Coordinate Attention and Transformer Neck-Based Marine Organism Detection

Xiangjun Kong[1], Ning Wang[2(✉)], Tingkai Chen[1], and Yanzheng Chen[2]

[1] School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China
[2] School of Marine Engineering, Dalian Maritime University, Dalian 116026, China
n.wang.dmu.cn@gmail.com

Abstract. Marine organism detection is crucial for the intelligent construction of open-sea farm. Suffering from low-contrast, color-deviation and detail-blurry underwater environment, a coordinate attention and transformer neck-based benthonic organism detection (CATNBOD) scheme has been devised. Main contributions are as follows: 1) The coordinate attention (CA) module is designed in the feature extraction network to obtain meaningful features, such that the small-scale benthonic organisms can be accurately detected. 2) To efficiently address the challenge derived from intra- and inter-class occlusions of benthonic organism, the rotation window-based swin transformer (ST) module is devised in the neck structure. Combining with CA and ST modules contributes to the proposed CATNBOD scheme. The effectiveness and superiority have been sufficiently demonstrated on publicly available UDD dataset.

Keywords: Marine organism detection · Coordinate attention · Swin transformer · Model optimization

1 Introduction

With the gradual regional saturation of offshore mariculture capacity, water quality deteriorated and aquaculture quality decreased. In order to improve the quality of mariculture, aquaculture enterprises are gradually turning their eyes to the far sea. Open-sea farm refers to the planned and purposeful marine stocking of marine resources by using the natural marine ecological environment, such as sea cucumbers, sea urchins and scallops. Currently, marine ranching has entered the era of fine management, which requires information such as the health status, individual size and density of marine organisms.The above information needs to be collected manually, which is low efficiency, high cost and dangerous. In order to effectively avoid dangerous and heavy works, intelligent marine robots

This work is supported by the National Natural Science Foundation of China (Grant 52271306), Innovative Research Foundation of Ship General Performance (Grant 31422120), and the Cultivation Program for the Excellent Doctoral Dissertation of Dalian Maritime University (Grant 2022YBPY004).

H. R. Karimi and N. Wang (Eds.): S-Cube 2022, LNICST 487, pp. 33–48, 2023.
https://doi.org/10.1007/978-3-031-34899-0_3

with biological status monitoring system have been gradually put into use [1–3]. Note that the underwater object detection is the most critical technique in above-mentioned system [4,5]. Due to the scattering and absorption of light in the water, underwater image usually presents color distortion, degradation blur, and dim light [6], which makes high-precision detection and recognition become challenging. In addition, marine organisms resemble degraded underwater background and possess small-size characteristic, which makes that rich hierarchical features can hardly be efficiently extracted. In this context, benthonic organism detection still faces severe challenges.

1.1 Related Works

Many scholars have conducted extensive research on underwater target detection techniques, which can be divided into two categories: traditional machine learning and deep learning. By virtue of sliding window technique [7], the traditional target detection algorithm firstly selects the region of interest. Subsequently, feature extraction is performed for each region of interest by using feature extractors, such as HOG [8], SURF [9], SIFT [10], *etc.* Finally, the extracted features are classified to determine whether the window contains object or not by deploying SVM technique [11,12]. Specifically, Ravanbakhsh et al. [13] used contour features for automatic fish detection. Note that the foregoing extracted features are poorly scalable in complex underwater environments. However, the color and structure of marine organisms are similar to seafloor background, which dramatically limits the feature extraction of manual extractors.

Compared with traditional machine methods, deep learning-based target detection algorithms have more powerful feature extraction capabilities. Deep learning-based target detection algorithms can be divided into two categories: region-proposal and regression-based target detection algorithms [14]. The former is also known as two-stage target detection algorithm, such as R-CNN [15], Faster R-CNN [16], and Mask R-CNN [17], which firstly extracts the feature from proposed regions of interest. Subsequently, the classifier is used to perform regression task. Besides, regression-based target detection algorithms, such as SSD [18], YOLOv3 [19], RetinaNet [20], *etc*, work in an end-to-end manner. To be specific, [21] designed an improved feature pyramid-based cucumber detection network. [22] proposed an one-stage CNN detector-based benthonic organism detection scheme. [23]proposed a data augmentation-based marine organism detection and recognition algorithm. Inspired by the human visual system, attention mechanisms have been used for various target detection tasks [24–26]. [27] added the SE attention mechanism to the deep convolutional layer for the aim of improving the detection accuracy of small fish, crabs, shrimps, and starfish. [28]proposed a residual block possessing channel attention mechanism to extract multi-scale effective features for underwater creature detection. [29]explored a visual attention for marine organism detection.

Many scholars mainly focus on the small target of marine organisms and the lightweight of detection model, and propose solutions such as feature fusion, prior anchor frame and data enhancement. However, most of them ignore the multi-target aggregation caused by the unique social property of marine organisms.

1.2 Contributions

The above methods have shown significant improvements in integrating attention mechanisms in the backbone feature extraction network and in target recognition, but not to the extent of considering that marine organisms such as sea urchins and sea cucumbers have swarming properties and there are a large number of aggregation and occlusion situations, and in addition, marine benthonic organism targets occupy only a small part of the image, and most of the image is background information [30], and in deep convolutional neural networks, a large number of image background information convolution iterations will accumulate a large amount of redundant invalid information and swamp the target information, these phenomena exacerbate the difficulty of detection, thus failing to obtain satisfactory detection performance on marine targets. In this paper, to overcome the above challenges, we propose a new lightweight marine life detection model based on deep learning.

Our contributions are as follows:

- We integrate coordinate attention (CA) into the backbone network, which helps the network find regions of interest in images with large areas and improves the feature extraction capability of the model.
- We integrate swin transformer (ST) into YOLOv5 neck structure, which can precisely locate marine organisms in high-density scenes;
- Finally, by integrating CA, ST and YOLOv5, a CATNBOD scheme is established, which has been experimentally demonstrated that the improved network detection accuracy can reach 72.7% (mAP), which exceeds its baseline detection accuracy of 66.1% (mAP), significantly improving the accuracy and robustness of marine organism detection and outperforming other general target detection models. It provides a new solution for the marine organism detection.

2 CATNBOD Network Design

In this section, we detail the proposed CATNBOD network architecture design, including CNN-based backbone network, coordinate attention module, and swin transformer, as shown in Fig. 1.

2.1 Model Backbone Network

Considering the tight computational resources underwater, the accuracy and speed required for marine organism detection, this work uses YOLOv5, a one-stage detection model with fewer parameters and excellent detection performance,

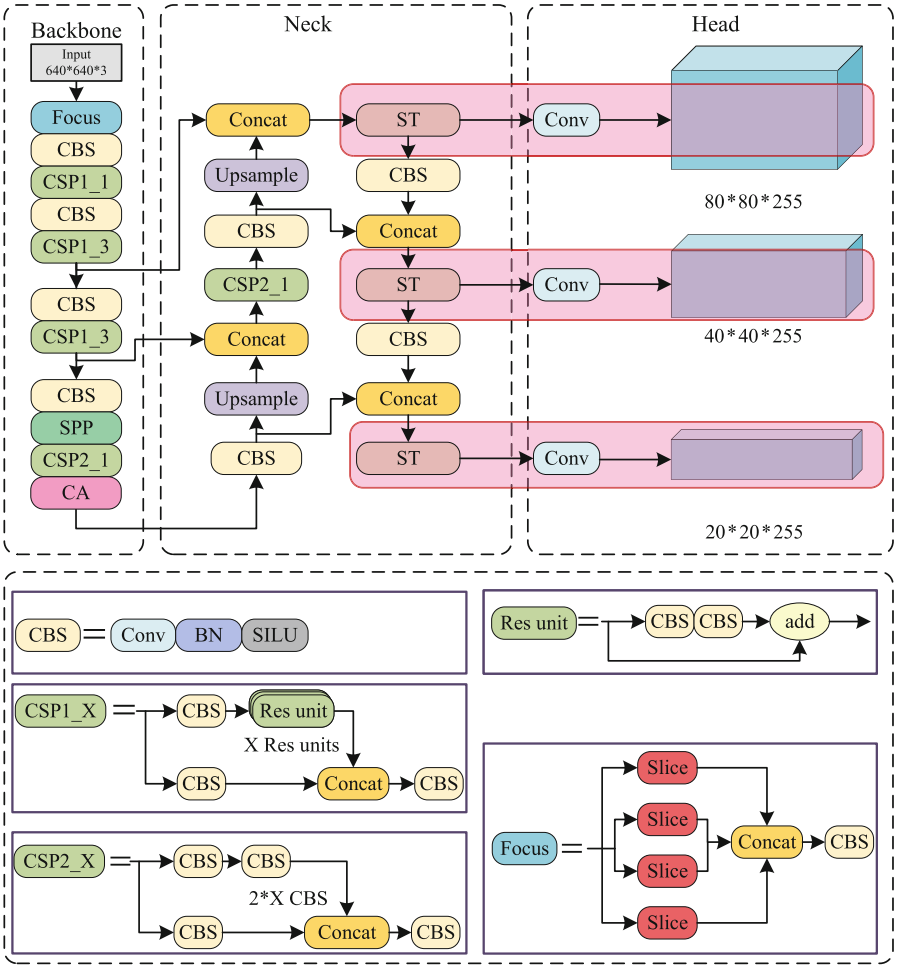

Fig. 1. Network structure for CATNBOD.

as the benchmark model. Depending on the length and width of the backbone network, it is divided into four versions, YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. We use YOLOv5s, which has the smallest parameters and the fastest inference speed, to embed into the proposed CATNBOD marine organism detection model. The backbone consists of Focus, Conv BN SiLU (CBS), Cross Stage Part (CSP) and Spatial Pyramid Pool (SPP) modules. The detailed architecture of these modules is shown in Fig. 1. The Focus role is to slice the image before it enters the backbone to expand the output space by a factor of four, and the

original three channels become twelve channels with no information loss, reducing the amount of operations. The CSP module, inspired by CSPNet [31], halves the number of channels by performing a separate convolution operation to allow the model to learn more distinguishing features. To achieve FeatherMap-level fusion of local and global features, the SPP module is in-serted at the end of the backbone network.

2.2 Coordinate Attention Fusion Module

The information of marine organisms in underwater images is easily obscured by redundant background information in convolutional iterations, which affects the accuracy of underwater biological target detection and recognition. Existing studies have shown that incorporating attention mechanisms into convolutional neural network models can lead to relatively significant performance improvements, but traditional attention mechanisms applied to lightweight networks significantly lag behind deep networks, and the resulting computational overhead is unaffordable for light-weight networks. The popular attention mechanisms include SE (Squeeze and Excitation) [32], BAM (Bottleneck Attention Module) and CBAM (Convolutional Block Attention Module) [33]. Among them, SE only considers internal channel information and ignores location information and spatial structure, which are important for vision tasks, and are crucial for generating spatially selective attention maps. BAM and CBAM introduce local location information by global pooling on channels, but can-not capture long-range dependencies on feature maps.

Coordinate Attention(CA) is a new efficient attention mechanism, the principle of which is shown in Fig. 2. We apply the CA module at the end of the backbone network because the feature mapping resolution at the end of the backbone network is low. Using CA on low-resolution feature maps can reduce the expensive computational overhead. The first step embeds the location information into the channel attention, allowing the lightweight network to obtain information about a larger area, reducing the number of parameters of the attention module while avoiding excessive computational overhead. In the coordinate information embedding process, a specific input $\boldsymbol{X} = [\boldsymbol{x^1}, \boldsymbol{x^2}, \cdots, \boldsymbol{x^c}] \in \mathbb{R}^{c \times w \times h}$ is given and each channel in the horizontal direction is encoded by a mean pooling layer of size $(H, 1)$ to obtain a perceptual feature map in the horizontal direction. As the XAvgPool part of the CA structure diagram, it is the result $z_c^h(h)$ obtained from the output of the $c-th$ channel of width h. The vertical perceptual feature map is obtained by encoding each channel in the vertical direction through an average pooling layer of size $(1, W)$. As the YAvgPool part in the CA structure diagram. $z_c^h(w)$ is the result obtained by matching the output of the $c-th$ channel of width w.

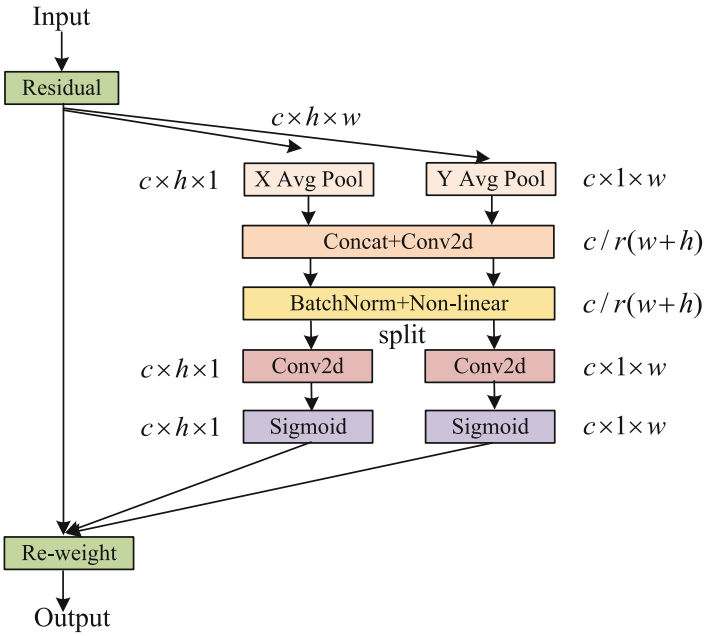

Fig. 2. Coordinate Attention structure schematic.

The resulting operation saves the position information. The principle equations are shown in Eqs. (1) and (2):

$$z_c^h(h) = \frac{1}{W} \sum_i x_c(h, i) \tag{1}$$

$$z_c^h(w) = \frac{1}{H} \sum_i x_c(j, i) \tag{2}$$

In the second step, the vertical and horizontal perceptual feature maps obtained by pooling are cascaded using Concat. Two feature maps $\boldsymbol{z}^h$ and $\boldsymbol{z}^w$ are obtained.

In the third step, on this basis, a convolutional transform function with a convolutional kernel size of 1 is used to transform the information operation to obtain the spatial information in two to encode the intermediate feature mapping law. The transformation formula is shown in Eq. (3):

$$\boldsymbol{f} = \delta(F_1([\boldsymbol{z}^h, \boldsymbol{z}^w])) \tag{3}$$

where:$\boldsymbol{f} \in \mathbb{R}^{c/r \times (H+W)}$ is the intermediate feature map of spatial information in the vertical and horizontal directions, δ is the activation function, F_1 is the convolutional transform function, $[\boldsymbol{z}^h, \boldsymbol{z}^w]$ operation is the two feature map splicing operation.

In the fourth step, two 1×1 convolutional transform functions F_h and F_w are used to transform the two tensors $\boldsymbol{f}^h$ and $\boldsymbol{f}^w$, respectively, into a tensor with the same number of channels as the output. The transformation equations are shown in Eqs. (4) and (5):

$$g^h = \sigma(F_h(\boldsymbol{f}^h)) \tag{4}$$

$$g^w = \sigma(F_w(\boldsymbol{f}^w)) \tag{5}$$

where σ is the sigmoid function, $f^h \in \mathbb{R}^{c/r \times H}$ and $f^w \in \mathbb{R}^{c/r \times W}$, and F^h and F^w are the convolutional transform functions.

In the fifth step, the outputs g^h and g^w are extended as the attention weight assignment values, which can help the network to focus its resources more on channel effective information and spatial effective information, respectively. Combining the above equation, the final extended output equation can be obtained as shown in Eq. (6):

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{6}$$

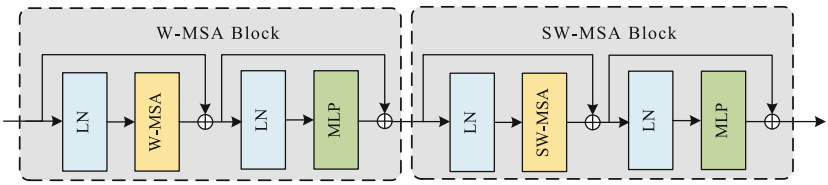

Fig. 3. The swin transformer architecture.

2.3 Swin Transformer Module

Due to the swarming property of marine organisms, this leads to occlusion between objects and thus requires high detector performance. Inspired by swin transformer [34] and TPH-YOLOv5 [35], we replace some convolution blocks and CSP bottleneck blocks in the original version of YOLOv5 with swin transformer, which is compared with the original bottleneck in CSPDarknet53 blocks in CSPDarknet53, we believe that swin transformer can capture global information and rich contextual information. We only apply the swin transformer block in the YOLOv5 neck part. The swin transformer module introduced in this paper consists of two consecutive structures, as shown in Fig. 3. W-MSA block and SW-MSA block are multi-head self attention modules with regular and shifted windowing configurations, respectively.

3 Experiments and Analysis

3.1 Experimental Environment and Data Set

The experimental environment is windows operating system, Intel Xeon Silver 4210 (CPU), 16 GB random access memory (RAM), GTX 2080Ti (GPU), and deep learning framework is Pytorch 1.6.5, software environment is CUDA 10.1, CUDNN 7.6.5, and python 3.8. The model is optimized using SGD (stochastic gradient descent) method was used for optimization. The specific settings of the network training hyperparameters are 40 for the training period, 10 for the batch size, 0.01 for the initial learning rate, 0.0005 for the weight decay, and 0.9 for the SGD momentum.

Optical images are mainly acquired by means of underwater robots carrying cameras when exploring fishery resources in marine pastures. Therefore, this paper uses the open source dataset UDD from Dalian University of Technology to verify the reliability of the algorithm in this paper. The dataset contains a total of 2227 images of three types of underwater biological targets, sea cucumber, seaurchin and scallop in multiple scenes, of which 1827 are used for training and 400 for testing.

3.2 Model Evaluation Metrics

The experiments use frame per second (FPS), recall(R), average precision (AP) and mean average precision (mAP) metrics to objectively evaluate the performance of the network. We use size of moudle, number of parameters, and Floating Point Operations (FLOPs) to evaluate model complexity.

The FPS metric is used to evaluate the detection speed of the network, and the larger the value, the faster the network detection speed.

The AP metric is used to evaluate the detection accuracy of a single category, which can be calculated as follows:

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{7}$$

where P represents the precision rate, which refers to the probability that all detected targets are correctly detected, and R represents the recall rate, which refers to the probability that all true positive samples are correctly detected, and the calculation can be expressed as follows:

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

where TP represents the number of positive samples detected correctly, FP represents the number of samples detected as positive but actually negative,

and FN represents the number of samples detected as negative but actually positive.

Using mAP to evaluate the combined detection accuracy of the model, the calculation can be expressed as follows:

$$mAP = \sum_{i=1}^{N} AP_i / N \tag{10}$$

where N represents the number of all categories and AP_i represents the average precision of category i. The model weight file size can visually reflect the complexity of the model, and the larger the weight file is, the more complex the model is.

The marine organism detection model is trained using the official pre-training weights to obtain new training weights. The model weight file size can visually reflect the complexity of the model, and a larger weight file indicates a more complex model.

The number of parameters is a common evaluation metric for network complexi-ty, and for the convolutional layer, the calculation of the number of parameters on the fully connected and batch normalized layers follows the following formula.

Denote the number of parameters for the convolutional layer as M^{conv}, which is calculated as follows:

$$M^{conv} = K_1 \times K_2 \times C_{in}^{conv} \times C_{out}^{conv} + C_{out}^{conv} \tag{11}$$

where K_1 and K_2 denote the size of the convolution kernel, C_{in}^{conv} and C_{out}^{conv} denote the number of input feature map channels and the number of output feature map channels of the convolution layer, respectively.

The number of parameters of the fully connected layer is M^{fc}, which is calculated as follows:

$$M^{fc} = C_{in}^{fc} \times C_{out}^{fc} + C_{out}^{fc} \tag{12}$$

where C_{in}^{fc} and C_{out}^{fc} denote the number of input features and output features of the fully connected layer, respectively.

Denote the number of parameters of the batch normalization layer as M^{bn}, which is calculated as follows:

$$M^{bn} = 2 \times C^{bn} \tag{13}$$

where C^{bn} denotes the number of channels in the batch normalization layer.

Denoting the total number of parameters in the model as M, which is calculated as follows:

$$M = \sum\nolimits_{i=1}^{C} M_i^{conv} + \sum\nolimits_{j=1}^{F} M_j^{fc} + \sum\nolimits_{k=1}^{B} M_k^{bn} \tag{14}$$

where C, F, and B denote the number of layers in the model for the convolutional, fully connected, and batch normalized layers, respectively.

FLOPs is a common metric for evaluating the computation of the model and measuring the performance of CNNs, and the calculation of FLOPs on convolutional, fully connected and batch normalized layers follows the following formula.

The FLOPs of the convolutional layer are F^{conv} and are calculated as follows:

$$F^{conv} = B \times H_{out} \times W_{out} \times (2 \times C_{in}^{conv} \times {K_1}^{conv} \times {K_2}^{conv} + 1) \times C_{out}^{conv} \quad (15)$$

where B denotes the size of the batch size, H_{out} and W_{out} denote the height and width of the output feature map of the convolutional layer, C_{in}^{conv} and C_{out}^{conv} denote the number of input and output feature map channels of the convolutional layer, respectively, and K_1^{conv} and K_2^{conv} denote the size of the convolutional kernel.

The FLOPs of the fully connected layer are denoted as F^{fc} and are calculated as follows:

$$F^{fc} = B \times (2 \times C_{in}^{fc} + 1) \times C_{out}^{fc} \quad (16)$$

where C_{in}^{fc} and C_{out}^{fc} denote the number of input features and output features of the fully connected layer, respectively.

The FLOPs of the pooling layer are F^{pl}, which are calculated as follows:

$$F^{pl} = B \times (C^{pl} \times K_1 \times K_2 \times H_{out} \times W_{out}) \quad (17)$$

where C^{pl} denotes the number of channels of the pooling layer, K_1^{pl} and K_2^{pl} denote the size of the convolution kernel of the pooling layer, and H_{out} and W_{out} denote the height and width of the output feature map of the pooling layer, respectively.

The FLOPs of the batch normalization layer are denoted as F^{bn}, which are calculated as follows:

$$F^{bn} = B \times (2 \times C^{bn}) \quad (18)$$

where C^{bn} denotes the number of channels in the batch normalization layer, which is multiplied by 2, because the batch normalization is subject to multiplication and addition operations.

Denoting the total floating-point operations of the model as G, we have:

$$G = B \times (\sum_{i=1}^{C} F_i^{conv} + \sum_{j=1}^{F} F_j^{fc} + \sum_{l=1}^{P} F_l^{pl} + \sum_{k=1}^{N} F_k^{bn} \quad (19)$$

where C, F, P, and N denote the number of layers of convolutional, fully connected, pooling, and batch normalization layers in the model, respectively.

3.3 Ablation Experiments

To evaluate the effectiveness of the improved algorithm in this paper, three aspects of detection accuracy, detection speed and model complexity were compared and analyzed with YOLOv5s algorithm, and the experimental results are shown in Table 1 and Table 2.

Table 1. Ablation studies within different modules.

Model	AP(%) seacucumber	AP(%) seaurchin	AP(%) scallop	mAP(%) @.5	mAP(%) @[.5, .95]	R	FPS/s
YOLOv5s	59.6	88.9	50.0	66.1	27.3	61.5	63
YOLOv5s+CA	56.7	90.1	60.5	69.1	28.4	64.7	59
YOLOv5s+ST	53.6	90.2	56.2	66.7	27.9	62.7	61
CATNBOD	57.9	90.5	69.7	72.7	30.0	70.7	58

Table 2. Comparison with YOLOv5s algorithm model complexity M/G denotes million/billion ($10^6/10^9$)respectively.

Model	Bacbone nework	Size/MB	Params/M	FLOPs/G
YOLOv5s	CSPDarknet53	14.1	7.1	16.3
YOLOv5s+CA	CSPDarknet53	15.5	8.0	17.5
YOLOv5s+ST	CSPDarknet53	14.2	7.3	16.7
CATNBOD	CSPDarknet53	16.0	8.2	17.7

As shown in Table 1 and Table 2, after YOLOv5s+CA introduces coordinate attention module, the model size, number of parameters, and floating point computation increase by 1.4 MB, 0.9 M, and 1.2 G, respectively, and the model inference speed decreases slightly, but the average detection accuracy increases by 3%. It shows that coordinate attention increases the model complexity and reduces the network inference speed, but this attention mechanism can capture the location information and channel relationship, suppress the non-essential feature information, make the network pay more attention to the target feature information, improve the quality of the feature map extracted by the network, and significantly improve the detection accuracy of the model.

It is worth noting that usually the shells of scallops are mostly grayish-brown in color similar to the background of sand and silt on the seabed. In the baseline model, the scallop detection rate of 50% is the lowest detection success rate among the three types of targets, but the addition of CA attention mechanism significantly improves the detection accuracy to 60.5%, indicating that the CA module plays a great role.

YOLOv5s+ST enhances the detection capability of dense small targets by introducing swin transformer module in the detection head, with 1.2%improvement in seaurchin and recall rate and 0.7% improvement in average detection accuracy. The model size, number of parameters, and floating point computation only increase by 0.1MB, 0.1M, and 0.4G, achieving detection performance improvement with little impact on model complexity.

CATNBOD integrated coordinate attention module and swin transformer module, compared with baseline, the detection accuracy of seaurchin improved by 1.6%, scallop detection accuracy improved by 19.7%, sea cucumber detection

accuracy decreased slightly, and the average detection accuracy improved by 6.6%. The scallop targets account for only 1.9% of the total number of targets, and are usually small in size, fuzzy in feature information and similar to the background, so the detection accuracy is low. This paper shows that the combination of different improvement strategies can significantly improve the feature extraction ability of the network, which can alleviate the limitation of the lack of data samples to some extent, and improve the detection ability of small and fuzzy targets.

Table 3. Comparison with other state-of-the-art detectors.

Model	Backbone	AP(%) seacucumber	AP(%) seaurchin	AP(%) scallop	mAP(%) @.5	mAP(%) @[.5, .95]
Faster R-CNN	ResNet50	66.8	76.2	51.0	64.7	26.8
YOLOv3	DarkNet53	62.9	89.6	49.4	67.3	28.7
YOLOv5	CSPDarknet53	59.6	88.9	50.0	66.1	27.3
CATNBOD	CSPDarknet53	57.9	90.5	69.7	72.7	30.0

To demonstrate the superiority of our proposed CATNBOD scheme, a comprehensive comparison with typical two-stage and one-stage detection methods including Faster R-CNN, YOLOv3, and YOLOv5 is performed. To ensure a fair comparison, the above schemes are thoroughly pre-trained on the COCO dataset, followed by fine-tuned training on the UDD dataset. Therefore, comparing the results listed in Table 3, from which we can observe that the proposed CATNBOD scheme gives the best results in the detection of sea urchins and scallops. While the Faster R-CNN scheme using Region Proposal Network has the best detection accuracy for sea cucumbers. the YOLOv3 algorithm also achieves good results for detecting sea cucumbers, but performs poorly for scallops. In the absence of CA and ST techniques, YOLOv5 cannot achieve the same detection accuracy as the proposed scheme. Our proposed algorithm achieves the best results in both mAP@.5 and mAP@[.5, .95]. In addition, the proposed algorithm can obtain better real-time performance by using the lightweight CSPDarknet53 backbone network.

3.4 Image Detection Effect Analysis

In order to fully demonstrate the superiority of the proposed CATNBOD scheme in terms of practical validation, we selected four more representative marine benthonic organism scenes from the validation set for comparison experiments, and Fig. 4 shows the sample detection results by adding different method modules.

It can be clearly seen from columns (a), (c) and (d) that more marine organisms can be detected accurately after adding CA module, which proves the necessity of adding CA attention to marine life detection scenes. After adding the ST

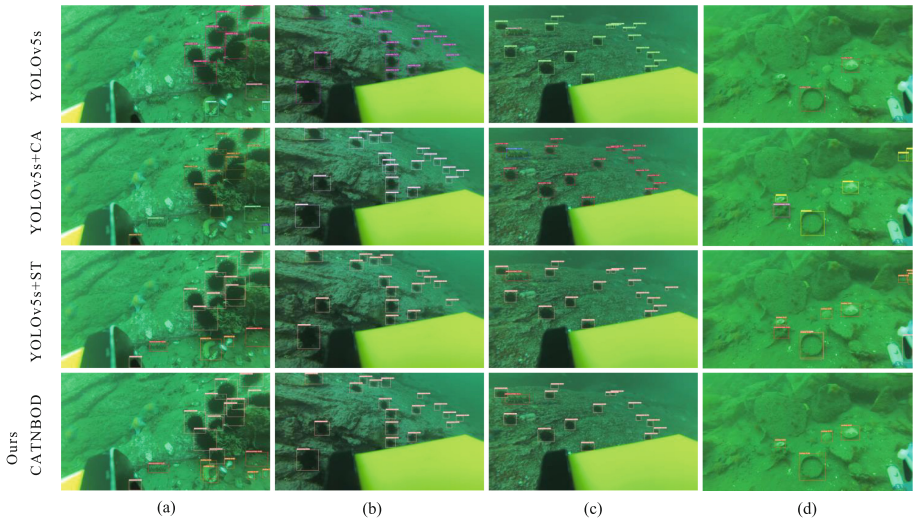

Fig. 4. Comparison of real-world detection results.

module, it can be seen from the complex and dense scenes in columns (a) and (d) that YOLOv5s+ST helps detect small targets of sea cucumbers and scallops in the rock crevices and improves the detection performance of marine organisms. However, when the environment is too complex, the YOLOv5s+CA and YOLOv5s+ST models still suffer from false detection and missed detection problems. (d) Columns of water plants were incorrectly identified as sea cucumbers, rocks were identified as scallops, and multiple scallops were not detected.

From the comparison Fig. 4 in column (a), it can be seen that YOLOv5s missed one sea cucumber, one seaurchin and multiple scallops between the stone gaps, and our proposed CATNBOD achieved accurate and complete identification of all sea cucumbers and sea urchins. From the comparison graphs in columns (b) and (c), it can be seen that our algorithm is able to identify more minute sea cucumber targets. From column (d), it can be seen that our algorithm can identify more scallop targets and only one false detection. In conclusion, the analysis of the detection results shows that the detection and recognition effect of the improved model in this paper is much better than the original YOLOv5s detection effect, and the false detection rate and the missed detection rate of the target organisms are reduced.

4 Conclusions

For the unique group living properties of marine organisms to produce target aggregation, obscuration phenomenon, thus leading to the problem of missed detection, wrong detection, as well as small target information and the background of the seabed similar to the problem of difficult to extract features. A CATNBOD framework is proposed for marine organism detection. Specifically,

the coordinate attention mechanism is introduced in the backbone network to capture location information and channel relationships to improve the feature extraction capability of the network; the swin transformer neck-based is constructed to improve the detection capability of the network for small targets; we conduct comprehensive experiments to illustrate the effectiveness of the proposed algorithm.

In our future research, we will continue to improve the performance of marine organism target detection in turbid environment, and integrate the improved algorithm in intelligent underwater robots for practical marine organism detection tasks.

References

1. Wang, N., Wang, Y., Er, M.J.: Review on deep learning techniques for marine object recognition: architectures and algorithms. Control. Eng. Pract. **118**(3), 104458 (2022)
2. Wang, N., Qian, C., Sun, J., Liu, Y.: Adaptive robust finite-time trajectory tracking control of fully actuated marine surface vehicles. IEEE Trans. Cybern. **24**(4), 1454–1462 (2016)
3. Wang, N., Er, M.J.: Direct adaptive fuzzy tracking control of marine vehicles with fully unknown parametric dynamics and uncertainties. IEEE Trans. Control Syst. Technol. **24**(5), 1845–1852 (2016)
4. Yeh, C., et al.: Lightweight deep neural network for joint learning of underwater object detection and color conversion. IEEE Trans. Neural Netw. Learn. Syst. **99**, 1–15 (2021)
5. Wang, Y., et al.: Real-time underwater onboard vision sensing system for robotic gripping. IEEE Trans. Instrum. Meas. **70**, 1–11 (2020)
6. Han, M., Lyu, Z., Qiu, T., Xu, M.: A review on intelligence dehazing and color restoration for underwater images. IEEE Trans. Syst. Man Cybern. Syst. **50**(5), 1820–1832 (2020)
7. Forsyth, D.: Object detection with discriminatively trained part-based models. Computer **47**(02), 6–7 (2016)
8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 886–893, San Diego, CA, USA (2005)
9. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. Lect. Notes Comput. Sci. **3951**, 404–417 (2006)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
11. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Netw. **17**(1), 113–126 (2004)
12. Wang, N., Er, M.J.: Self-constructing adaptive robust fuzzy neural tracking control of surface vehicles with uncertainties and unknown disturbances. IEEE Trans. Control Syst. Technol. **23**(3), 991–1002 (2014)
13. Villon, S.; Chaumont, M.; Subsol, G.; Villéger, S.; Claverie, T.; Mouillot, D.: Coral reef fish detection and recognition in underwatervideos by supervised machine learning: Comparison between Deep Learning and HOG+ SVM methods. In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Lecce, Italy, pp. 160–171 (2016)

14. Serban, A., Poll, E., Visser, J.: Adversarial examples on object recognition: a comprehensive survey. ACM Comput. Surv. **53**(3), 1–38 (2020)
15. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
16. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2015)
17. He, K., Gkioxari, G., Dollár, P.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
18. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
19. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv 2018. arXiv:1804.02767
20. Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
21. Peng, F., Miao, Z., Li, F., Li, Z.: S-FPN: a shortcut feature pyramid network for sea cucumber detection in underwater images. Expert Syst. Appl. **182**, 115306 (2021)
22. Chen, T., Wang, N., Wang, R., Zhao, H., Zhang, G.: One-stage CNN detector based benthonic organisms detection with limited training dataset. Neural Netw. **144**, 247–259 (2021)
23. Huang, H., Zhou, H., Yang, X.: Faster R-CNN for marine organisms detection and recognition using data augmentation. Neurocomputing **337**, 372–384 (2019)
24. Wang, N., Karimi, H.R., Li, H., Su, S.-F.: Accurate trajectory tracking of disturbed surface vehicles: a finite-time control approach. IEEE/ASME Trans. Mechatron. **24**(3), 1064–1074 (2019)
25. Wang, N., Er, M.J., Sun, J., Liu, Y.: Adaptive robust online constructive fuzzy control of a complex surface vehicle system. IEEE Trans. Cybern. **46**(7), 1511–1523 (2016)
26. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
27. Wei, X., Yu, L., Tian, S., Feng, P., Ning, X.: Underwater target detection with an attention mechanism and improved scale. Multimed. Tools Appl. **80**(25), 33747–33761 (2021). https://doi.org/10.1007/s11042-021-11230-2
28. Li, A., Yu, L., Tian, S.: Underwater biological detection based on YOLOv4 combined with channel attention. J. Mar. Sci. Eng. **10**(4), 469 (2022)
29. Shi, Z., et al.: Detecting marine organisms via joint attention-relation learning for marine video surveillance. IEEE J. Ocean. Eng. **47**(4), 959–974 (2022)
30. Xu, F., Wang, H., Peng, J., Fu, X.: Scale-aware feature pyramid architecture for marine object detection. Neural Comput. Appl. **33**(8), 3637–3653 (2021)
31. Wang, C., Liao, H., Wu, Y., Chen, P., Hsieh, J., Yeh, I.: CSPNet: A new backbone that can enhance learning capability of CNN. In: roceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 390–391 (2020)
32. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)

33. Woo, S., Park, J., Lee, J., Kweom, I.: CBAM: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision, pp. 3–19 (2018)
34. Liu, Z., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
35. Zhu, X., Lyu, S., Wang, X., Zhao, Q.: TPH-YOLOv5: improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2778–2788 (2021)