Computing and Informatics, Vol. 42, 2023, 210–233, doi: 10.31577/cai_2023_1_210

MDM-YOLO: RESEARCH ON OBJECT DETECTION ALGORITHM BASED ON IMPROVED YOLOV4 FOR MARINE ORGANISMS

Sha Li, Yong Liu, Shuang Wu

College of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
e-mail: liuyongqust@163.com

Shoujiang J. Zhang

Haier Overseas Electric Industry Co
Qingdao, China

Abstract. Vision-based underwater object detection technology is a hot topic of current research. In order to address the issues of low accuracy and high missed rate of marine life detection, an object detection algorithm called MDM-YOLO (Marine Detection Model with YOLO) for marine organisms based on improved YOLOv4 is proposed. To improve the network's capacity for feature extraction, a multi-branch architecture CSBM is integrated into the backbone. Based on this, the feature fusion structure introduces shuffle attention to reinforce the focus on important information. The experimental results demonstrate that the MDM-YOLO algorithm increases the mean average precision (mAP) by 2.31 % compared to the YOLOv4 algorithm on the Underwater Robot Picking Contest (URPC) dataset. Moreover, on the RSOD dataset and PASCAL VOC dataset, MDM-YOLO obtained an mAP of 87.54 % and 86.87 %, respectively. According to these advancements, the MDM-YOLO model is more suitable for the identification of items on the seafloor.

Keywords: Object detection, deep learning, YOLOv4, marine organisms

Mathematics Subject Classification 2010: 68U10

1 INTRODUCTION

There is no denying the ocean's significance and its role as a catalyst for social and economic development [1]. For reliable sea life identification and location, underwater robots are essential for ocean exploration [2]. Robots have been developed in a variety of fields [3, 4, 5], but underwater robots struggle with issues including distorted underwater images and weak computational capabilities [6]. As a result, undersea creature detection remains a challenging task.

Traditional object detection generally starts with a sliding window algorithm that adopts sliding windows of various widths for the initial target localization. After that, a local binary pattern and directed gradient histogram are introduced to extract the characteristics of candidate regions. Eventually, the collected features were categorized using the support vector machine or adaboost algorithms. However, untargeted region selection and ineffective artificial feature extraction are the primary difficulties that classical object detection systems face [7]. These issues make these methods slow, inaccurate, and time-consuming, and they prevent them from being accurate and real-time enough to identify marine life.

Due to recent developments in deep learning technology, the convolutional neural network has significantly improved object detection algorithms. The convolutional neural network avoids the laborious process of obtaining features by using the original image as input and extracting features from a large number of samples through a nonlinear model. Deep learning-based object detectors are mainly separated into two-stage and one-stage depending on whether there are region proposals generated. Two-stage detectors are mainly represented by the regional convolutional neural network (RCNN) [8], Fast R-CNN [9], and Faster R-CNN [10]. They first generate candidate object bounding boxes for the image and then carry out a secondary correction to them for the following classification and bounding-box regression tasks. The one-stage detectors are mainly dominated by the YOLO [11] and SSD [12], which directly process images to generate the outcomes without the region proposal step [13, 14]. Compared with the previous type of detectors, the one-stage detectors simplify the detection process and have superior detection efficiency. Due to the high complexity of the anchor box setup, some researchers have also proposed some anchor-free detection algorithms. One example of this tendency is CornerNet [15] which creates a bounding box by combining the target's top left and lower right corners. By directly anticipating the nine representative points, RepPoint [16] determines the location of the closest bounding box surrounding these points.

The development of these potent detection algorithms has also contributed to the study of marine species. The improvement of the underwater detecting system has drawn more attention from researchers. For example, using Fast RCNN, Li et al. [17] developed a fish species detector that outperformed RCNN in term of accuracy and speed. The YOLOv2 model was proposed by Xia et al. [18] to identify sea cucumbers, and it was demonstrated how significance training samples and detection model optimization are to increase accuracy. Han et al. [19] integrated max-RGB and grayscale gradient to ameliorate image quality and up-

graded deep convolutional neural networks to detect sea cucumbers and sea urchins. Song et al. [20] used an MSRCR image enhancement algorithm paired with Mask RCNN, exhibiting more than 90 % mAP on a small sample underwater dataset, yet its speed is too slow to be practical. Fan et al.[21] constructed six prediction layers to create a new multi-scale feature to identify the new underwater dataset UWD, while the advance was inapparent in terms of precision. Chen et al. [22] set up multiple high-resolution and semantic-rich feature maps combined to form Sample-Weighted hyPEr network (SWIPENet) for small underwater objects but the accuracy was not adequate. Zeng et al. [23] designed an adversarial occlusion network (AON) and Faster RCNN against each other to learn to correctly classify intercepted targets, making the undersea detection model more robust. Adding some shortcut structures, Fang et al. [24] created the S-FPN by incorporating a few shortcut structures in order to lessen the typical message loss and produce a high-precision detecting impact on sea cucumbers. Liu and Wang [25] reduced the omission ratio of small and dense marine benthos by embedding the kernel adaptive selection unit in the backbone network of Faster RCNN, whereas it was not effective at recognizing multi-scale objectives. Hu et al. [26] embedded dense_unit in YOLOv4 and applied high-resolution feature maps to realize the detection of underwater dense tiny particles. Zhang et al. [27] introduced AFFM attention for feature fusion in YOLOv4 to obtain richer semantic information and increase the accuracy of underwater object detection.

According to above analysis, there are two troubles with underwater object detection. Because of problems such as uneven light distribution and relatively large water waves, the quality of underwater collected datasets is generally inferior, which may make feature extraction more difficult. Secondly, the small volume and easy aggregation of many halobios can easily create missed detection. Based on the above discussion, we first take advantage of image enhance algorithm to process images. After this, some ameliorations are made in the efficient one-stage algorithm YOLOv4 [28] for marine creature object detection. The contributions and benefits of our method are as follows:

1. Using the Multi-Scale Retinex with Color Restoration (MSRCR) [29] algorithm to increase the visibility and recognition of the image.
2. Designing the CSBM structure in CSPdarknet53 to heighten the backbone network feature extraction capability.
3. Introducing the shuffle attention (SA) [30] mechanism to reinforce the network's competency to capture representations of the objects of interest.

2 RELATED METHODOLOGY AND WORK

2.1 YOLOv4 Detection Algorithms

YOLOv4 is one of the most widely used one-stage object detection algorithms. The backbone network for feature extraction, the neck for feature fusion, and the

detecting head for classification regression make up the majority of YOLOv4. The backbone network CSPDarknet53 is improved from Darknet53 in YOLOv3 [31], and Darknet53 is mainly composed of five residual network structures. The cross-stage partial network (CSPNet) [32] concept, which divides the residual block stacking into two halves, is adopted by YOLOv4. Half of the channels in the feature map remain the original residual block stacking, and with minimal processing, the other half is connected to the end directly like a residual edge. By lowering the number of channels engaged in residual block stacking and increasing the gradient path through chunking, the CSPNet effectively lowers the computational cost. To prevent various layers from picking up redundant gradient information, these two components are then aggregated. This split-and-merge structure can improve the backbone's feature extraction while also addressing the gradient disappearance issue brought on by deeper deep neural networks. The CSPDarknet53 backbone structure is shown in Figure 1.

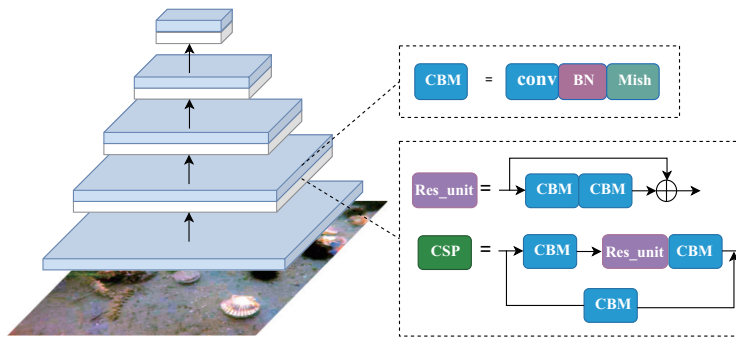


Figure 1. CSPDarknet53 structure

The Spatial Pyramid Pooling (SPP) [33] module and Path aggregation network (PANet) [34] structure are located in the neck, where the SPP module is positioned following CSPdarknet53's deepest feature layer. It is comprised of four max-pooling layers with pooling kernels of 1×1, 5×5, 9×9, and 13×13 in four distinct sizes. The SPP module performs max-pooling and feature fusion by diverse pooling kernels, which decreases information consumption, gathers local receptive field information, and retains the most important contextual properties.

The PANet structure is an amelioration of FPN [35], which has a bottom-up, lateral connection, and top-down network structure. FPN can construct multi-scale feature maps with rich high-level semantic information and handle detection objects at different scales flexibly. Nevertheless, in the process of downsampling, the higher-level feature maps are more likely to lose information. To alleviate this problem, PANet as in Figure 2 adds deep to shallow feature fusion paths to strengthen the flow between multiple layers of information and improve the problem of missing information at the shallow level of FPN.

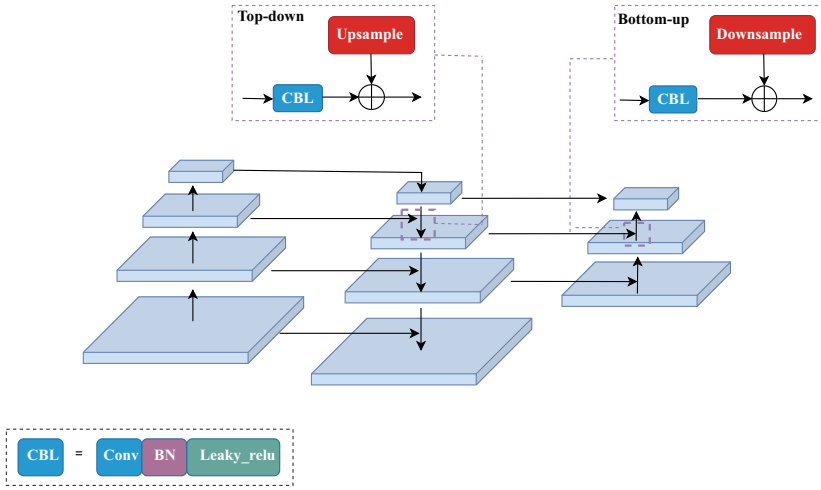


Figure 2. PANet structure

The YOLOv4 network input is generally of specific lengths such as $608 \times 608 \times 3$, $416 \times 416 \times 3$. Since after five downsamplings in the network, three feature maps of different scale sizes are formed, and all of them are multiples of 32. The processed pictures are inputted into the CSPDarknet53 backbone network for feature extraction, generating feature maps of three sizes 52×52, 26×26, and 13×13 after SPP and PANet. The detection head maps the feature maps of these three sizes back to the original image, divides the image into grids of the corresponding size, and realizes the detection of small, medium and large targets separately. Compared with the two-stage algorithms, YOLOv4 has great advantages in accuracy and speed, while saving a lot of computational resources and training time costs.

CBM and CBL in Figures 1 and 2 are convolutional layers in the network incorporating batch normalization (BN) [36] and activation functions. Not using the Leaky relu function that most network apply, YOLOv4 replaces the activation function of the backbone network with the Mish function, which has the characteristics of low cost, smooth curve, non-monotonic, lower bound without upper bound, and has better generalization ability and results of effective optimization capability.

2.2 Improved YOLOv4 Algorithm for Marine Organisms Detection

An effective solution to the problem of blurred and distorted underwater picture imaging and low biological detection accuracy is to improve the feature extraction capability of the network. Method adopted in this paper embeds CSBM module in the CSPdarknet53 network, which can strengthen the convolution network's ability to extract features without deepening the network hierarchy. Furthermore, an ultra-lightweight shuffle attention mechanism is added in PANet to enhance the focus of

aim information, further improve the detection accuracy and reduce the missed detection of marine organisms.

2.2.1 The Feature Extraction Backbone Network Based on CSBM Module

In most detection network architectures, there are three main functional modules: a backbone network for initial feature extraction, a specific network for deep feature extraction, and a head network for detection. Of these, the backbone network is used to extract some low-level general features such as color, shape, and texture. As a benchmark network for many high-level tasks, its performance largely determines part of the upper limit of that network. To reduce the loss of feature information and optimize the detection performance, it is necessary to improve the feature extraction ability in the backbone network. Consequently, inspired by RepVGG [37], a multi-branch structure is designed into CSPNet to reconstitute a new module, as shown in Figure 3, called cross-stage multiple-branching block (CSMB). This multi-branching convolutional block contains two convolutions of different sizes, 1×1 and 3×3, replacing the original 3×3 convolution in the CSPNet to fuse different learned knowledge and retain richer spatial information.

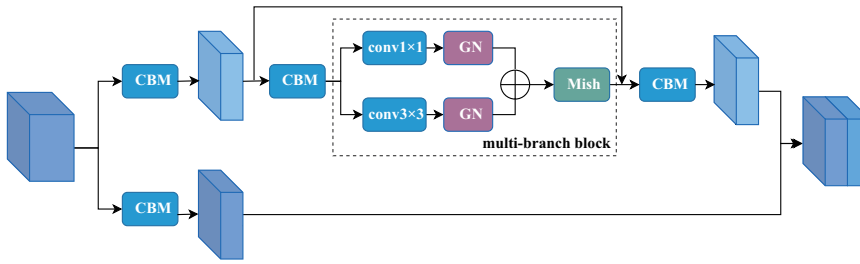


Figure 3. CSBM structure

The CSBM module can be divided into two parts in the inference stage:

1. fusing the 1×1 convolutional and 3×3 convolutional layers with the group normalization (GN) [38] layer separately;
2. superimposing the fused convolutional layers to get a new 3×3 convolutional layer.

The specific process can be described as follows:

Convolution layer: Assume the parameters of a convolution layer with C input channels and D outputchannels. The input feature map is $I \in R^{C \times H \times W}$ and the output feature map is $O \in R^{C \times H' \times W'}$. The convolution kernel is a four-dimensional vector $F \in R^{D \times C \times K \times K}$ of size $K \times K$, and an optional bias $b \in R^D$. The equation of the convolution process can be expressed as Equation (1):

$$O = I * F + REP(b), \tag{1}$$

where "*" denotes the convolution operation, and formulate the bias-adding as replicating the bias b into $REP(b) \in R^{D\times H'\times W'}$. The value at (h, w) on the i^{th} output channel is given by Equation (2):

$$O_{i,h,w} = \sum_{c=1}^{C}\sum_{u=1}^{K}\sum_{v=1}^{K} F_{i,c,u,v}X(c,h,w)_{u,v} + b_i, \tag{2}$$

where $X(c, h, w) \in R^{K\times K}$ is the sliding window on the c^{th} channel of corresponding to the position (h, w) on o. Such a correspondence is determined by the padding and stride. From the above equation, it is easy to infer the linearity of the convolution, which contains the homogeneity and additivity [39]. The equations can be show in Equations (3), (4):

$$I * (pF) = p(I * F), \forall p \in R, \tag{3}$$

$$I * F^{(1)} + I * F^{(2)} = I * \left(F^{(1)} + F^{(2)}\right). \tag{4}$$

The additivity holds only if the two convolutions have the same configurations (e.g., number of channels, kernel size, stride, padding, etc.).

Convolution-GN layer: A BN layer usually is configured after convolution for channel normalization and linear scaling. Batch normalization, on the other hand, has a small-batch-size issue since it normalizes the activation using mini-batch statistics during training but overall statistics during inference. This might cause network performance to suffer by changing the distribution of the data during testing. To combat the buildup of BN estimate bias, a number of batch-free normalizations [38, 40, 41] are presented and proven as a solution [42]. Layer normalization (LN) [40], which uniformizes the layer input within the neurons for each training sample, is one such technique. Let j be the channel position, μ_j and σ_j are the cumulative channel-wise mean and standard deviation, γ_j and β_j be the learned scaling factors and bias term, respectively, the output channel j becomes Equation (5):

$$O_{j,:,:} = \left((I * F)_{j,:,:} - \mu_j\right)\gamma_j/\sigma_j) + \beta_j. \tag{5}$$

GN is a further generalization of LN and is more flexible, enabling it to achieve good performance on visual tasks limited to small-batch-size training (e.g., object detection and segmentation). The homogeneity of the convolution allows the GN layer to be fused into the convolution when inferred. The convolution kernel F' and b' bias of the fused channel can be represented as Equation (6):

$$F'_{j,:,:,:} \leftarrow F_{j,:,:,:}\gamma_j/\sigma_j F_{j,:,:,:}, b'_j \leftarrow -\mu_j\gamma_j/\sigma_j + \beta_j. \tag{6}$$

Branch Fusion: According to the additivity of convolution, convert all the convolutions into a 3×3 convolution. This process can be easily implemented by first zero-padding the 1×1 kernels to 3×3. In this way, all branch convolutions are added to be a new 3×3 convolution. Noted that the additivity of the convolution

here requires the same configuration, so the stride values of both convolutions are set to 1. The padding of the 1×1 convolution should be one pixel less than that of the 3×3 convolution, thus setting the padding $= 0$ of the former and the padding $= 1$ of the latter.

The network training becomes simpler and more efficient since more gradient flow paths are generated by adding convolution branch. However, a major deficiency of the multi-branch structure is that it is not friendly to memory and inference speed. It may not be effective for YOLOv4 models that are sufficiently complex themselves if an over-complicated multi-branch structure is used. That is why we do not design more branches to improve network performance.

2.2.2 Feature Fusion Based on SA Mechanism

The feature fusion phase is a further extraction of the generic traits obtained from the backbone network, which transforms into the features needed in the detection task. YOLOv4 applies the PANet structure for separate detection of targets at different scales in this stage, which allows more small objects traits to be focused on. However, the feature maps are inclined to information loss during multiple dimensionality reduction. Therefore, the SA mechanism is embeded into PANet to enhance the focus of the network on the key targets and reduce the wastage of target information.

The SA is an efficient shuffle attention mechanism, which effectively integrates the channel attention mechanism and spatial attention mechanism. The interaction between spatial locations dynamically depends on their respective features, enhancing the momentous trait information of the feature map in channel and space, making the network training more capable of capturing the principal target features for learning. The SA mechanism consists of four main components: feature grouping, channel attention mechanism, spatial attention mechanism, and feature aggregation. The architecture is shown in Figure 4.

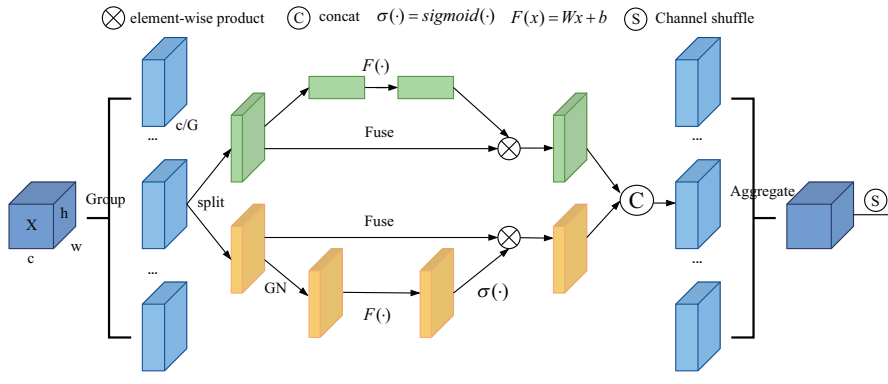


Figure 4. SANet structure

Feature Grouping: It is mainly used to group the input feature maps. Suppose the input feature map $X \in R^{C\times H\times W}$, where C, H, W indicate the channel number, spatial height, and width, respectively. SA is first split into groups along the channel dimension: $X = [X_1, \ldots, X_G], R^{C/G\times H\times W}$. Each group of features is split into two branches along the channel dimension: X_{k1},$X_{k2} \in R^{C/2G\times H\times W}$, which generate different importance coefficients focusing on the target class and location information through the channel and spatial attention mechanisms, respectively.

Channel Attention: In terms of channel attention implementation, the more classical one is the squeeze-and-excitation (SE) [43] module. However, it will bring too many parameters, which is not conducive to the design of lightweight attention mechanisms. To be as lightweight as possible, the global information firstly is emdedded by using global averaging pooling (GAP) to generate channel-wise statistics as $S \in R^{C/2G\times 1\times 1}$, which can be calculated by shrinking X_{k1} through the spatial dimensions $H \times W$. It can be shown in Equation (7):

$$s = F_{gp}\left(X_{k1}\right) = \sum_{i=1}^{H}\sum_{j=1}^{W} X_{k1}(i,j)/H \times W. \tag{7}$$

In addition, precise and adaptive weight changes are achieved by a simple gating mechanism with sigmoid activation functions. The final output of the channel attention can be obtained by Equation (8):

$$W'_{k1} = \sigma(F(s)) \cdot X_{k1} = \sigma\left(W_1 s + b_1\right) \cdot X_{k1}, \tag{8}$$

where $W_1 \in R^{C/2G\times 1\times 1}$ and $b_1 \in R^{C/2G\times 1\times 1}$ are learnable parameters for scaling and shifting s.

Spatial Attention: Unlike channel attention, spatial attention focuses on the location information of the target. In practice, we use GN over X_{k2} to get spatial-wise statistics, and then $\hat{X}_{k2}$ representation is enhanced by F(x). The final output of the channel attention can be got by Equation (9):

$$X'_{k2} = \sigma\left(W_2 \cdot GN\left(X_{k2}\right) + b_2\right) \cdot X_{k2}, \tag{9}$$

where $W_2 \in R^{C/2G\times 1\times 1}$ and $b_2 \in R^{C/2G\times 1\times 1}$ denote learnable parameters. After that, the two branches are fused by a simple concatenation to obtain $X'_k = [X'_{k1}, X'_{k2}] \in R^{C/G\times H\times WV}$, making the number of channels and the number of inputs the same.

Aggregation: It tends to prevent the flow of information between channels and weaken the expressiveness of the model with too many groups. The channel shuffle operation is adopted to disrupt the order of groups and then connect them to achieve the flow of information along the channel dimension across groups.

The idea behind the attention mechanism is to give the feature map weights, and those weight values typically correspond to how important the feature information is. The backbone network filters out the majority of useless features, however some do still exist. Richer information can be retained more appropriately since the attention

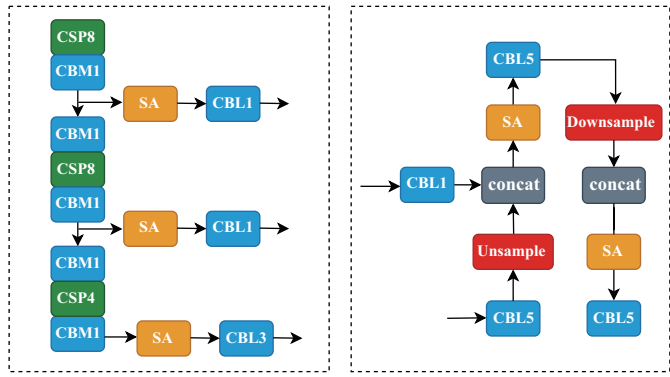


Figure 5. Location embedding structure of SA mechanism

mechanism suppresses the available background information while keeping it. We conducted an experimental investigation to determine whether the SA mechanism is more successful when incorporated before or after feature fusion, as shown in Figure 5, due to the lightweight and "plug-and-play" nature of the SA mechanism. Finally, the SA mechanism is put after feature fusion.

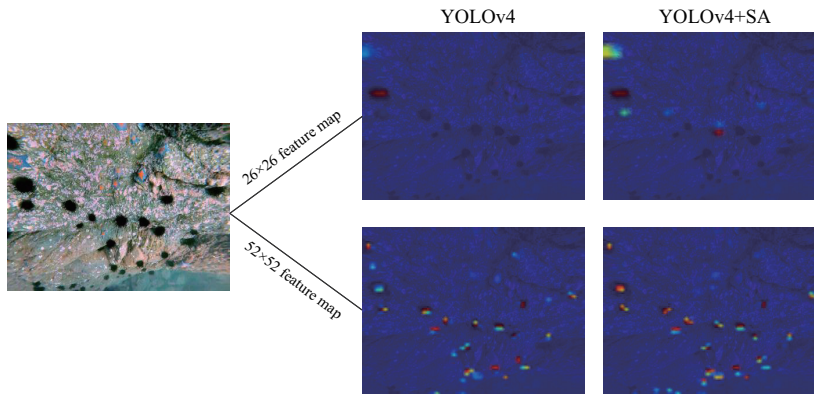


Figure 6. Visualization of feature maps

The visualization of the feature map after adding the SA mechanism is shown in Figure 6, and there is a picture with more small sea urchins to be marked. Considering that the deepest layer feature map generally identifies large-sized targets, the two layers of 26×26 and 52×52 feature maps are visualized and compared. It is evident that there are additional targets in the 26×26 visualization diagram that can be concentrated on following the enhancement. The suppression of starfish misidentification and the enhancement of the attention on the right target are seen

in the 52×52 visualization plot. From this outcome, the SA mechanism is effective in capturing vital pixels and suppressing useless target information. After CSBM and SA strength, the improved overall structure is shown in Figure 7.

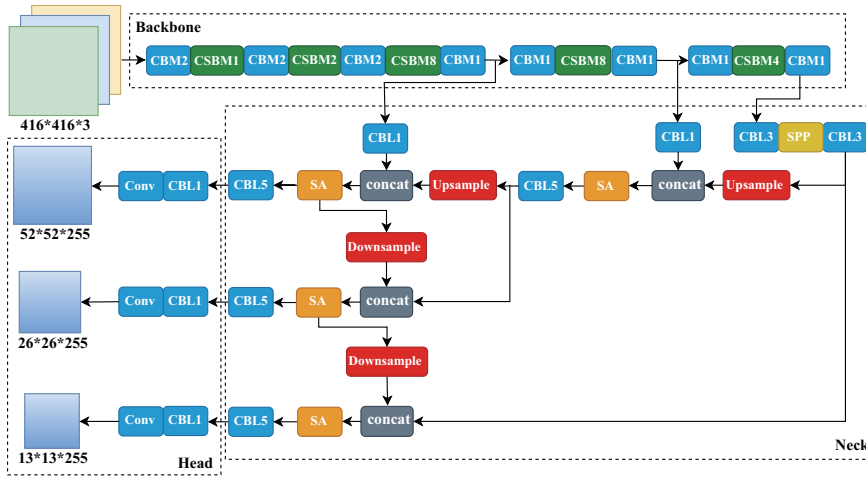


Figure 7. MDM-YOLO structure

3 EXPERIMENTS AND RESULTS

The experimental model training and testing were implemented on a server with a Xeon E5 CPU and NVIDIA GTX3090-24G GPU combined with CUDNN8.2.4 and CUDA11.4, using Python 3.7 and PyTorch deep learning framework.

3.1 Experimental Setup

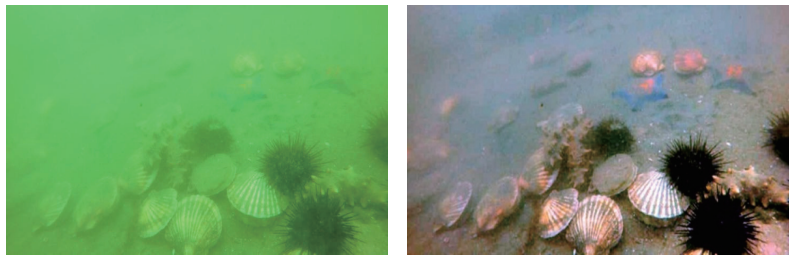
3.1.1 Dataset

In this paper, the official dataset of the 2020 China Underwater Robotics Competition is used in our research, and all images are taken in real underwater environments. The whole dataset includes four categories of marine organisms: holothurian, echinus, scallop, and starfish. The dataset contains 5400 images of underwater creatures in jpg format. The training set and test set are divided in the ratio of 9:1, where 1/10 of the image data in the training set is the validation set. That is, 4 860 images are used for the training set, 486 of them for the validation set, and the rest 540 images are for the test set. The following characteristics exist in this dataset:

1. Owing to the large underwater shaking and limited visibility, the captured image is not clear enough to extract the target for processing;

2. There is a limited amount of data, which makes the network training prone to overfitting;
3. The marine organisms in the data are generally small and obscured, generating more obstacles to capturing accurate object information.

For this reason, the MSRCR algorithm, which is able to raise the brightness of the image overall and improve the local contrast of the image, is selected to deal with the light problem. Overall targets, like starfish and sea cucumbers, become simpler to discern in the fuzzy zone, as seen in Figure 8.



a) Before MSRCR b) After MSRCR

Figure 8. The samples of URPC dataset

Futhermore, the RSOD [44, 45] dataset and the PASCAL VOC [46] dataset are utilized to confirm the model's effectiveness. In the following, we introduce the RSOD and VOC datasets.

RSOD: this is an open dataset for object detection in remote sensing images, annotated by the remote sensing image target detection team of Wuhan University. It includes four types of targets: aircraft, playground, overpass and oil-tank. The dataset includes 4 files, and each file represents one kind of object.

VOC: The PASCAL VOC challenge mainly has the sub-tasks of Object Classification, Object Detection, Object Segmentation, Human Layout, and Action Classification. The current object detection commonly used is the VOC2007 and VOC2012 datasets, which are divided into a total of 4 major categories: vehicle, household, animal, and person, with a total of 20 subcategories. Here, the VOC2007 dataset is selected for training.

3.1.2 Evaluation Metrics

In this experiment, four performance metrics, namely Average Precision (AP), mean Average Precision (mAP), Missing Rate (MR), Parameters (Params), and Floating-point Operations (FLOPs), are used to evaluate the model performance.

1. The AP evaluates the accuracy of image category detection for a single label,

and the calculation formulas are in following Equations (10), (11), (12):

$$P = TP/(TP + FP), \tag{10}$$

$$R = TP/(TP + FN), \tag{11}$$

$$AP = \int_0^1 P(R)\,\mathrm{d}r, \tag{12}$$

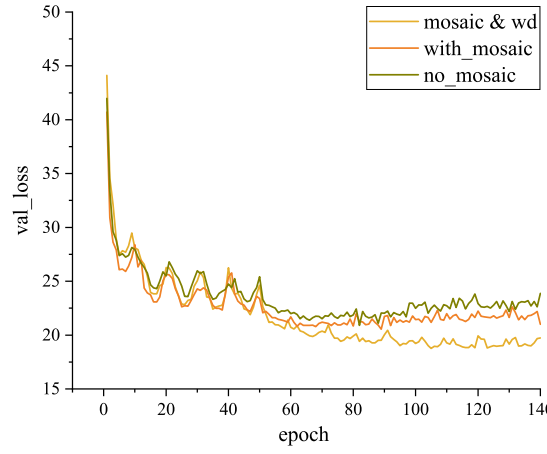
where, P is the precision rate, R is the recall rate, True Positive (TP) indicates the number of positive samples included in the predicted positive samples, False Positive (FP) indicates the number of negative samples included in the predicted positive samples, and False Negative (FN) means the number of positive samples included in the predicted negative samples. $P(R)$ is the P-R curve, AP is the area under the P-R curve.

2. The mAP metric is used for the evaluation of multi-label image classification tasks and is an important indicator of the overall detection accuracy of the model in multi-category target detection. The mAP can be calculated by Equation (13):
$$mAP = \textstyle\sum_{n=1}^{n} AP/n, \tag{13}$$
where n denotes the number of detected dataset classes.
3. The MR is the missed detection rate, which is used to evaluate the model detection performance. The calculation equation is Equation (14):
$$MR = FN/(FN + TP). \tag{14}$$
4. The FLOPs denote the amount of computation and are used to measure the complexity of the algorithm model.

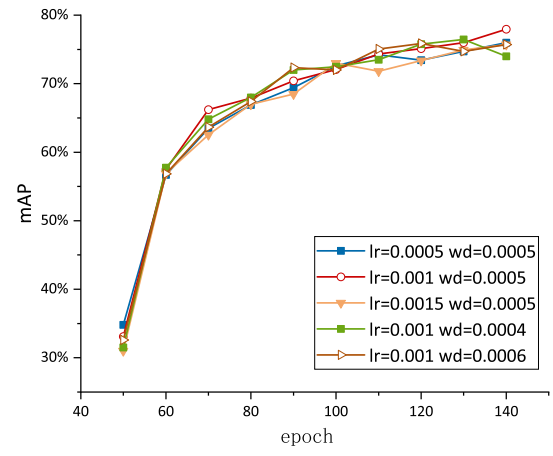
3.2 Experimental Results

3.2.1 Parameters Settings

Several experiments are done to tune the parameters to select more suitable ones for model training. Overfitting is easy due to the small amount of data in our dataset. We introduce the mosaic algorithm to perform the data enhancement operation and add the regularized weight decay (wd) to the optimizer to relieve the model overfitting problem. Figure 9 presents that with the addition of data enhancement and weight decay, the validation set loss is subsequently reduced in the late training period as show in Figure 9 a). Since our model may not converge if the learning rate changes too much, which will result in failure to train properly, the learning rate is fine-tuned as shown in Figure 9 b). Although the change is not evident, it is still seen that the accuracy change is more stable and higher when lr = 0.001 and wd = 0.0005.



a)



b)

Figure 9. Parameter comparison experiment

3.2.2 Results

Additionally, the MDM-YOLO algorithm in this work, the basic YOLOv4, the two-stage detector Faster R-CNN, the widely used one-stage detector SSD, the YOLOv3 and YOLOv5 are all tested on the URPC dataset. As shown in Table 1, our method surpasses the other five state-of-the-art methods. It can be seen that MDM-YOLO has increased AP over YOLOv4 in all categories, whereas Faster RCNN and SSD both have slightly lower accuracy.

Ablation experiments are adopted to demonstrate the effectiveness of various methods. There are three groups of experiments in Table 2, group 01 without any

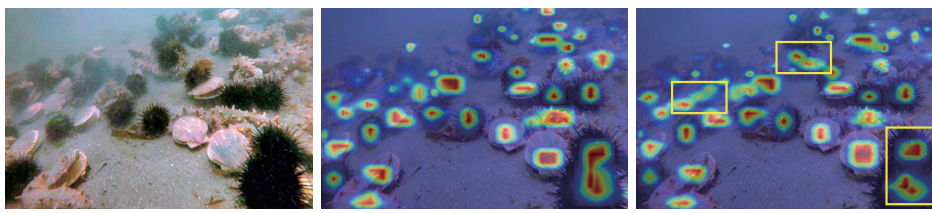
Model	AP (%)				mAP (%)
	Starfish	Holothurian	Echinus	Scallop	
Faster RCNN	77.72	60.37	78.53	48.00	66.16
SSD	72.65	59.59	74.37	46.62	63.31
YOLOv3	80.97	58.48	85.65	68.87	73.49
YOLOv4	80.52	64.41	86.72	70.87	75.63
YOLOv5	80.30	67.04	86.72	76.22	77.57
MDM-YOLO	82.22	70.34	87.90	71.30	77.94

Table 1. Experimental comparison of different algorithm models

treatment, group 02 introducing CSBM structure, and group 03 embedding SA mechanism in PANet based on group 02 to be the MDM-YOLO model. Where "+" expresses that the method is used. After adding the CSBM module, the mAP is improved by 1.29 % compared with the original one, and after superimposing the SA mechanism, the mAP is improved by another 1.02 %. Ultimately, the total mAP is improved by 2.31 % compared with YOLOv4. Additionally, it is clear that they only contribute a relatively tiny number of additional parameters and calculations. Figure 10 show the visualization results under different methods. Although the influence on the heat map is not immediately apparent, it can still be used to recognize the benefits of the upgraded network. The detection rate of each category except starfish in Figure 11 has decreased. From these comparisons, it is observed that MDM-YOLO presents better results both in terms of accuracy and in terms of miss detection rate.

	CSBM	SA	mAP (%)	Params (M)	FLOPs (G)
01			75.63	69.13	29.98
02	+		76.92	71.24	30.98
03	+	+	77.94	71.24	30.98

Table 2. Comparison of ablation experiments



a) Original image b) YOLOv4 c) MDM-YOLO

Figure 10. The visualization of different methods

Convolutional branches can be added to the network to improve performance, but doing so it will result in an increase in complexity that must be taken into

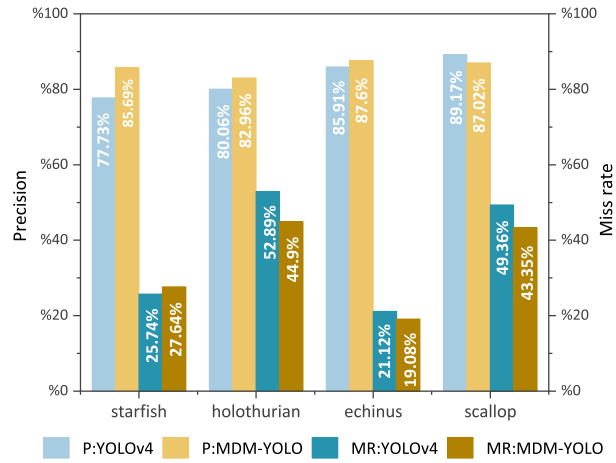


Figure 11. The precision and Missed Rate of different methods

account. We must decide whether it is worthwhile to forego some computational efficiency in order to achieve better detection results. Additionally, it needs to be confirmed that GN actually produces beneficial consequences. As shown in Table 2, experiment 01's outcomes were average despite using identity, 1×1, and 3×3 as multi-branch structures to train the model. Experiment 02 does away with the identification branch, which results in considerably better performance and less complexity – a worthwhile trade-off for enhancing network performance. In Experiment 03, we introduced GN blocking distribution bias and mAP improved by 0.46 %, which is an optimization worth trying without adding any burden.

CSBM	Starfish		Holothurian		Echinus		Scallop		mAP (%)
	AP (%)	MR (%)	AP (%)	MR (%)	AP (%)	MR (%)	AP (%)	MR (%)	
01	81.35	34.23	63.24	55.52	87.09	24.45	69.58	57.51	75.31
02	80.63	35.12	64.77	46.23	87.37	22.50	73.01	46.35	76.46
03	82.52	23.19	64.58	46.75	86.42	21.44	74.18	37.05	76.92

Table 3. CSBM comparison experiment

In order to select the correct SA mechanism placement, two sets of experiments are conducted, and this experiment is performed on the original YOLOv4 algorithm. To reduce the computation, the SA mechanism is set to 64, i.e., the feature map is divided into 64 groups before attention weighting. Experiment 01 SA mechanism is placed before feature fusion, and Experiment 02 is placed after feature fusion. As shown in Table 4, Experiment 01 puts the SA mechanism before feature fusion and it does not improve the performance, and the AP value of each category is the lowest,

and the mAP decreases by 0.48 % in comparison with YOLOv4. Experiment 02 puts the SA mechanism after feature fusion with better results, and the mAP improves by 1.05 %.

SA	Starfish		Holothurian		Echinus		Scallop		mAP
	AP (%)	MR (%)	AP (%)	MR (%)	AP (%)	MR (%)	AP (%)	MR (%)	
01	79.74	34.67	62.60	51.49	86.39	21.42	71.86	48.93	75.15
02	83.01	24.26	62.81	56.57	87.40	21.38	73.32	43.86	76. 63

Table 4. SA mechanism comparison experiment

Figure 12 shows the effect of our experiments presented in the picture specifically, where the increased detected objects are circled using yellow circles. After MSRCR enhancement, more sea urchins and starfishes are detected, which reduces the target miss detection to some extent. Adding CSBM, more scallop and starfishe are able to be recognized. Finally, adding the SA mechanism, MDM-YOLO can recognize four more scallops. This fully illustrates that image enhancement is very necessary, and MDM-YOLO performs better than YOLOv4 in the leakage detection problem.

4 DISCUSSION

Today, target detection is used in everything from self-driving cars and identity detection to security and medical applications, and the marine sector is no exception. Underwater image processing has shown great potential for exploring underwater environments, such as automated underwater vehicles (AUVs)-driven applications [27], novel low-cost integrated system prototype for recognizing lifeforms underwater [44], and video-based or image-based underwater object detection. However, the complexity of the marine environment, the attenuation of artificial light sources, and the impact of low-end optical imaging equipment have all led to degradation of imaging distortion, making underwater image targets more difficult to detect. The urgent need to protect marine species and exploit marine resources continue to drive the development of underwater detection methods. Continuously optimizing the performance of these algorithms is a top priority to achieve more effective detection results. The main contribution of this paper is to propose a YOLOv4-based marine organisms target detection algorithm for improving the detection accuracy and reducing the missed detection of marine life in underwater blurred images.

Convolution is an operation to obtain local information, and different convolution kernel sizes are able to capture different ranges of information. The backbone network preserves richer multi-scale semantic information through the superposition of multi-branch convolution, which actively participates in capturing small-scale marine biological aspects. For detection, the target to be recognized is the foreground and everything else is the background, and most of the background information is considered as picture noise. The SA module, which combines channel attention

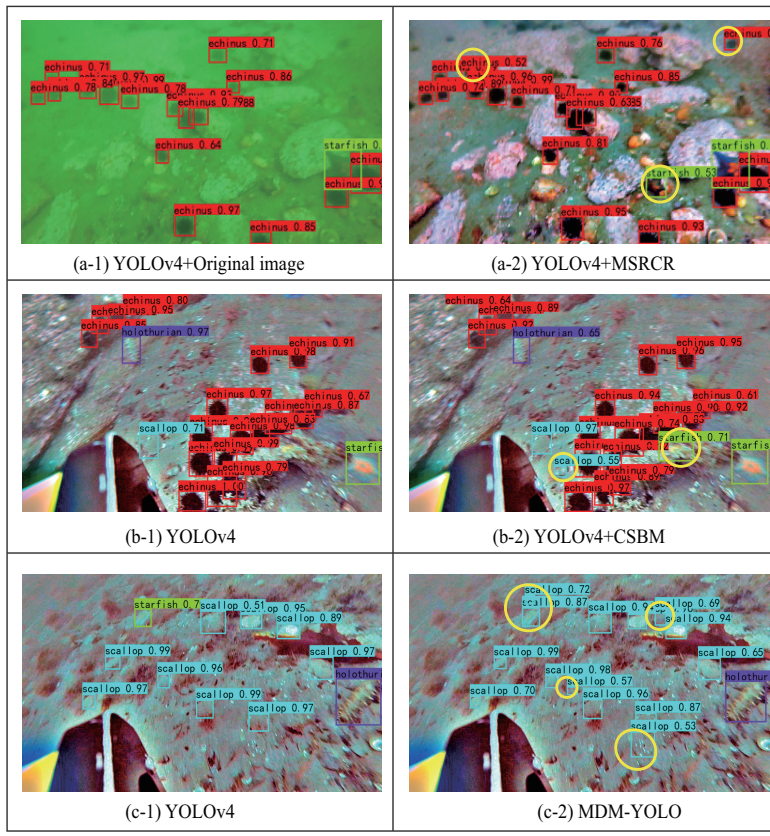


Figure 12. Comparison of detection results in different algorithms

with spatial attention to boost the network's ability to collect important pixels and channels, helps the network become more focused on the region of interest information. This operation is helpful to drive further accuracy improvement of the network.

To demonstrate the broad validity of the MDM-YOLO model, we also conduct experimental training on both the RSOD dataset, which has many small-sized targets, and the PASCAL VOC dataset, which has more large-sized targets.

As in Table 5, we use Params, FLOPs and the mAP value as performance evaluation metrics. It can be seen that our model improves the marine organism

detection accuracy without increasing too much complexity and does not affect the high efficiency of YOLOv4 algorithm detection. The Faster RCNN, a two-stage detector, has a not bad mAP but immense calculating costs, and both SSD and YOLOv3 accuracy and speed are slightly inferior. In addition, MDM-YOLO achieves better performance in both small target dataset RSOD and large target dataset VOC.

Model	Backbone	Params(M)	FLOPs(G)	RSOD	VOC
				mAP (%)	mAP (%)
Faster RCNN	Resnet50	28.47	470.17	86.1	80.36
SSD	VGG	26.15	31.39	73.4	78.55
YOLOv3	Darknet53	61.63	32.83	76.8	81.97
YOLOv4	CSPDaknet53	69.13	29.98	85.9	85.25
MDM-YOLO	CSPDaknet53	71.24	30.98	87.54	86.87

Table 5. Experimental comparison of different datasets

5 CONCLUSION

In this study, an object detection algorithm based on improved YOLOv4 for marine organisms is proposed. The CSBM module is introduced to enhance the feature extraction capability of backbone network and the SA mechanism is embedded to enrich the semantic information of the network so as to improve the detection accuracy of marine organisms. In practice, five performance metrics: Average Precision (AP), mean Average Precision (mAP), Missing Rate (MR), Parameters (Params), and Floating-point Operations (FLOPs) are adopted to evaluate the performance of the improved detection algorithm architecture. The model was able to achieve a 77.94 % mAP on the URPC 2020 dataset. Furthermore, 86.87 % and 87.54 % mAPs were obtained in the public dataset PASCAL VOC and the remote sensing dataset RSOD, respectively. Compared with other detection algorithms, the experimental results indicate that the MDM-YOLO algorithm is more suitable for underwater object detection and is valuable for the study of marine species richness. MDM-YOLO continues to use the strategy of expanding network complexity in order to increase detection accuracy, and we'll keep working to do so in the future. It is worth mentioning that although our research is mainly aimed at the marine field, it can be extended to more fields after corresponding optimization, and has great prospects for development.

Acknowledgements

This project is partially supported by the program of The Institute of Oceanology, Chinese Academy of Sciences "Deep sea biological in situ intelligent recognition system and quantitative analysis system development project" (KEXUE2019GZ04).

REFERENCES

[1] Raphael, A.—Dubinsky, Z.—Iluz, D.—Netanyahu, N. S.: Neural Network Recognition of Marine Benthos and Corals. Diversity, Vol. 12, 2020, No. 1, Art. No. 29, doi: 10.3390/d12010029.

[2] Huang, H.—Tang, Q.—Li, J.—Zhang, W.—Bao, X.—Zhu, H.—Wang, G.: A Review on Underwater Autonomous Environmental Perception and Target Grasp, the Challenge of Robotic Organism Capture. Ocean Engineering, Vol. 195, 2020, Art. No. 106644, doi: 10.1016/j.oceaneng.2019.106644.

[3] Li, C.—Fahmy, A.—Li, S.—Sienz, J.: An Enhanced Robot Massage System in Smart Homes Using Force Sensing and a Dynamic Movement Primitive. Frontiers in Neurorobotics, Vol. 14, 2020, Art. No. 30, doi: 10.3389/fnbot.2020.00030.

[4] Li, C.—Zhu, S.—Sun, Z.—Rogers, J.: BAS Optimized ELM for KUKA iiwa Robot Learning. IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 68, 2020, No. 6, pp. 1987–1991, doi: 10.1109/TCSII.2020.3034771.

[5] Li, C.—Fahmy, A.—Sienz, J.: Development of a Neural Network-Based Control System for the DLR-HIT II Robot Hand Using Leap Motion. IEEE Access, Vol. 7, 2019, pp. 136914–136923, doi: 10.1109/ACCESS.2019.2942648.

[6] Yeh, C. H.—Lin, C. H.—Kang, L. W.—Huang, C. H.—Lin, M. H.—Chang, C. Y.—Wang, C. C.: Lightweight Deep Neural Network for Joint Learning of Underwater Object Detection and Color Conversion. IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, 2022, No. 11, pp. 6129–6143, doi: 10.1109/TNNLS.2021.3072414.

[7] Zhao, Z. Q.—Zheng, P.—Xu, S. t.—Wu, X.: Object Detection with Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, Vol. 30, 2019, No. 11, pp. 3212–3232, doi: 10.1109/TNNLS.2018.2876865.

[8] Girshick, R.—Donahue, J.—Darrell, T.—Malik, J.: Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587, doi: 10.1109/cvpr.2014.81.

[9] Girshick, R.: Fast R-CNN. 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, doi: 10.1109/ICCV.2015.169.

[10] Ren, S.—He, K.—Girshick, R.—Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, 2017, No. 6, pp. 1137–1149, doi: 10.1109/TPAMI.2016.2577031.

[11] Redmon, J.—Divvala, S.—Girshick, R.—Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[12] Liu, W.—Anguelov, D.—Erhan, D.—Szegedy, C.—Reed, S.—Fu, C. Y.—Berg, A. C.: SSD: Single Shot MultiBox Detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (Eds.): Computer Vision – ECCV 2016. Springer, Cham, Lecture Notes in Computer Science, Vol. 9905, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[13] Jiao, L.—Zhang, F.—Liu, F.—Yang, S.—Li, L.—Feng, Z.—Qu, R.: A Survey of Deep Learning-Based Object Detection. IEEE Access, Vol. 7, 2019, pp. 128837–128868, doi: 10.1109/ACCESS.2019.2939201.

[14] Kang, J.—Tariq, S.—Oh, H.—Woo, S. S.: A Survey of Deep Learning-Based Object Detection Methods and Datasets for Overhead Imagery. IEEE Access, Vol. 10, 2022, pp. 20118–20134, doi: 10.1109/ACCESS.2022.3149052.

[15] Law, H.—Deng, J.: CornerNet: Detecting Objects as Paired Keypoints. International Journal of Computer Vision, Vol. 128, 2020, No. 3, pp. 642–656, doi: 10.1007/s11263-019-01204-1.

[16] Yang, Z.—Liu, S.—Hu, H.—Wang, L.—Lin, S.: RepPoints: Point Set Representation for Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9657–9666, doi: 10.1109/ICCV.2019.00975.

[17] Li, X.—Shang, M.—Qin, H.—Chen, L.: Fast Accurate Fish Detection and Recognition of Underwater Images with Fast R-CNN. OCEANS 2015 – MTS/IEEE Washington, IEEE, 2015, pp. 1–5, doi: 10.23919/OCEANS.2015.7404464.

[18] Xia, C.—Fu, L.—Liu, H.—Chen, L.: In Situ Sea Cucumber Detection Based on Deep Learning Approach. 2018 OCEANS – MTS/IEEE Kobe Techno-Oceans (OTO), IEEE, 2018, pp. 1–4, doi: 10.1109/OCEANSKOBE.2018.8559317.

[19] Han, F.—Yao, J.—Zhu, H.—Wang, C.: Underwater Image Processing and Object Detection Based on Deep CNN Method. Journal of Sensors, Vol. 2020, 2020, Art. No. 6707328, doi: 10.1155/2020/6707328.

[20] Song, S.—Zhu, J.—Li, X.—Huang, Q.: Integrate MSRCR and Mask R-CNN to Recognize Underwater Creatures on Small Sample Datasets. IEEE Access, Vol. 8, 2020, pp. 172848–172858, doi: 10.1109/ACCESS.2020.3025617.

[21] Fan, B.—Chen, W.—Cong, Y.—Tian, J.: Dual Refinement Underwater Object Detection Network. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. M. (Eds.): Computer Vision – ECCV 2020. Springer, Cham, Lecture Notes in Computer Science, Vol. 1236, 2020, pp. 275–291, doi: 10.1007/978-3-030-58565-5_17.

[22] Chen, L.—Liu, Z.—Tong, L.—Jiang, Z.—Wang, S.—Dong, J.—Zhou, H.: Underwater Object Detection Using Invert Multi-Class Adaboost with Deep Learning. 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9207506.

[23] Zeng, L.—Sun, B.—Zhu, D.: Underwater Target Detection Based on Faster R-CNN and Adversarial Occlusion Network. Engineering Applications of Artificial Intelligence, Vol. 100, 2021, Art. No. 104190, doi: 10.1016/j.engappai.2021.104190.

[24] Peng, F.—Miao, Z.—Li, F.—Li, Z.: S-FPN: A Shortcut Feature Pyramid Network for Sea Cucumber Detection in Underwater Images. Expert Systems with Applications, Vol. 182, 2021, Art. No. 115306, doi: 10.1016/j.eswa.2021.115306.

[25] Liu, Y.—Wang, S.: A Quantitative Detection Algorithm Based on Improved Faster R-CNN for Marine Benthos. Ecological Informatics, Vol. 61, 2021, Art. No. 101228, doi: 10.1016/j.ecoinf.2021.101228.

[26] Hu, X.—Liu, Y.—Zhao, Z.—Liu, J.—Yang, X.—Sun, C.—Chen, S.—Li, B.—Zhou, C.: Real-Time Detection of Uneaten Feed Pellets in Underwater Images for Aquaculture Using an Improved YOLO-V4 Network. Com-

puters and Electronics in Agriculture, Vol. 185, 2021, Art. No. 106135, doi: 10.1016/j.compag.2021.106135.

[27] Zhang, M.—Xu, S.—Song, W.—He, Q.—Wei, Q.: Lightweight Underwater Object Detection Based on YOLO v4 and Multi-Scale Attentional Feature Fusion. Remote Sensing, Vol. 13, 2021, No. 22, Art. No. 4706, doi: 10.3390/rs13224706.

[28] Bochkovskiy, A.—Wang, C. Y.—Liao, H. Y. M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. 2020, doi: 10.48550/arXiv.2004.10934.

[29] Rahman, Z.—Jobson, D. J.—Woodell, G. A.: Multi-Scale Retinex for Color Image Enhancement. Proceedings of 3rd IEEE International Conference on Image Processing, Vol. 3, 1996, pp. 1003–1006, doi: 10.1109/ICIP.1996.560995.

[30] Zhang, Q. L.—Yang, Y. B.: SA-Net: Shuffle Attention for Deep Convolutional Neural Networks. ICASSP 2021 – 2021 IEEE International Conference on Acoustics, Speech and Signal Processing, 2021, pp. 2235–2239, doi: 10.1109/ICASSP39728.2021.9414568.

[31] Redmon, J.—Farhadi, A.: YOLOv3: An Incremental Improvement. 2018, doi: 10.48550/arXiv.1804.02767.

[32] Wang, C. Y.—Liao, H. Y. M.—Wu, Y. H.—Chen, P. Y.—Hsieh, J. W.—Yeh, I. H.: CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.

[33] He, K.—Zhang, X.—Ren, S.—Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, 2015, No. 9, pp. 1904–1916, doi: 10.1109/TPAMI.2015.2389824.

[34] Liu, S.—Qi, L.—Qin, H.—Shi, J.—Jia, J.: Path Aggregation Network for Instance Segmentation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.

[35] Lin, T. Y.—Dollar, P.—Girshick, R.—He, K.—Hariharan, B.—Belongie, S.: Feature Pyramid Networks for Object Detection. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.

[36] Ioffe, S.—Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach, F., Blei, D. (Eds.): Proceedings of the 32nd International Conference on Machine Learning. PMLR, Proceedings of Machine Learning Research, Vol. 37, 2015, pp. 448–456.

[37] Ding, X.—Zhang, X.—Ma, N.—Han, J.—Ding, G.—Sun, J.: RepVGG: Making VGG-Style ConvNets Great Again. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13728–13737, doi: 10.1109/CVPR46437.2021.01352.

[38] Wu, Y.—He, K.: Group Normalization. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.): Computer Vision – ECCV 2018. Springer, Cham, Lecture Notes in Computer Science, Vol. 11217, 2018, pp. 3–19, doi: 10.1007/978-3-030-01261-8_1.

[39] Ding, X.—Zhang, X.—Han, J.—Ding, G.: Diverse Branch Block: Build-

ing a Convolution as an Inception-Like Unit. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 10881–10890, doi: 10.1109/CVPR46437.2021.01074.

[40] Ba, J. L.—Kiros, J. R.—Hinton, G. E.: Layer Normalization. 2016, doi: 10.48550/arXiv.1607.06450.

[41] Li, B.—Wu, F.—Weinberger, K. Q.—Belongie, S.: Positional Normalization. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 32 (NeurIPS 2019). Curran Associates, Inc., 2019, pp. 1622–1634, doi: 10.48550/arXiv.1907.04312.

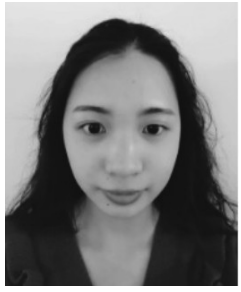
[42] Huang, L.—Zhou, Y.—Wang, T.—Luo, J.—Liu, X.: Delving into the Estimation Shift of Batch Normalization in a Network. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 753–762, doi: 10.1109/CVPR52688.2022.00084.

[43] Hu, J.—Shen, L.—Sun, G.: Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.

[44] Long, Y.—Gong, Y.—Xiao, Z.—Liu, Q.: Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. IEEE Transactions on Geoscience and Remote Sensing, Vol. 55, 2017, No. 5, pp. 2486–2498, doi: 10.1109/TGRS.2016.2645610.

[45] Xiao, Z.—Liu, Q.—Tang, G.—Zhai, X.: Elliptic Fourier Transformation-Based Histograms of Oriented Gradients for Rotationally Invariant Object Detection in Remote-Sensing Images. International Journal of Remote Sensing, Vol. 36, 2015, No. 2, pp. 618–644, doi: 10.1080/01431161.2014.999881.

[46] Everingham, M.—Eslami, S. M. A.—Van Gool, L.—Williams, C. K. I.—Winn, J.—Zisserman, A.: The Pascal Visual Object Classes (VOC) Challenge: A Retrospective. International Journal of Computer Vision, Vol. 111, 2015, No. 1, pp. 98–136, doi: 10.1007/s11263-014-0733-5.



Sha Li is studying in Qingdao University of Science and Technology, majoring in electronic information, mainly researching about marine biological object detection.

Yong Liu graduated from the Ocean University of China in 2011, majoring in computer application technology. She is dedicated to the research of intelligent identification and quantitative analysis of marine organisms, medical knowledge graph and intelligent medical big data. So far, she has obtained one national invention patent, seven software copyrights, and published more than 20 research papers (SCI/EI). She has published two textbooks.

Shuang Wu is studying in Qingdao University of Science and Technology, majoring in electronic information, mainly researching about marine biological object detection.

Shoujiang Zhang is the engineer of Haier Group, he has long been devoted to the research of artificial intelligence in the field of home appliances, and has obtained 7 invention patents.