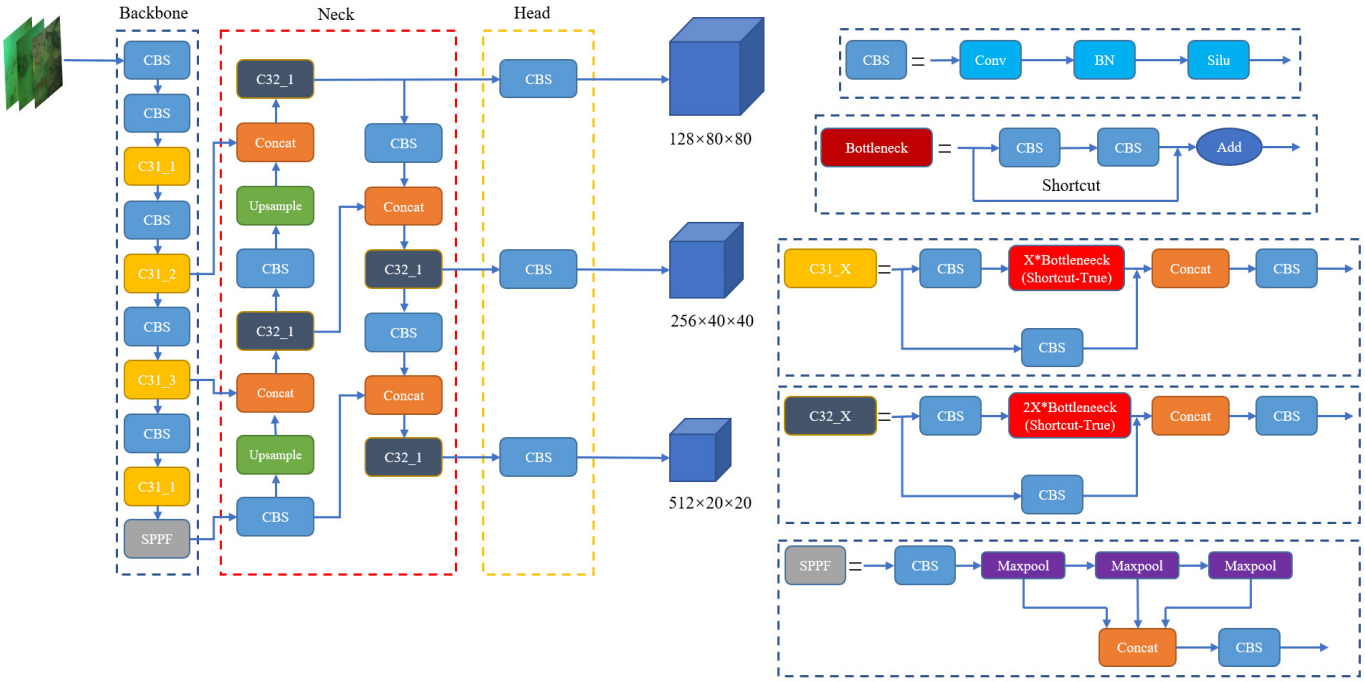

Figure 4. YOLOv5s network structure.

The network structure of YOLOv5s is mainly composed of the backbone, neck, head and detect, whose structural components and features are as follows:

(1) The backbone is the feature extraction network of YOLOv5s, which is mainly composed of the convolutional module, C3 module and SPPF module [23]. When the image enters the network, it goes through these modules for feature extraction, respectively, and the extracted features can be called the effective feature layer, which contains the feature information of the image.

(2) The neck is an enhanced feature extraction network composed of FPN, which mainly fuses features from three effective feature layers of different scales in the backbone part. The fusion process is mainly upward and downward fusion in the form of feature pyramids. At the same time, YOLOv5s uses down-sampling for large-scale effective feature layers, and fuses the features with other scale effective feature layers, which greatly preserves the semantic features of the images.

(3) The head performs the classification and regression of network prediction. CIoU loss [24] is applied to compute the loss in training process. At the same time, the generated multiple prediction bounding boxes are suppressed away by non-maximal suppression to obtain a most accurate prediction bounding box.

The YOLOv5 target detection algorithm can achieve a relatively high mAP for target detection of land-based clear datasets. However, target detection of underwater datasets still needs to be optimized to recognize sea cucumbers and sea urchins easily. When optimizing the target detection algorithm, the main focus is to improve the accuracy of the algorithm in the underwater environment. Moreover, the detection speed of the algorithm should be ensured.

2.3. Improved YOLOv5s Network Design

The backbone is mainly improved in the YOLOv5s network, which is used to extract features from the images of sea cucumbers and sea urchins. In this paper, the HorBlock module is added after each C3 module. The HorBlock module is a self-attentive sub-layer based on g^nConv to achieve an arbitrary order of priority complexity exchanges. The improved YOLOv5s backbone improves the feature extraction ability for sea cucumbers and sea urchins. The structure diagram of the improved YOLOv5s network is shown in Figure 5. Meanwhile, the hyperparameters are optimized using a genetic algorithm, and the original hyperparameters of YOLOv5s are evolutionary transferred based on the COCO dataset. The experiment uses the evolved hyperparameters based on the improved network model to obtain the optimal hyperparameters, which can effectively improve the recognition accuracy of the network. In YOLOv5s version 6.1, three scales feature maps are output in the head: 20 × 20, 40 × 40 and 80 × 80. Sea cucumbers and sea urchins appear larger in the image when they are closer to the lens. In this case, the 20 × 20 scale is used to predict the larger objects in the image. However, sea cucumbers and sea urchins appear smaller in the image when they are further from the lens; 40 × 40 and 80 × 80 scales are used to predict medium and small-sized objects in the image. The parameters of the improved YOLOv5s network are shown in Table 2.

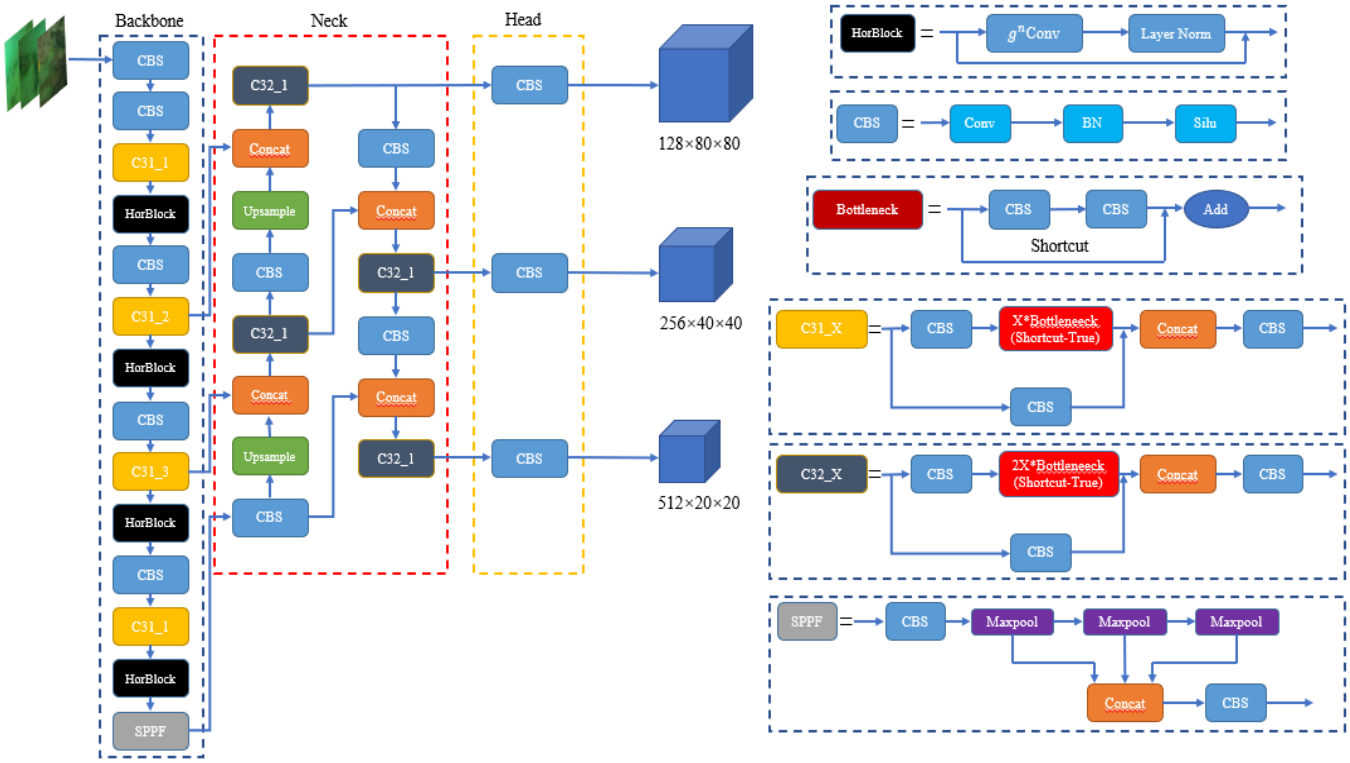

Figure 5. Improved YOLOv5s network structure.

Table 2. Improved YOLOv5s network parameters.

Layer Number	Network Layer	Input Dimension	Stride
0	Conv 6 × 6	3 × 640 × 640	2
1	Conv 3 × 3	32 × 320 × 320	2
2	C3	64 × 160 × 160	-
3	HorBlock	64 × 160 × 160	-
4	Conv 3 × 3	64 × 160 × 160	2
5	C3	128 × 80 × 80	-
6	HorBlock	128 × 80 × 80	-
7	Conv 3 × 3	128 × 80 × 80	2
8	C3	256 × 40 × 40	-
9	HorBlock	256 × 40 × 40	-
10	Conv 3 × 3	256 × 40 × 40	2
11	C3	512 × 20 × 20	-
12	HorBlock	512 × 20 × 20	-
13	SPPF	512 × 20 × 20	-
14	Conv 1 × 1	512 × 20 × 20	1
15	nn.Upsample	256 × 20 × 20	-
16	Concat	256 × 40 × 40	-
17	C3	512 × 40 × 40	-
18	Conv 1 × 1	512 × 40 × 40	1
19	nn.Upsample	128 × 40 × 40	-
20	Concat	128 × 80 × 80	-
21	C3	256 × 80 × 80	-
22	Conv 3 × 3	128 × 80 × 80	2
23	Concat	128 × 40 × 40	-
24	C3	256 × 40 × 40	-
25	Conv 3 × 3	256 × 40 × 40	2
26	Concat	256 × 20 × 20	-
27	C3	512 × 20 × 20	-
28	Detect	128 × 80 × 80 256 × 40 × 40 512 × 20 × 20	-

2.3.1. Self-Attentive Sub-Layer HorBlock Based on g^nConv

Due to the complex environment underwater, some sea cucumbers and sea urchins cannot be recognized effectively, and there are problems of missed detection and low accuracy. Therefore, in order to solve this problem, the HorBlock module, which is a self-attention sub-layer based on g^nConv, is added to the YOLOv5s model. Its main purpose is to improve the recognition accuracy of sea cucumbers and sea urchins, and reduce the influence of the surrounding environment. The g^nConv used in the HorBlock module is designed to perform higher-order spatial interactions through gated convolution and recursion. It is highly flexible, customizable and compatible, and able to extend second-order interactions in self-attentiveness to arbitrary orders without much additional computation. The g^nConv can be used as a plug-and-play module to form an independent module or embedded in various convolution-based models. The g^nConv is built using standard convolution, linear projection, and element multiplication, whose basic operation is gated convolution ($gConv$). Let $t \in I^{H\times W\times C}$ be the input feature and the output of the gated convolution $y = gConv(t)$ can be written as [25]:

$$\left[b_0^{H\times W\times C},\ d_0^{H\times W\times C}\right] = \theta_{in}(t) \in I^{H\times W\times 2C} \tag{1}$$

$$y = \theta_{out}(f(d_0) \odot b_0) \in I^{H\times W\times C} \tag{2}$$

where, θ_{in}, θ_{out} are linear projection layers and f is the depth convolution. Note that $(f(d_0) \odot b_0)^{(i,c)} = \Sigma_{j\in\Omega}\ w_{i\to j}^{c} d_0^{(j,c)} b_0^{(i,c)}$, where, Ω_i is the local window centered on i and w

is the convolution weight of f. Thus, the above equation introduces the interaction between adjacent features $b_0^{(i)}$ and $d_0^{(i)}$ via element-wise multiplication. Since each $b_0^{(i)}$ interacts with its neighboring feature $d_0^{(i)}$ only once, the interaction in $gConv$ is considered as a *1-order* interaction.

After implementing *1*-order spatial interactions with $gConv$, g^nConv is constructed by introducing higher-order interactions. First, a set of projection features b_0 and $\{d_m\}_{m=0}^{n-1}$ are obtained using θ_{in} [25]:

$$\left[b_0^{H\times W\times C_0},\ d_0^{H\times W\times C_0},\ldots,d_{n-1}^{H\times W\times C_{n-1}}\right]=\theta_{in}(t)\in I^{H\times W\times(C_0+\sum_{0\le m\le n-1}C_m)} \tag{3}$$

Then, perform the gated convolution recursively using [25]:

$$b_{m+1}=f_m(d_m)\odot g_m(b_m)/\delta,\qquad m=0,1\ldots,n-1 \tag{4}$$

where, they scale the output by $1/\delta$ to stabilize the training. $\{f_m\}$ are a set of depth-wise convolution layers, and $\{g_m\}$ are used to match the dimension in different orders [25]:

$$g_m=\begin{cases}Identity, & m=0\\ Linear(C_{m-1},C_m), & 1\le m\le n-1\end{cases} \tag{5}$$

$$C_m=\frac{C}{2^{n-m-1}},\qquad 0\le m\le n-1 \tag{6}$$

C_m is the channel dimension, and finally, the last recursive output d_n is fed into the projection layer θ_{out} to obtain the result of g^nConv.

In this paper, spatial modeling operations that are representative and perform different interaction sequences are compared, as shown in Figure 6, where the blue structure is a feature and the gray structure is the feature's adjacent region. Each spatial model individually shows the spatial interactions between the feature and the neighboring regions. (a) Standard convolutional operations do not explicitly consider spatial interactions. (b) Dynamic convolution introduces the SE module, which makes the weights dynamic and improves the modeling ability of convolution by adding channel interactions [26–28]. (c) The self-attentive operation performs second-order spatial interactions through two consecutive matrix multiplications [29]. (d) The g^nConv uses gated convolution and a recursive design to implement arbitrary-order spatial interactions [25].

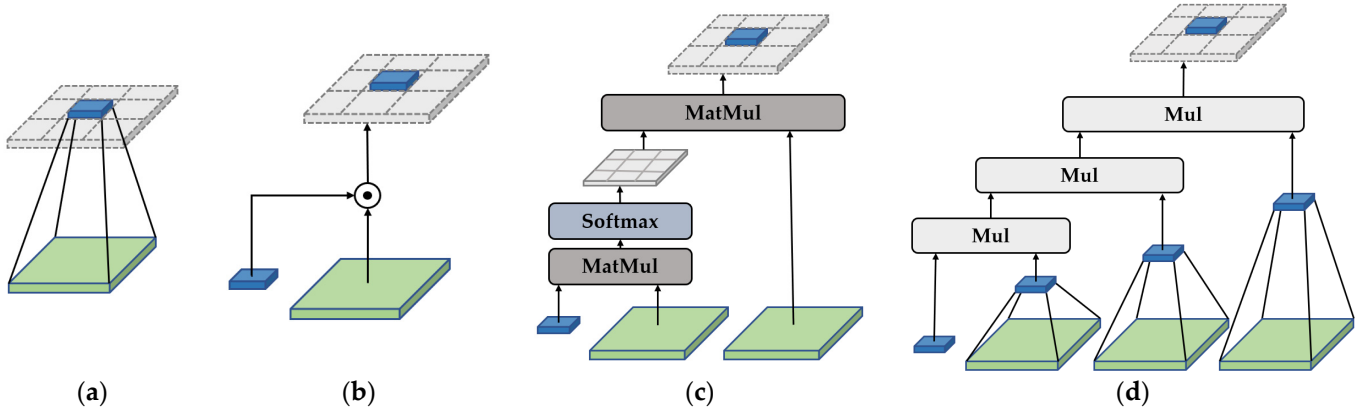

Figure 6. Representative spatial modeling operations performing different interaction sequences: (**a**) Standard convolutional operations; (**b**) Dynamic convolution operations; (**c**) Second-order spatial interaction operations; (**d**) Arbitrary-order space interaction operations.

The HorBlock constructed by combining the g^nConv module with Layer Norm is incorporated into the backbone network, which can enhance the network's performance by further extracting the input images features.

2.3.2. Hyperparametric Evolution Based on a Genetic Algorithm

Hyperparameter evolution uses a genetic algorithm for optimization [30]. During model training, hyperparameters control all aspects of model training. Having the best hyperparameters is important to the training outcome, but it is also a greater challenge. Due to the unknown correction between high-dimensional search space and the fitness of each point, traditional hyperparameter evolution methods such as grid searches are difficult to meet the requirements. Therefore, hyperparameter evolution based on a genetic algorithm is a better selection for hyperparameter optimization methods [31,32], which can obtain desired results.

There are 29 hyperparameters in YOLOv5s that regulate the training process. With such a large number of hyperparameters, the optimal solution for each parameter can be obtained using a genetic algorithm, and then can get the optimal hyperparameter results. For each hyperparameter in the training process, it can be considered as an independent variable x. The result obtained by performing network training is y. In YOLOv5s, seven results for representation are used. They are: "metrics/precision", "metrics/recall", "metrics/mAP 0.5", "metrics/mAP 0.5:0.95", "val/box loss", "val/obj loss" and "val/cls loss". At the same time, a fitness function needs to be defined to judge the effect of the input x. Fitness is the precision (P), recall (R), and the fraction-weighted sum of mAP 0.5 and mAP 0.5:0.95, independent of the loss part. For the variant x, after training, the result y is returned. If the calculated fitness is higher than the current one, the variant can be considered effective, and the parameters of the variant will be saved at this time. This experiment is set to mutate once every 50 epochs, 300 times in total. The one with the best mutation effect is selected as the final result of mutation. A comparison of the hyperparameter evolution results is shown in Table 3.

Table 3. Comparison of hyperparameter evolutionary results.

Parameter	lr0	lrf	Momentum	Weight Decay	Warmup Epochs	Warmup Momentum	Warmup Bias lr	Box	cls	cls pw
Original	0.001	0.01	0.937	0.0005	3.0	0.8	0.1	0.05	0.5	1.0
After Evolution	0.00886	0.01	0.85395	0.0004	3.0549	0.84169	0.06114	0.10873	0.2	1.0017

Parameter	obj	obj pw	iou t	anchor t	fl gamma	hsv h	hsv s	hsv v	degrees
Original	1.0	1.0	0.20	4.0	0.0	0.015	0.7	0.4	0.0
After Evolution	1.0651	0.75344	0.2	3.2016	0.0	0.01024	0.36303	0.18256	0.0

Parameter	translate	scale	shear	perspective	flipud	fliplr	mosaic	mixup	copy paste	anchors
Original	0.1	0.5	0.0	0.0	0.0	0.5	1.0	0.0	0.0	3
After Evolution	0.14064	0.39425	0.0	0.0	0.0	0.5	1.0	0.0	0.0	3.9496

The main genetic operator for hyperparametric evolution uses crossover and mutation. During the training process, it is the offspring created with 80% probability and 0.04 variance based on the best combination of parents from all previous generations. The results are saved separately for each time and for the highest fitness. Each hyperparameter is visualized in the evolutionary process, as shown in Figure 7. Each figure corresponds to a hyperparameter, where the x-axis represents the value of the hyperparameter and the y-axis represents fitness. The concentration of data points gradually decreases as the color changes from yellow to gray. The yellow color indicates a higher concentration of data

points, while the vertical line indicates that a parameter has been fixed and will no longer be subject to further mutation.

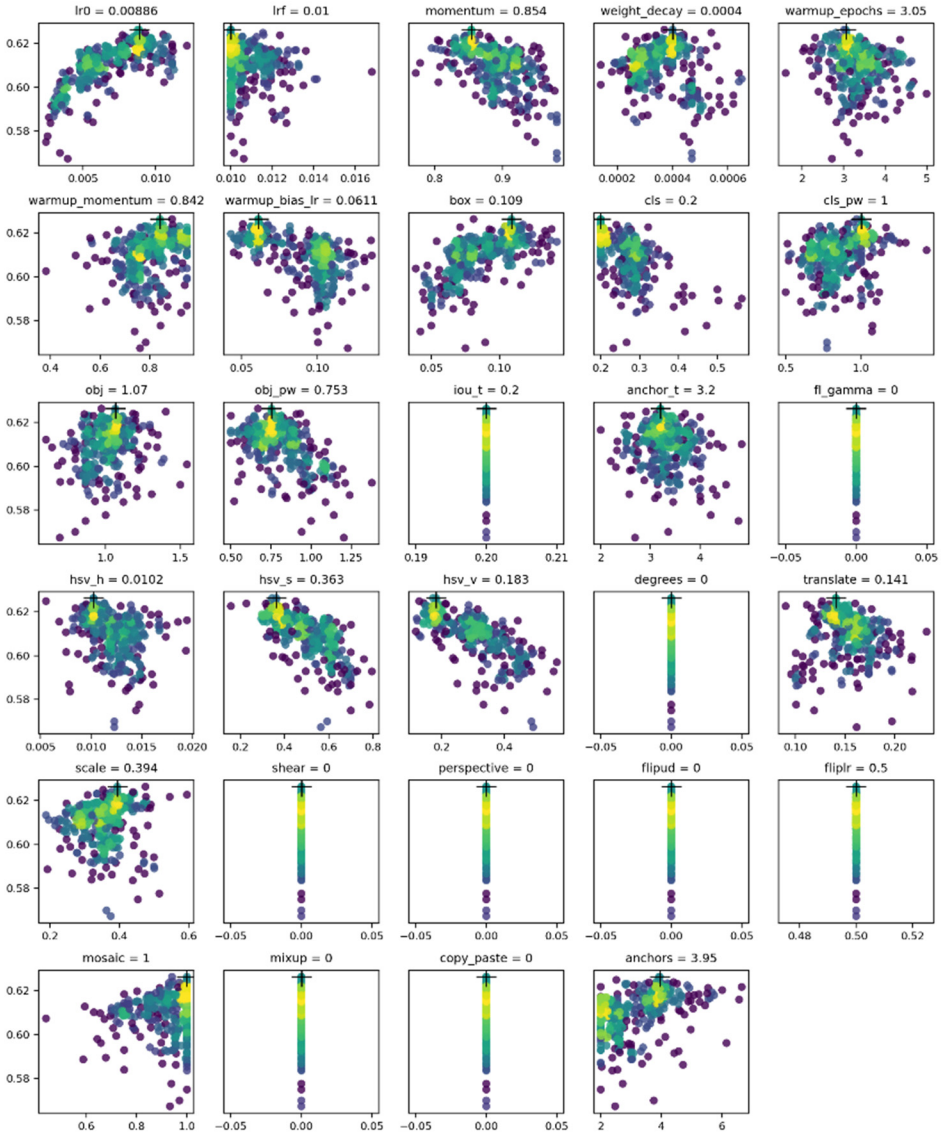

Figure 7. Visualization of the hyperparametric evolutionary process.

2.4. Evaluation Indicators

The loss function of YOLOv5s mainly consists of three components: bounding box regression loss, classification loss and objectness loss. The loss function is calculated as follows [33]:

$$L = L_{box} + L_{cls} + L_{obj} \tag{7}$$

where, L_{obj} is obtained by the CIoU loss function, and the CIoU calculation formula is as follows [33]:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,\ b^{gt})}{c^2} + av \tag{8}$$

$$a = \frac{v}{1 - IoU + v} \tag{9}$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2 \tag{10}$$

where, $\rho^2(b, b^{gt})$ is the Euclidean distance between the real bounding box and the prediction bounding box. c is the length of the diagonal of the smallest outer rectangle of the real bounding box and the prediction bounding box. v is the distance between the aspect ratio of the real bounding box and the prediction bounding box. a is the weight coefficient. w is the width of the prediction bounding box. h is the height of the prediction bounding box. w^{gt} is the width of the real bounding box. h^{gt} is the height of the real bounding box.

In the classification loss, the loss is calculated via binary cross entropy, which is characterized by the fact that it is calculated only when the detected target is a positive sample. The bounding box regression loss is divided into two parts: loss with the target and loss without the target, which is different from the other two loss functions in that it is calculated regardless of positive or negative samples.

3. Results and Discussion

3.1. Model Training

The model training platform is achieved based on the Python 3.8 PyTorch 1.10 deep learning framework. The system environment is Windows 10, and training is performed using a GPU configured with CUDA version 11.1 and the neural network acceleration library cuDNN. The overall configuration of the training environment is shown in Table 4.

Table 4. Training environment configuration table.

Item	Parameter
Operating System	Windows 10
Central Processing Unit	Intel(R) Core(TM) i5-11400F
GPU	GeForce RTX 3090
Graphics Card Driver	CUDA 11.1
Software Environment	OpenCV-Python 4.5.5.62
Deep Learning Framework	PyTorch 1.10.2

All models are not loaded with pre-training weights during the training process. As the hyperparameter evolution based on the genetic algorithm is adopted, the default hyperparameters are no longer used in model training. The initial learning rate is 0.00886. At this learning rate, the model converges faster. As mAP continues to increase and the loss continues to decrease, the model reaches its optimal state finally. The weight decay coefficient is 0.0004, and the momentum factor is 0.85395. In the training, the batch size is 32, the optimizer uses Adam optimizer [34] and epochs are 200. The image sizes in the input network are resized to the default size of 640 × 640 pixels. The loss variation curves and mAP variation curves of the improved YOLOv5s model and the original YOLOv5s model are shown in Figure 8.

From the curve changes in the figure, it can be seen that the two models start to converge rapidly after 20 epochs, when the loss values start to decrease rapidly and the mAP rises rapidly. After 50 epochs, the loss and the mAP changes start to become smaller and tend to stabilize. From Figure 8a, the bounding box regression loss of the improved YOLOv5s model is higher than that before the improvement, but the convergence speed is better than that of the original model. Because the initial learning rate of the improved model is larger when using hyperparameter evolution, bounding box regression loss of the model becomes larger. From Figure 8b,c, the curves for classification loss and objectness loss are smoother and less volatile. In terms of the objectness loss, overfitting occurred for the original model during training, whereas the improved model can solve the overfitting. From Figure 8d, the improved YOLOv5s model has a higher mAP in the training process, and the convergence speed is also faster.

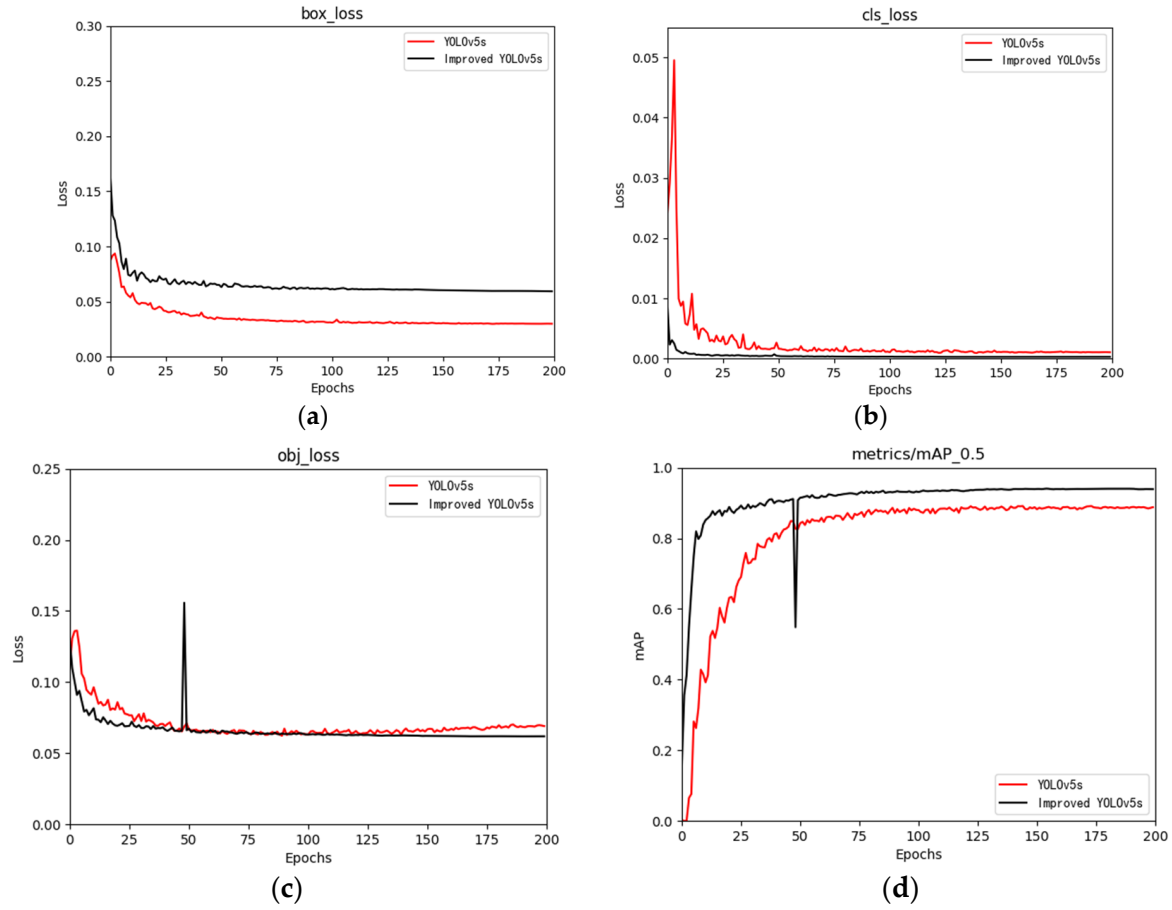

Figure 8. Loss and mAP variation curve: (**a**) Bounding box regression loss; (**b**) Classification loss; (**c**) Objectness loss; (**d**) mAP.

3.2. Analysis of Results

In this experiment, the average precision (AP), mAP and mean detection time will be used to evaluate the performance of the model. Their calculation equations are as follows [33]:

$$AP = \int_0^1 \frac{TP}{TP + FP} d \frac{TP}{TP + FN} \tag{11}$$

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{12}$$

where, TP is the number of cases where the true case is positive and the predicted outcome is positive. FP is the number of cases where the true case is negative and the predicted outcome is positive. FN is the number of cases where the true outcome is positive and the predicted outcome is negative.

3.2.1. Analysis of Ablation Experiments

Ablation experiments are carried out for different improvements of the improved YOLOv5s of the same dataset, which can get the impact of the fusion of self-attentive sub-layer HorBlock, hyperparameter evolution and offline data augmentation on the accuracy of the model. The comparison results are shown in Table 5.

Table 5. Comparison results of ablation experiments.

HorBlock	Hyperparametric Evolution	Offline Data Augmentation	mAP/%	Parameters/M	Average Detection Time/ms
			89.5	7.02	7.09
√			90.4	11.19	14.09
√	√		91.5	11.20	11.54
√	√	√	94.0	11.20	11.18

According to the data in Table 5, after adding the HorBlock module, the mAP of the improved YOLOv5s increases by 0.9% compared with the original YOLOv5s, which indicates that the network pays more attention to sea cucumbers and sea urchins and reduces the influence of useless features. While using the hyperparameter evolution, the mAP increases by 1.1%, which indicates that the adjustment of hyperparameters can help the network obtain a better result in the training process. The mAP increases by 2.5% with offline data augmentation, which indicates that the original samples are relatively not rich enough. The use of data augmentation can greatly enrich the samples and improve the generalization ability of the model. The final detection accuracy of the model increases by 4.5% compared with the original YOLOv5s. Therefore, the model is able to meet the experimental requirements in terms of the detection accuracy. The experimental result also shows that parameters in the improved YOLOv5s increases by 4.18 M compared with them in the original YOLOv5s and increases by 4.09 ms considering the average detection time. The comparison effect of the original model and the improved model is shown in Figure 9.

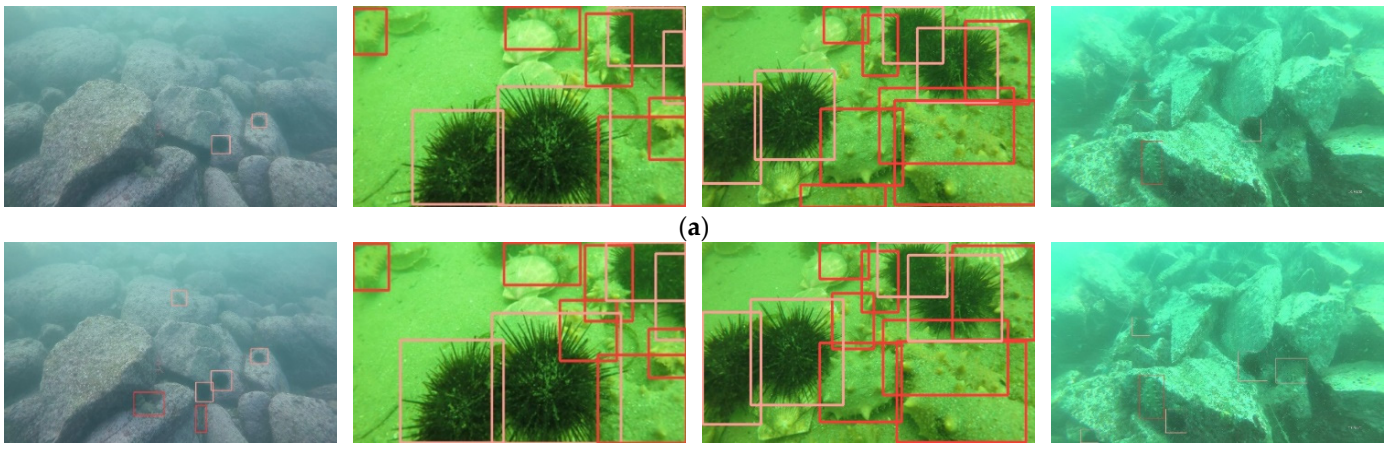

Figure 9. Comparison effect between the original model and the improved model: (**a**) Original model; (**b**) Improved model.

From Figure 8, it can be seen that both the detection accuracy and the recall rate are low, and the objectness loss of the prediction bounding box is high in the original model. The detection accuracy and recall rate are improved in the improved model. In addition, as shown by the comparison of the middle images, the objectness loss of the prediction bounding box of the improved model is lower and the bounding box selection of the detected sea cucumbers and sea urchins is more accurate.

3.2.2. Comparative Analysis of Model Detection Effects

To further validate the performance of the improved YOLOv5s model, a comparison experiment is conducted with RetinaNet, SSD, YOLOv4, YOLOv5m, YOLOV5l and YOLOv5x. In the experiment, all other models use default parameters and all models are not loaded with pre-training weights during the training process. The ratio of the training set, validation set and test set for all models during training is 7:2:1. The experi-

mental results are shown in Table 6. where the mAP of the improved YOLOv5s is 11.75%, 10.31%, 8.08%, 3.2%, 3.1% and 3% higher than that of RetinaNet, SSD, YOLOv4, YOLOv5m, YOLOV5l and YOLOV5x, respectively. The parameters of the improved YOLOv5s are 4.18 M larger than those of the original YOLOv5s, but smaller than those of the other models. The improved YOLOv5s has an average detection time that is 6.29 ms slower than that of SSD, but it is 93.84 ms, 6.46 ms, 6.63 ms, 6.18 ms and 2.63 ms faster than those of RetinaNet, YOLOv4, YOLOv5m, YOLOv5l and YOLOv5x, respectively. The AP of sea cucumber and sea urchin detected using the improved YOLOv5s are both better than those detected with the other models. The detection effect of the models is shown in Figure 10. The improved YOLOv5s, in terms of both missed and false detections, significantly improved, and the regression accuracy is improved compared to the other models.

Table 6. Comparison data between different models.

Models	Sea Cucumber AP/%	Sea Urchin AP/%	mAP/%	Parameters/M	Average Detection Time/ms
RetinaNet	72.23	92.27	82.25	36.1	105.02
SSD	78	90	83.69	24.5	4.89
YOLOv4	77	95	85.92	64.36	17.64
YOLOv5s	83.9	95.1	89.5	7.02	7.09
YOLOv5m	87.0	94.6	90.8	20.88	17.81
YOLOv5l	86.5	95.2	90.9	46.14	17.36
YOLOv5x	87.3	94.8	91.0	86.18	13.81
Improved YOLOv5s	91.2	96.7	94.0	11.20	11.18

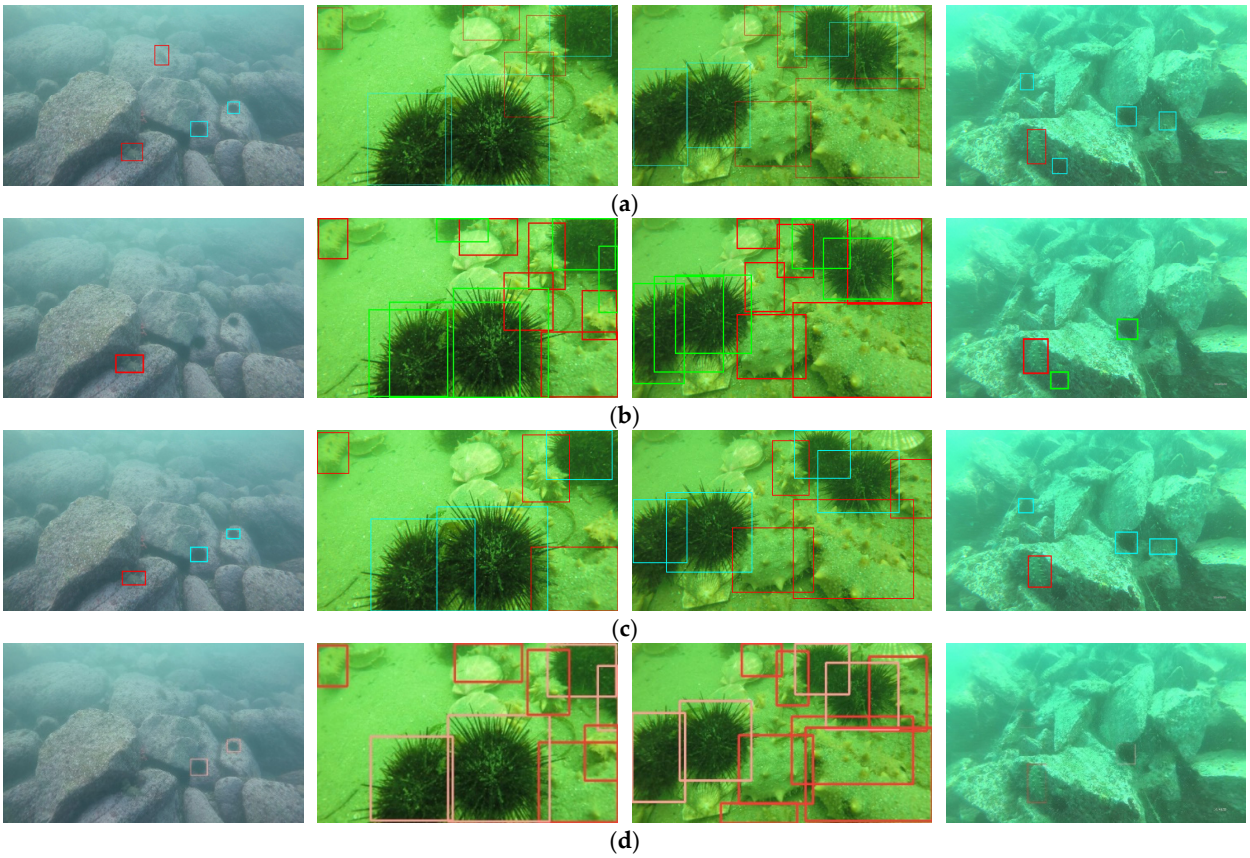

Figure 10. *Cont.*

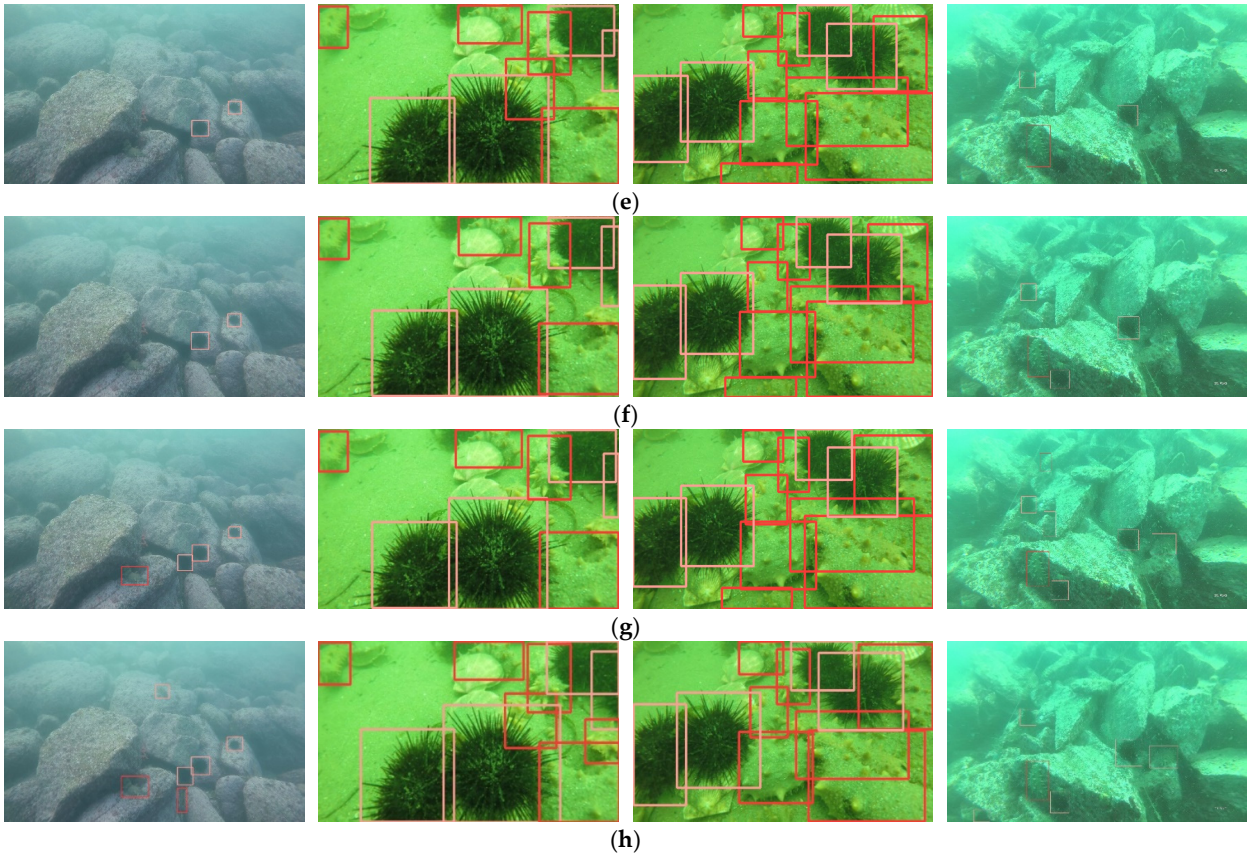

Figure 10. Inter-model detection results graph: (**a**) RetinaNet (**b**) SSD; (**c**) YOLOv4; (**d**) YOLOv5s; (**e**) YOLOv5m; (**f**) YOLOv5l; (**g**) YOLOv5x; (**h**) Improved YOLOV5s.

4. Conclusions

In this paper, a target detection algorithm based on improved YOLOv5s is proposed that can adapt to accurately recognize and detect sea cucumbers and sea urchins in complex underwater environments. The main improvement of YOLOv5s is to integrate the HorBlock module into the backbone network to enhance the feature extraction ability of the backbone network. The original hyperparameters of the improved YOLOv5s are adjusted using a hyperparameter evolution method based on a genetic algorithm to obtain hyperparameters suitable for complex underwater environments, which enables the model to obtain the optimal recognition accuracy during the training process. In this experiment, online and offline data augmentation methods are used to process the dataset. Mosaic data augmentation is used for online data augmentation, while offline data augmentation is achieved by expanding the dataset. During the experimental process, the ablation experiment and the comparison experiment of each model are conducted, respectively. In the ablation experiment, the methods of the HorBlock module, genetic algorithm hyperparameter evolution and offline data augmentation are experimentally verified. The final experimental results show that although the improved YOLOv5s has a slight decrease in detection speed, mAP increases by 4.5% compared to the original model. In the model comparison experiments, the improved YOLOv5s model has a higher mAP than the RetinaNet, SSD, YOLOv4, YOLOv5m, YOLOV5l and YOLOV5x models.

Due to the complex underwater environment, the underwater datasets have certain difficulties in the acquisition process. The underwater environment is affected by light and plankton in the water. This greatly affects the clarity of underwater images, which brings some difficulties in making the dataset and training the network model. Therefore, the image enhancement algorithms to improve the clarity of underwater images is a crucial task in the underwater data processing process. In future research, more kinds of datasets should

be collected and image enhancement algorithms should be added to the datasets during training and detection. Moreover, the network model should continue to be optimized and the detection speed should also be improved while continuing to improve the accuracy of the model, which will be a direction of continued research in the future.

Author Contributions: Conceptualization, C.H. and Z.G. (Zhiguang Guan); methodology, C.H., Z.G. (Zhiguang Guan) and M.L.; software, C.H.; validation, C.H.; data curation, S.Z.; writing—original draft preparation, C.H. and Z.G (Ziyi Guo); writing—review and editing, Z.G. (Zhiguang Guan) and M.L.; supervision, M.L.; funding acquisition, Z.G. (Zhiguang Guan) All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Natural Science Foundation of Shandong Province (ZR2020ME267), 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).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data cannot be shared at this time as the data also form part of an ongoing study.

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

References

1. Guan, Z.G.; Hou, C.L.; Zhou, S.Q.; Guo, Z.Y. Research on Underwater Target Recognition Technology Based on Neural Network. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 4197178. [CrossRef]
2. Kang, S.; Yu, J.C.; Zhang, J. Research Status of Micro Autonomous Underwater Vehicle. *Robot* **2023**, *2*, 218–237.
3. Li, L.; Yang, X.; Qin, X.J.; Wang, Y.Q. Research Status and Development Trend of Wall-climbing Cleaning Robots. *Mach. Build. Autom.* **2023**, *52*, 1–6.
4. Zhou, J.C.; Pang, L.; Zhang, W.S. Underwater image enhancement method by multi-interval histogram equalization. *IEEE J. Ocean. Eng.* **2023**, *48*, 474–488. [CrossRef]
5. Zhou, J.C.; Zhang, D.H.; Zhang, W.S. Cross-view enhancement network for underwater images. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105952. [CrossRef]
6. Yuan, X.; Guo, L.X.; Luo, C.T.; Zhou, X.; Yu, C. A survey of target detection and recognition methods in underwater turbid areas. *Appl. Sci.* **2022**, *12*, 4898. [CrossRef]
7. Chen, W.; Wei, Q.Y.; Zhang, J.F.; Guo, B.Y. Target detection of underwater vehicle based on RetinaNet. *Comput. Eng. Des.* **2022**, *43*, 2959–2967.
8. Qiang, W.; He, Y.Y.; Guo, Y.J.; Li, B.Q.; Hi, L.J. Exploring Underwater Target Detection Algorithm Based on Improved SSD. *J. Northwestern Polytech. Univ.* **2020**, *38*, 747–754. [CrossRef]
9. Deng, Q.; Lin, M.X. Marine Organism Detection Algorithm Based on Improved SSD. *Comput. Technol. Dev.* **2022**, *32*, 51–56.
10. Wang, X.H.; Zhu, Y.G.; Li, D.Y.; Zhang, G. Underwater Target Detection Based on Reinforcement Learning and Ant Colony Optimization. *J. Ocean Univ. China* **2022**, *21*, 323–330. [CrossRef]
11. Guo, T.T.; Song, Y.Z.; Kong, Z.J.; Lim, E.; López-Benítez, M.; Ma, F.; Yu, L.M. Underwater Target Detection and Localization with Feature Map and CNN-Based Classification. In Proceedings of the 2022 4th IEEE International Conference on Advances in Computer Technology, Information Science and Communications (CTISC), Suzhou, China, 22–24 April 2022; pp. 1–8.
12. Zeng, L.C.; Sun, B.; Zhu, D.Q. Underwater target detection based on Faster R-CNN and adversarial occlusion network. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104190. [CrossRef]
13. Li, J.F.; Zhu, Y.W.; Chen, M.X.; Wang, Y.L.; Zhou, Z.Q. Research on Underwater Small Target Detection Algorithm Based on Improved YOLOv3. In Proceedings of the 2022 16th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 21–24 October 2022; pp. 76–80.
14. Guo, T.X.; Wei, Y.H.; Shao, H.; Ma, B.Y. Research on Underwater Target Detection Method Based on Improved MSRCP and YOLOv3. In Proceedings of the 2021 IEEE International Conference on Mechatronics and Automation (ICMA), Takamatsu, Japan, 8–11 August 2021; pp. 1158–1163.
15. Zhang, M.H.; Xu, S.B.; Song, W.; He, Q.; Wei, Q.M. Lightweight Underwater Object Detection Based on YOLOv4 and Multi-Scale Attentional Feature Fusion. *Remote Sens.* **2021**, *13*, 4706. [CrossRef]
16. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
17. Chen, L.Y.; Zheng, M.C.; Duan, S.Q.; Luo, W.L.; Yao, L.G. Underwater Target Recognition Based on Improved YOLOv4 Neural Network. *Electronics* **2021**, *10*, 1634. [CrossRef]

18. Li, S.S.; Li, C.; Yang, Y.; Zhang, Q.; Wang, Y.H.; Guo, Z.Y. Underwater scallop recognition algorithm using improved YOLOv5. *Aquac. Eng.* **2022**, *98*, 102273. [CrossRef]
19. Li, Y.; Bai, X.Y.; Xia, C.L. An Improved YOLOV5 Based on Triplet Attention and Prediction Head Optimization for Marine Organism Detection on Underwater Mobile Platforms. *J. Mar. Sci. Eng.* **2022**, *10*, 1230. [CrossRef]
20. Li, J.Y.; Liu, C.N.; Lu, X.C.; Wu, B.L. CME-YOLOv5: An Efficient Object Detection Network for Densely Spaced Fish and Small Targets. *Water* **2022**, *14*, 2412. [CrossRef]
21. Cao, J.R.; Zhuang, Y.; Wang, M.; Han, F.T.; Zheng, X.H.; Gao, H. ECA-based YOLOv5 Underwater Fish Target Detection. *Comput. Syst. Appl.* **2023**, *32*, 1–8.
22. Lei, F.; Tang, F.F.; Li, S.H. Underwater Target Detection Algorithm Based on Improved YOLOv5. *J. Mar. Sci. Eng.* **2022**, *10*, 310. [CrossRef]
23. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *Pattern Anal. Mach. Intell. IEEE Trans.* **2015**, *37*, 1904–1916. [CrossRef]
24. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
25. Rao, Y.M.; Zhao, W.L.; Tang, Y.S.; Zhou, J.; Lim, S.-N.; Lu, J.W. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *arXiv* **2022**, arXiv:2207.14284.
26. Bert, D.B.; Xu, J.; Tinne, T.; Luc, V.G. Dynamic fifilter networks. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1–14.
27. Chen, Y.P.; Dai, X.Y.; Liu, M.C.; Chen, D.D.; Yuan, L.; Liu, Z.C. Dynamic convolution: Attention over convolution kernels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11030–11039.
28. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, P. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
30. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2020**, *80*, 8091–8126. [CrossRef]
31. Aszemi, N.M.; Dominic, P.D.D. Hyperparameter optimization in convolutional neural network using genetic algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 269–278. [CrossRef]
32. Bochinski, E.; Senst, T.; Sikora, T. Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms. In Proceedings of the 2017 IEEE international conference on image processing (ICIP), Beijing, China, 17−20 September 2017; pp. 3924–3928.
33. Du, W.S.; Zhu, Y.J.; Li, S.S.; Liu, P. Spikelets detection of table grape before thinning based on improved YOLOV5s and Kmeans under the complex environment. *Comput. Electron. Agric.* **2022**, *203*, 107432. [CrossRef]
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.