frontiers | Frontiers in Marine Science

TYPE Original Research
PUBLISHED 27 July 2023
DOI 10.3389/fmars.2023.1219155

OPEN ACCESS

EDITED BY
Simone Marini,
National Research Council (CNR), Italy

REVIEWED BY
Daniele D'Agostino,
University of Genoa, Italy
Luciano Ortenzi,
University of Tuscia, Italy

*CORRESPONDENCE
Tingqiang Song
songtq@qust.edu.cn

RECEIVED 08 May 2023
ACCEPTED 12 July 2023
PUBLISHED 27 July 2023

CITATION
Liang H and Song T (2023) Lightweight marine biological target detection algorithm based on YOLOv5.
Front. Mar. Sci. 10:1219155.
doi: 10.3389/fmars.2023.1219155

Lightweight marine biological target detection algorithm based on YOLOv5

Heng Liang and Tingqiang Song*

College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, China

Underwater object detection currently faces many challenges, such as the large number of parameters in existing object detection models, slow inference speed, blurring of underwater images, and aggregation of small targets, making it difficult to conduct efficient underwater object detection. This paper proposes a lightweight underwater object detection algorithm based on YOLOv5.The method uses depth-wise separable convolution instead of ordinary convolution to reduce the number of parameters and computational complexity. A C3 module based on Ghost convolution is designed to further compress the model size and improve the computational speed. In the feature extraction stage, a RepVgg module based on structural reparameterization is used to convert the multi-branch structure into a single-branch structure in the inference stage, improving the feature extraction ability of the model and increasing the inference speed. A Rep-ECA module is designed to embed the efficient channel attention module ECANet into the RepVGG module, selecting more effective channel information and improving the model's feature extraction ability for small objects in blurred images, thereby improving detection precision. Experimental results show that in the URPC underwater object detection dataset, the proposed algorithm has a 39% lower model parameter count compared to the original model, a 42% reduction in computational complexity. The model can achieve a frame rate of 85 on a single Nvidia GTX 1080ti GPU, which is a 24% improvement over the original model, while mAP reaches 85.1%, a 1.1% improvement over the original model. The algorithm can improve the detection precision and achieve lightweight, which lays a foundation for the deployment of underwater equipment.

KEYWORDS

underwater object detection, YOLOv5, Ghost convolution, RepVGG, ECANet

1 Introduction

The marine underwater environment is extremely complex, and there are many potential dangers, which make the development and exploration of the ocean still face many difficulties and challenges. Today, with the rapid development of artificial intelligence and robotics industries, using underwater robots for exploration has become an important

method for humans to explore the ocean world. By controlling underwater robots to carry out underwater operations and seabed exploration (Sahoo et al., 2019) can be widely used in many fields such as fishery industry, biological research, marine mineral exploration, pipeline inspection, and marine archaeology. It can not only reduce the risk of manual operation and improve work efficiency but also reduce costs, which can be well adapted to the development direction of future marine economy and help the growth of the water industry.

For underwater robots, the key technology is the detection and recognition of underwater objects. Underwater object detection is a key technology for human exploration of the ocean. By equipping robots with underwater cameras, optical images of targets can be obtained, which are intuitive and provide more abundant underwater information, making them superior in detecting objects at close range underwater. However, unlike land images, the underwater environment is complex and the lighting is uneven (Liang et al., 2022). Under the influence of water absorption, underwater light is dark, and the collected image information suffers from problems such as low clarity, blurring, and color distortion. In addition, there are many small targets in underwater images. These small targets have a small coverage range in the image and there are problems with clustering, occlusion, and insufficient feature expression. Therefore, it is difficult to accurately recognize underwater targets.

With the rapid development of artificial intelligence technology, computer vision has provided new tools for human exploration of the ocean (Marini et al., 2018). In response to the issue of underwater image blurring, many scholars use deep learning methods for underwater image enhancement. Lopez-Vazquez et al. (2023) proposed a convolutional residual network for underwater image enhancement, while achieving high classification accuracy. Guan et al. (2023) proposed a lightweight underwater image enhancement model based on GAN, which improves the quality and efficiency of underwater image generation. Meanwhile, many scholars have done extensive research on the object detection of marine organisms. For example, Bazeille et al. (2007) proposed a completely color based underwater object detection method, which models the color changes of underwater objects and detects them by comparing color information. However, this method does not take into account the color similarity of continuous samples, resulting in low detection accuracy. Chuang et al. (2016) proposed a fish recognition algorithm based on completely unsupervised feature learning and robustness detection fusion, which can accurately identify underwater fish schools. Qin et al. (2016). proposed a deep framework for underwater fish recognition by combining principal component analysis (PCA), spatial pyramid pooling (SPP), and SVM classifiers, achieving high precision. However, the training process of this method is too cumbersome to conduct end-to-end training. Banan et al. (2020) designed an algorithm based on the VGG network to recognize fish species, which has high precision but can only recognize single-class objects. Bonofiglio et al. (2022) used the YOLO model and target tracking algorithm to identify and classify sablefish. Huang et al. (2022) used an improved SSD network for underwater object detection, and improved the detection accuracy by adding attention mechanisms and feature fusion. However, the models are still complex. Shi and Wang (2023) proposed an improved lightweight underwater object detection network based on YOLOv4. The model significantly reduces the model size, but the parameter volume is still 49.2M and requires further optimization.

In summary, deep learning has made outstanding progress in object detection and other aspects (Hoeser and Kuenzer, 2020; Zhao et al., 2019; Zaidi et al., 2021), achieving good recognition results in large datasets such as COCO (Lin et al., 2014) and VOC (Everingham et al., 2010). This provides new development directions for underwater image object detection and recognition. However, the problem of lightweight network model while improving network accuracy has always been a pain point and difficulty in underwater object detection. Compared to the two-stage object detection models: RCNN (Girshick et al., 2014), Fast-RCNN (Girshick, 2015), Faster-RCNN (Ren et al., 2016) and Mask-RCNN (He et al., 2017), YOLO has faster and higher detection precision. Therefore, this study designed a lightweight network based on the faster one-stage network model YOLOv5s.The main contributions of this paper are as follows: (1) Using depthwise separable convolution instead of conventional convolution to reduce the number of parameters and computation complexity, (2) Designing a C3 structure based on Ghost convolution to further reduce parameter and computational complexity,(3) Using the RepVGG module with structural reparameterization technology to ensure inference speed while improving the network's feature extraction ability. (4) Embedding an efficient channel attention module into the RepVGG module to enhance the network's ability to filter important channels and further improve detection precision.

The remaining sections of this article are as follows: In Section 2, we introduce the dataset and image preprocessing methods used in this study, as well as the evaluation metrics and model parameter settings used in the experiments. In Section 3, we provide a detailed description of the proposed improvement methods. In Section 4, we conduct ablation experiments and discuss the experimental results. In Section 5, we summarize and prospect the proposed algorithm.

2 Dataset and training strategy

2.1 Dataset and data augment

This study uses the dataset from the Underwater Robot Picking Contest (URPC). All the images in this dataset were taken by underwater robots, and the competition provides annotated data for four categories: starfish, holothurian, echinus, and scallop. The competition officials have noted that the dataset contains annotation noise. Therefore, before training, we reannotated the mislabeled and missed data. The dataset used in this study contains 5543 images, which were divided into a training set and a validation set in an 8:2 ratio. A total of 4434 images were used for training, and 1109 images were used for validation. Some sample images from the dataset are shown in Figure 1.

Observing some of the images in the dataset, there are many small objects that are clustered together, and there is some degree of

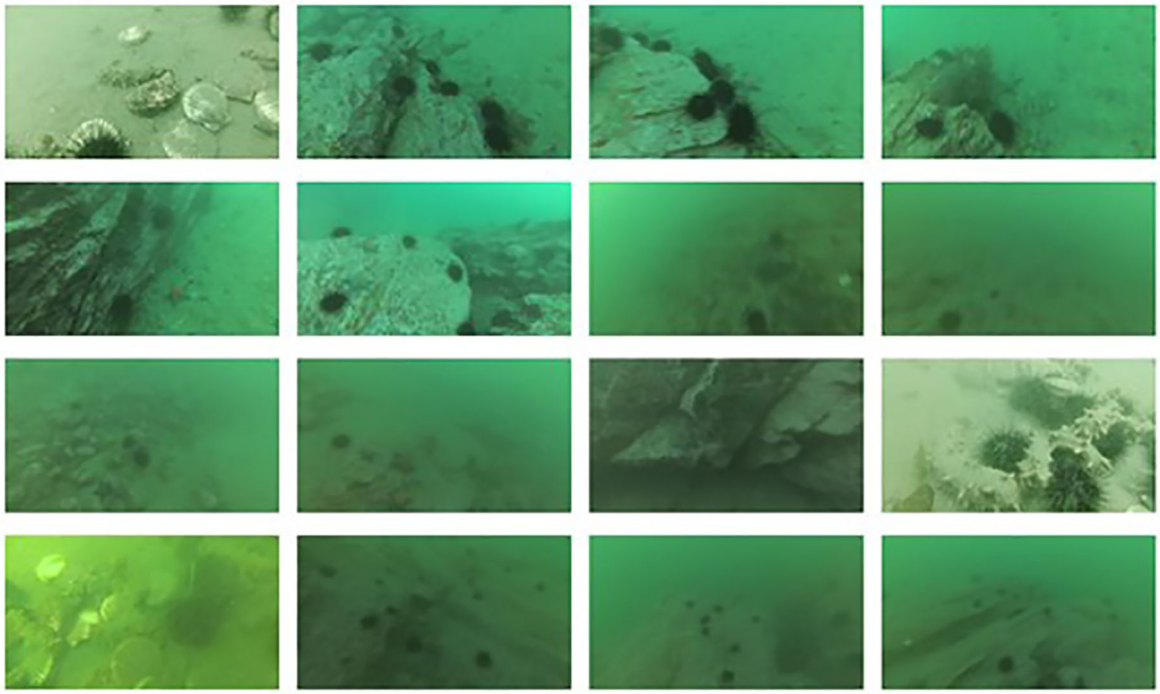

FIGURE 1
Sample Images from the Dataset.

occlusion between objects. Moreover, due to the influence of light scattering, there is a serious color deviation problem, which seriously affects the accuracy of object detection. Additionally, the samples for each class in the dataset are not balanced, which also poses a challenge for underwater object detection. Specifically, there are 16217 echinus, 4075 holothurian, 4838 starfish, and 4950 scallop in the dataset, as shown in Figure 2.

According to the label size distribution shown in Figure 2B, most of the labels in the dataset have a ratio of width and height to the original image's width and height between 0 and 0.1. Currently, there are two main ways to define small objects: based on absolute scale and based on relative scale. According to the definition of relative scale of small objects, objects with a ratio of size to the original image size less than 0.1 are considered small object (Wang et al., 2021). Therefore, there are a large number of small objects in the dataset, and the proposed improved model in this paper can improve the detection performance of small objects.

Since the distribution of each category in the dataset used in this experiment is unbalanced, it will affect the training effect of the model. Therefore, before inputting the images into the network, image data augmentation techniques such as random flipping, cropping, and rotation are used to enhance the images. Moreover, underwater images are generally dark and have serious color deviation problems, so the brightness, contrast, saturation, and

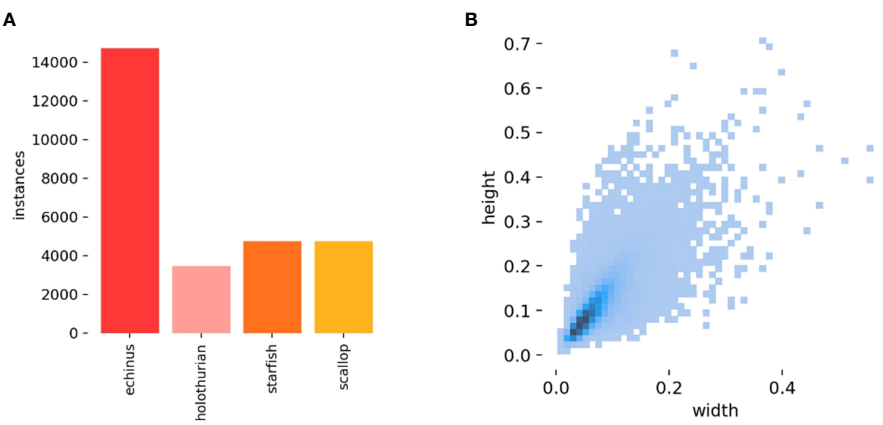

FIGURE 2
Visualization of the dataset used in this study: **(A)** distribution of label categories, **(B)** distribution of label sizes.

other aspects of the images are adjusted to enhance the network's fitting ability. Finally, the Mosaic data augmentation technique is used to combine four random images during training to alleviate the impact of the unbalanced distribution of categories in the dataset.

2.2 Model evaluation metrics

This paper mainly uses P (precision), R (recall) (Fawcett, 2006), AP (average precision), and mAP (mean average precision) as the evaluation metrics for model accuracy, and frames per second (Number of images processed per second) as the evaluation metric for model speed. At the same time, we use the number of parameters and computational complexity to evaluate the model, where the computational complexity (GFLOPS) is 10^9 floating-point operations per second, The calculation of P, R, AP and mAP is shown in formulas (1) - (4)

$$P = \frac{TP}{TP+FP} \tag{1}$$

$$R = \frac{TP}{TP+FN} \tag{2}$$

$$AP = \int_0^1 P(R)\mathrm{d}R \tag{3}$$

$$mAP = \frac{\sum_{i=1}^{M} AP_i}{M} \tag{4}$$

In formulas (1) and (2), TP represents the number of correctly identified instances of a certain class, FP represents the number of instances mistakenly identified as that class, and FN represents the number of instances missed. A PR curve for a certain class is plotted with P as the x-axis and R as the y-axis, and the average precision (AP) is obtained by calculating the area under the PR curve. The mean average precision (mAP) is calculated as the average of AP across all classes to comprehensively evaluate the performance of the model. The calculations for AP and mAP are shown in formulas (3) and (4).

2.3 Model training and parameter settings

All experiments in this paper were conducted on a Linux system, with an operating system version of Ubuntu 20.04, an Intel Xeon E5-2650 V4 CPU, 32GB of RAM, and an NVIDIA GeForce GTX 1080ti graphics card. The Python version used was 3.9.0, PyTorch version was 1.12.1, and CUDA version was 11.0.

In this experiment, the number of training epochs was set to 120, the batch size was set to 16, and the initial learning rate was set to 0.01. The learning rate was warmed up using a specific method. The optimizer used in the experiment was SGD, with a weight decay coefficient of 0.5. Pretrained weights provided by the official were used, and the size of the input images to the network was set to 640×640 by default. K-means algorithm was used to select the initial anchor boxes.

3 Methods

Due to the low contrast and blurriness of underwater images, as well as the clustering of small objects, it is necessary to fuse high-level and low-level information to improve the detection ability of the model for objects of different sizes. The YOLOv5 model uses the PAN (Liu et al., 2018) and FPN (Lin et al., 2017) strategy to achieve detection of objects of different sizes. In this paper, we selected the YOLOv5s network as the baseline network and designed a object detection model suitable for marine organisms, as shown in Figure 3. It mainly consists of three parts: the backbone network,

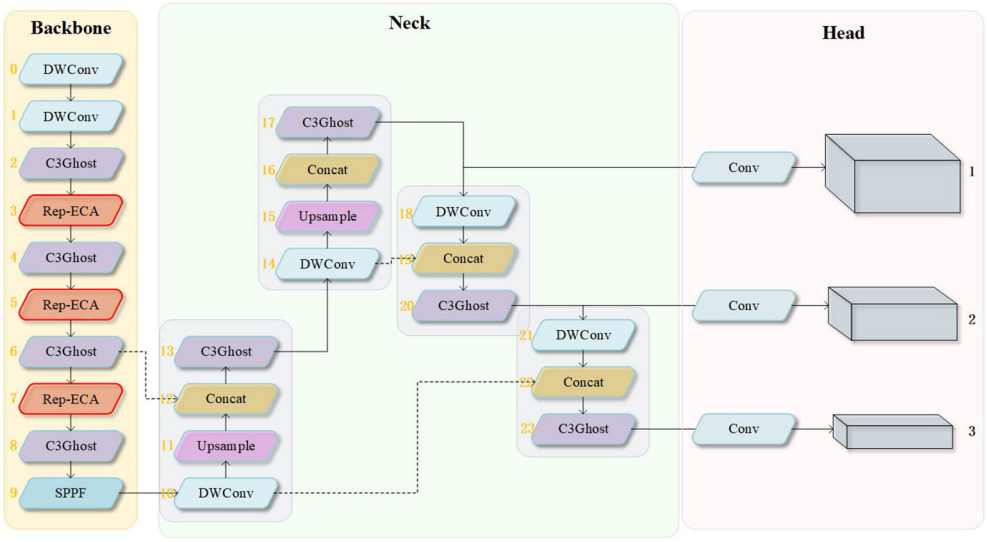

FIGURE 3
A lightweight detection structure based on YOLOv5.

the neck feature fusion network, and the prediction head. This paper mainly makes the following improvements to YOLOv5s:

1) Using depthwise separable convolutions instead of the regular convolutions in the feature extraction backbone and neck feature fusion network has the effect of reducing the number of parameters and computational complexity.
2) A C3Ghost module based on Ghost convolution was designed, and the original C3 model in the algorithm was replaced with the C3Ghost module, further reducing the number of network parameters and improving the model detection speed.
3) To address the problem of reduced feature extraction ability and detection precision caused by depthwise separable convolution and Ghost convolution, a RepVGG module was introduced into the feature extraction backbone. The module uses a multi-branch structure to enhance the model's feature extraction ability in the training phase and uses structural reparameterization to convert the multi-branch structure into a single-branch structure in the inference phase, while maintaining high efficiency and improving detection precision with only a small number of additional parameters.
4) To address the problem of blurry underwater images with small target clusters, a Rep-ECA module is proposed, which embeds a more efficient channel attention mechanism, the ECA module, into the RepVGG module. This module discriminates the importance of different channels, enhances the network's ability to extract features of small targets in blurry backgrounds, and further improves detection accuracy.

3.1 Depthwise separable convolution

To reduce the number of parameters and computation, this paper replaces the regular convolution in the YOLO network with depthwise separable convolution (Chollet, 2017). Depthwise separable convolution consists of depthwise convolution and pointwise convolution. In regular convolution, each convolution kernel corresponds to all channels of the feature map, while in depthwise convolution, each convolution kernel corresponds to one channel, and the number of channels in the resulting feature map is the same as that in the original feature map. However, because depthwise convolution operates on each channel separately, it cannot achieve information interaction between channels. Therefore, pointwise convolution is used to combine information between channels to generate new feature maps. The operations of regular convolution and depthwise separable convolution are shown in Figures 4 and 5, respectively.

When using standard convolution operation, assuming the size of the input feature map is $c \times h \times w$, where c represents the number of channels of the input feature map, h represents the height of the feature map, and w represents the width of the feature map. The size of the output feature map is $h\} \times w\} \times n$, where n represents the number of channels of the output feature map. The size of the convolution kernel is $k \times k \times m \times n$, where k represents the kernel sizes. The number of parameters (P_c) is shown in Formula (5), and the computational complexity (G_C)is shown in Formula (6).

$$P_c = k \times k \times c \times n \tag{5}$$

$$G_C = k \times k \times c \times n \times h' \times w' \tag{6}$$

When using depthwise separable convolution operation, a $k \times k \times c$ convolutional kernel is used to perform depth convolution on the input feature map firstly. Then, a $1 \times 1 \times c \times n$ convolutional kernel is used to perform point-wise convolution on the obtained feature map to fuse information between different channels. The number of parameters of depthwise separable convolution (P_D) is shown in Formula (7), and the computational complexity (G_D) is shown in Formula (8).

$$P_D = k \times k \times c + 1 \times 1 \times c \times n \tag{7}$$

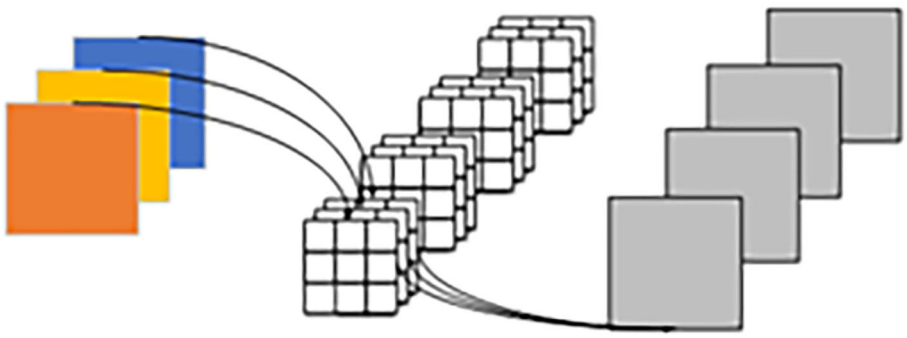

FIGURE 4
Regular convolution operation.

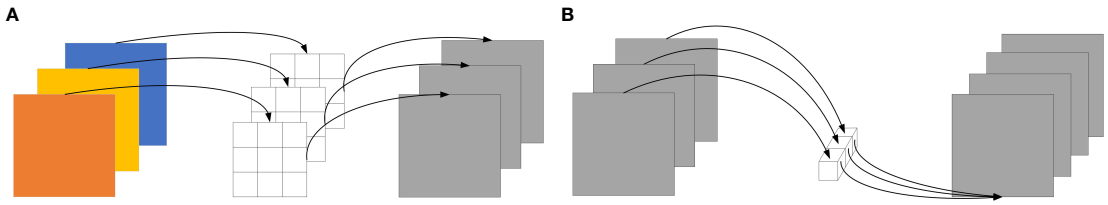

FIGURE 5
Depthwise separable convolution operation. **(A)** depthwise convolution, **(B)** pointwise convolution operation.

$$G_D = k \times k \times c \times h' \times w' + 1 \times 1 \times c \times n \times h' \times w' \quad (8)$$

Depthwise separable convolution operation can significantly reduce the number of parameters and computational complexity compared to standard convolution operation. Formula (9) is the ratio of the number of parameters and computational complexity between the two.

$$R_C = \frac{1}{k^2} + \frac{1}{n} \quad (9)$$

From Formula (9), it can be seen that the number of parameters and computational complexity of depthwise separable convolution depends on the size of the convolution kernel. When the size of the convolution kernel is 3×3, the computational complexity and number of parameters of depthwise separable convolution will decrease to about 1/9 of that of standard convolution. This reduces the training time of the model and improves its inference speed.

3.2 C3Ghost module

In order to further reduce the size of the model, this paper based on the idea of GhostNet (Han et al., 2020) and incorporated Ghost convolution into the C3 structure of the YOLOv5 model. Ghost convolution consists of two parts: the first part is a normal convolution, and the second part is a cheap linear operation. It has been experimentally proven that the feature maps produced by regular convolution usually contain a lot of redundancy (Han et al., 2020). Therefore, the core idea of Ghost convolution is to use a small number of convolution kernels to generate a part of the feature map by using regular convolution, then extract information from these feature maps by using depthwise convolution, and finally merge concat two parts of feature maps to generate a light-weight feature map. The Ghost convolution process is shown in Figure 6

Assuming that the input feature map is $F \epsilon R^{c \times h \times w}$, Ghost convolution first performs a regular convolution operation on the feature map, as shown in Formula (10).

$$Y' = F \times \omega \quad (10)$$

$$Y_{ij} = \varphi_{i,j}(Y_i') \quad (11)$$

In the formula, ω represents the Convolutional kernel. Y_i' represents the i- *th* feature map in Y', $\varphi_{i,j}$ represents the *j- th* linear operation in the i- *th* feature map of Y_i', and Y_{ij} represents all the feature maps obtained after linear operation.

In the last stage of Ghost convolution, the two feature maps are concatenated to obtain the final feature map. Compared with

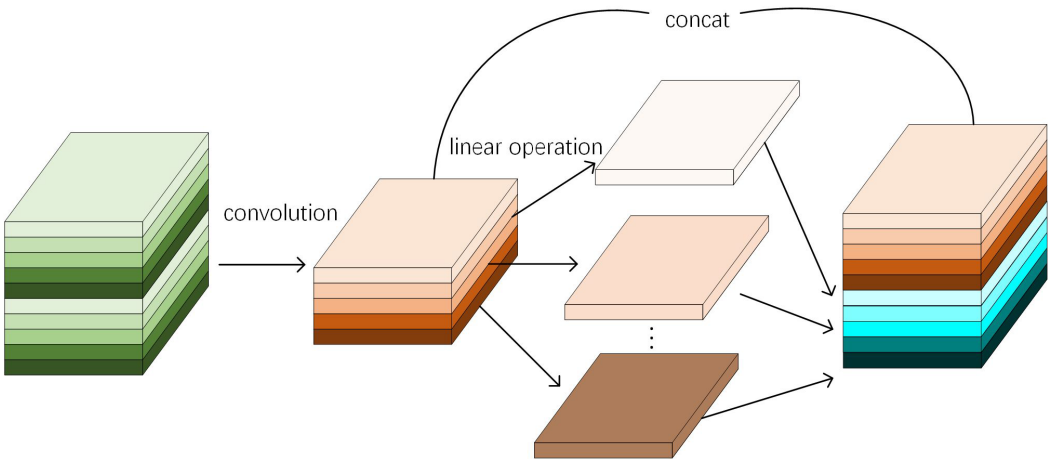

FIGURE 6
Ghost convolution process.

regular convolution, Ghost convolution can significantly reduce computational complexity. The computational complexity formula of Ghost convolution is shown in Formula (12).

$$G_{ghost} = \frac{n}{s} \times h' \times w' \times k \times k \times c + (s-1) \times \frac{n}{s} \times h' \times w' \times d \times d \quad (12)$$

In the formula (12), s represents the channel compression ratio of the first convolution, $d \times d$ represent the kernel sizes. Based on the above formula and the computational complexity formula of regular convolution, the ratio of Ghost convolution to regular convolution can be obtained as shown in Formula (13).

$$\begin{aligned} R_G &= \frac{k \times k \times c \times n \times h' \times w'}{\frac{n}{s} \times h' \times w' \times c \times k \times k + (s-1) \times \frac{n}{s} \times h' \times w' \times d \times d} \\ &= \frac{k \times k \times c}{\frac{1}{s} \times k \times k \times c + \frac{s-1}{s} \times d \times d} \\ &= \frac{s \times c}{s-1} \\ &\approx s \end{aligned} \quad (13)$$

Therefore, it can be seen that the computational complexity of regular convolution is about s times that of Ghost convolution, which theoretically proves that using Ghost convolution to reduce parameter and computational complexity is feasible. Therefore, this paper improves the C3 structure in YOLOv5 by using Ghost convolution and depthwise separable convolution to construct the GhostBottleNeck structure. The GhostBottleNeck main branch consists of two consecutive Ghost convolutions, and to reduce parameter, the shortcut branch uses a depthwise separable convolution. Based on this structure, the C3 structure is rebuilt as C3Ghost, which contains three regular convolutions and n GhostBottleNeck structures. The interaction between regular convolution and Ghost convolution in C3Ghost module is beneficial to balance detection precision and speed. The improved C3 structure is shown in Figure 7.

3.3 RepVGG module

This paper uses Ghost convolution and depthwise separable convolution to reduce parameter and computational complexity, which may lead to a decrease in detection precision. Therefore, we use the RepVGG (Ding et al., 2021) module to improve the feature extraction ability of the model. The RepVGG structure is simple, mainly composed of 3×3 convolution and ReLU activation function. The RepVGG module draws inspiration from the idea of ResNet (He et al., 2018; Shafiq and Gu, 2022). During training, shortcut branches are created to reduce the performance degradation caused by gradient vanishing and gradient explosion. However, the multi-branch structure will increase memory consumption. As shown in Figure 8A, the calculation results of each step in the multi-branch structure will be saved in memory, which will bring huge computational and storage costs. In the inference stage, the RepVGG module adopts structural reparameterization strategy to decouple training and inference, and convert the multi-branch structure into a single-branch structure to improve the model's inference speed. The RepVGG module is shown in Figure 8.

3.4 Rep-ECA module

Due to the influence of underwater light, the images captured by underwater robots are usually very blurry. Attention mechanisms can improve network performance. To enhance the model's ability

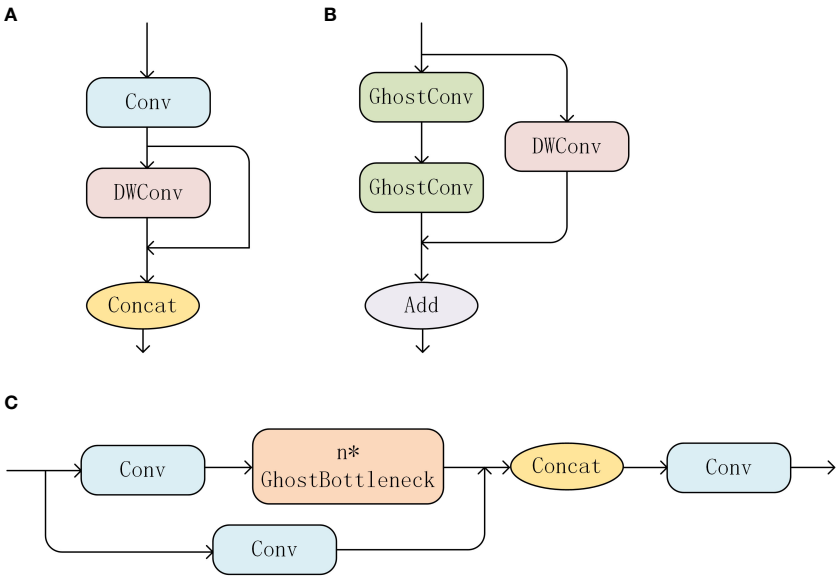

FIGURE 7
C3Ghost series modules. **(A)** Ghost Conv module, **(B)** GhostBottleNeck module, **(C)** C3Ghost module.

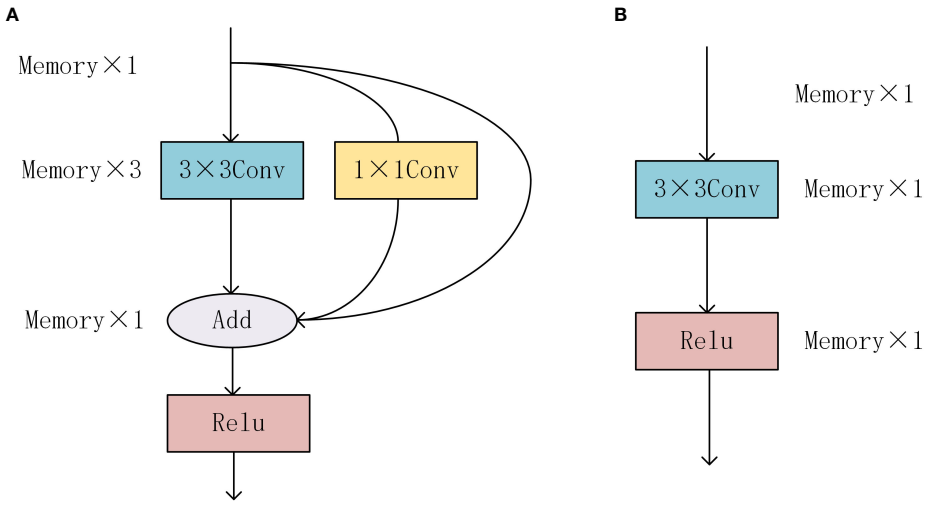

FIGURE 8
RepVGG module structure diagram, **(A)** Multi branch Repvgg module during training phase. **(B)** Single branch Repvgg module in inference stage.

to focus on underwater biological targets, this study adds the ECA (Efficient Channel Attention) module on the basis of the RepVGG module. The ECA module is an improved version of the channel attention module. It was proposed by Wang et al. (2022) Compared with the SE (Hu et al., 2018) module, the ECA module can reduce the number of parameters. The ECA module replaces the two fully connected layers of the SE module with a one-dimensional convolution, avoiding dimension reduction and achieving information interaction between channels, thereby reducing the number of parameters. The calculation process of the ECA attention module is shown in Figure 9.

Figure 9 shows the calculation process of the ECA attention mechanism. First, the input feature map undergoes global average pooling to obtain a one-dimensional feature map. Then, it undergoes a one-dimensional convolution layer with an adaptive kernel size to obtain the importance of each channel. After that, the Sigmoid function is used for normalization operation and multiplied by the input feature map to obtain the filtered feature map through ECA. In this study, the ECA module is added to the RepVGG module to allocate different weights to different channels of the feature map, thereby filtering the feature information of marine organisms, suppressing irrelevant information, and improving detection precision.

As there is currently no theoretical research on which part of the network the ECA module should be embedded to achieve the best network performance, this study designed three types of RepVGG modules based on the ECA attention mechanism, as shown in Figure 8. The first type adds the ECA attention mechanism before the 3×3 convolution, as shown in Figure 10A. The second type adds the ECA module after the 3×3 convolution and before the feature map addition operation of the three branches, as shown in Figure 10B. The third type adds the ECA module after the feature map addition operation of the three branches, as shown in Figure 10C.

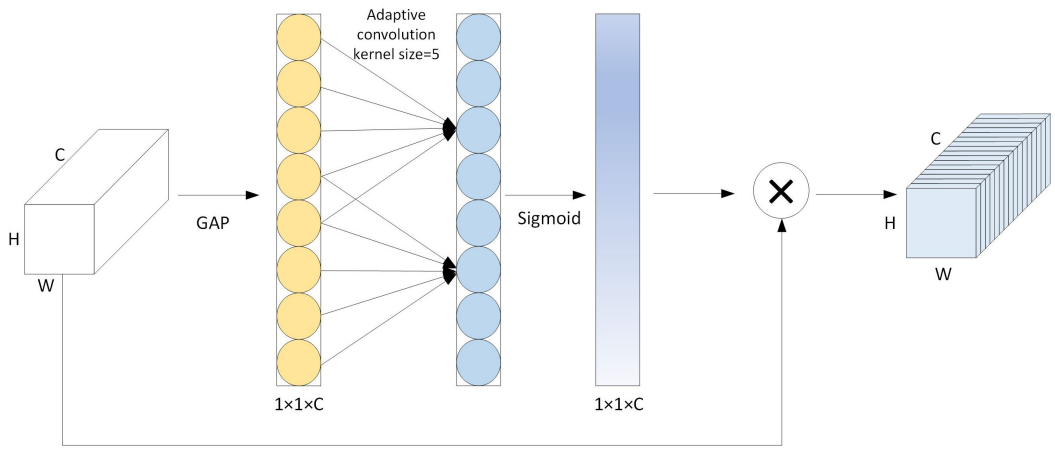

FIGURE 9
Calculation process of the ECA module.

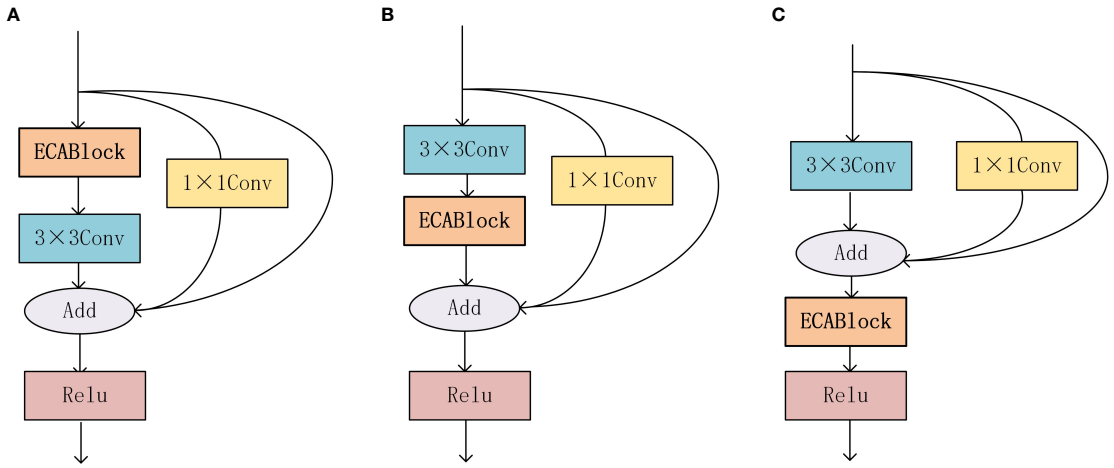

FIGURE 10
RepVGGBlock+ECA structure diagram, **(A)** Rep-ECA1 structure diagram, **(B)** Rep-ECA2 structure diagram, **(C)** Rep-ECA3 structure diagram.

4 Experiments and discussion

4.1 Comparative experiments on depthwise separable convolution

To reduce the model's parameter and computational complexity, we use depth-wise separable convolution to optimize the model. The conventional convolution in the feature extraction backbone and feature fusion parts of YOLOv5 is replaced by depth-wise separable convolution, and the optimized model is named YOLOv5s-D. Comparative experiments were conducted on the URPC dataset to compare the optimized model with the original model, and the experimental results are shown in Table 1.

According to Table 1, using depthwise separable convolution can significantly reduce the number of parameters and computational complexity of the original YOLOv5 algorithm. The number of parameters is reduced from 7.03×10^6 to 4.59×10^6, and the computational complexity is reduced from 16.0GFlOPs to 12.0GFlOPs. At the same time, it also improves the inference speed to some extent. However, the lightweight of the model may lead to a certain degree of decrease in detection precision.

4.2 Comparative experiments on RepVGG modules

Due to the decrease in detection performance caused by using depthwise separable convolution to reduce parameter and computational complexity, we use RepVGG modules to improve the detection performance of the model. To verify the effectiveness of the RepVGG modules, the YOLOv5s model is improved by incorporating RepVGG modules into the feature extraction backbone, and the improved model is named YOLOv5s-Rep. The experimental results are shown in Table 2.

Comparing the experimental results in Table 2, it can be found that using RepVGG modules can improve the detection accuracy of the model from 84.0% in the original YOLOv5s algorithm to 84.84%. Although the number of parameters and computational complexity of the YOLOv5s-Rep model slightly increased, the FPS values for both models are not significantly different. This is because the RepVGG module uses structural reparameterization during inference, converting the multi-branch structure in the training phase to a single-branch structure during inference to ensure inference speed.

TABLE 1 Comparative experiments on depthwise separable convolution.

Algorithm	Param(M)	GFLOPs(G)	mAP(%)	FPS
YOLOv5s	7.03	16.0	84.0	64
YOLOv5s-D	4.59	12.0	83.14	80

TABLE 2 Comparative experiments on RepVGG modules.

Algorithm	Param(M)	GFLOPs(G)	mAP(%)	FPS
YOLOv5s	7.03	16.0	84.0	64
YOLOv5s-Rep	7.20	16.3	84.84	63

4.3 Attention mechanism comparative experiments

To verify the effectiveness of the ECA module for underwater object detection algorithms and to validate the best way to incorporate the ECA module into the RepVGG module, this paper conducted comparative experiments by integrating the three types of RepVGG modules based on the ECA attention mechanism designed in Section 3.4 into YOLOv5s. These three models are named YOLOv5s-Rep-ECA1, YOLOv5s-Rep-ECA2, and YOLOv5s-Rep-ECA3, respectively. The experimental results are shown in Table 3.

Comparing the experimental results shown in Table 3, it can be concluded that adding the ECA attention mechanism to the RepVGG module can improve the model accuracy while keeping the number of parameters and computational complexity basically unchanged. This proves that the ECA attention mechanism can make the network pay more attention to useful object information in blurry underwater images. Among the three different embedded methods, the third method can improve the model's detection precision by 0.73%. Therefore, this paper uses the method of embedding the ECA attention mechanism after the three branches are added in the RepVGG module, and the model is named YOLOv5s-Rep-ECA. To further verify the impact of introducing different attention mechanisms on the model, this paper conducts comparative experiments by adding the SE (Hu et al., 2018), CBAM(Woo et al., 2018), CA (Hou et al., 2021), and the RepVGG module with the third embedding method of the ECA attention mechanism named YOLOv5s-Rep-ECA based on YOLOv5s-Rep. The experimental results are shown in Table 4.

Comparing the comparative data of different attention mechanisms in Table 4, it can be concluded that using attention modules can improve the detection accuracy of underwater object detection. This is because the color information of underwater objects is similar to the background information, and introducing attention mechanisms can highlight the feature information of these objects, thus improving the performance of the network model. In terms of accuracy, the ECA attention mechanism embedded in the RepVGG module achieves the largest accuracy improvement. In terms of speed, adding attention mechanisms usually makes the model more complex, so the inference speed of the model is reduced to some extent. The ECA attention mechanism has the least speed loss. In summary, embedding the ECA attention mechanism in the RepVGG module performs better than other attention mechanisms in improving the model.

4.4 Ablation experiment

To further verify the effectiveness of the proposed improved algorithm in this paper, ablation experiments were conducted on several proposed improvements. The ablation experiment design is shown in Table 5, where √ indicates that the method was used in the experiment, and × indicates that the method was not used in the experiment. Table 6 shows the results of the six groups of ablation experiments, which were conducted under the same configuration environment and parameter settings.

According to the experimental results shown in Table 6, Model 1 is the unimproved YOLOv5s, Model 2 improves the conventional convolution by using depthwise separable convolution on the basis of YOLOv5s, reducing the model's parameter and computational complexity, proving the effectiveness of depth-wise separable convolution. Model 3 improves the C3 structure to C3Ghost structure based on Model 1, and the model's parameter and computational complexity are also reduced to some extent, proving the effectiveness of the C3Ghost structure. Model 4 adds the C3Ghost module on the basis of depthwise separable convolution, further reducing the model's parameter and computational complexity. Although the mAP has slightly decreased, it has decreased by 1.06% compared with the original

TABLE 3 Comparative experiments on the ECA module.

Algorithm	Param(M)	GFLOPs(G)	mAP(%)	FPS
YOLOv5s-Rep	7.20	16.3	84.84	63
YOLOv5s-Rep-ECA1	7.20	16.3	85.01(+0.17)	60
YOLOv5s-Rep-ECA2	7.20	16.3	85.06(+0.22)	60
YOLOv5s-Rep-ECA3	7.20	16.3	85.57(+0.73)	60

TABLE 4 Comparative experiments on different attention modules.

Algorthim	Param(M)	GFLOPs(G)	Map(%)	FPS
YOLOv5s-Rep	7.20	16.3	84.84	63
YOLOv5s-Rep-ECA	7.20	16.3	85.57(+0.73)	60
YOLOv5s-Rep-SE	7.24	16.4	84.96(+0.12)	57
YOLOv5s-Rep-CBAM	7.25	16.5	85.13(+0.29)	54
YOLOv5s-Rep-CA	7.24	16.4	85.20(+0.36)	55

TABLE 5 Ablation experiment design.

Order Number	DWConv	C3Ghost	RepVGGBlock	ECA
1	×	×	×	×
2	√	×	×	×
3	×	√	×	×
4	√	√	×	×
5	√	√	√	×
6	√	√	√	√

YOLOv5 model. However, the inference speed of the model has further improved, with an increase of 24 compared to the original model, proving that the combination of the two can also make the model lighter. Model 5 adds the RepVGG module on the basis of Model 4, and the parameter and computational complexity have slightly increased compared to Model 4. However, the model's accuracy has increased by 1.3% compared to Model 4 and 0.24% compared to the original model, and the inference speed has not changed compared to Model 4, proving that the RepVGG module can improve the network's feature extraction ability and thus improve the network's accuracy. Model 6 is the final version of the algorithm improvement. Compared with Model 1, its accuracy has improved by 1.1%, its parameter has reduced by 2.72×10^6, its computational complexity has reduced by 6.7GFLOPs, and at the same time, its inference speed has improved by 21. In summary, the proposed model in this paper has better detection performance for underwater object detection in terms of both speed and accuracy.

In summary, the proposed improved algorithm in this paper has certain advantages in both speed and precision compared with different object detection algorithms. Lightweight models are more convenient to deploy in underwater equipment. Meanwhile, the proposed improved algorithm improves the impact of low underwater image quality on object detection. It can perform high-precision object detection in blurry underwater images.

4.5 Comparative experiments on mainstream object detection algorithms

To further verify the superiority of the proposed algorithm compared to other mainstream algorithms, this paper conducted comparative experiments between the improved model and current mainstream object detection models, including single-stage object detection algorithms: SSD (Liu et al., 2016), Retinanet (Lin et al., 2017), two-stage object detection algorithm Faster-RCNN, YOLO series algorithms: YOLOv3 (Redmon and Farhadi, 2018), YOLOv4 (Bochkovskiy et al., 2020), YOLOv7 (Wang et al., 2022), and Anchor-Free based CenterNet (Duan et al., 2019) algorithm. All experiments were conducted under the same environment and parameter settings, and the experimental results are shown in Table 7.

According to the experimental data shown in Table 7, compared with single-stage object detection algorithms SSD and RetinaNet, this study has significant advantages in terms of parameter and computational complexity, detection precision and inference speed. Compared with the traditional two-stage object detection algorithm Faster-RCNN, the proposed algorithm has a more significant improvement in speed and accuracy. Compared with the YOLOv3 model, the parameter has decreased by nearly 11 times, the computational complexity has decreased by about 9 times, and the mean average precision has increased by 12.27%. For the lightweight YOLOv4-tiny model, the parameter has decreased by 1.79×10^6, and although the computational complexity has increased by 2.3GFLOPs, the detection mean average precision has increased by 21.84%, and the inference speed has increased from 55FPS to 85FPS. For the lightweight YOLOv7-tiny model, the parameter has decreased by 1.69×10^6, the computational complexity has decreased by 3.9GFLOPs, and the detection mean average precision has increased by 1.3%, and the inference speed has increased by 15FPS. Compared with the object detection model CenterNet, the parameter has decreased by 28.39×10^6, the computational complexity has decreased by 60.9GFLOPs, and the mean average precision has increased by 17.57%, and the inference speed has increased by 41FPS.

In summary, the proposed improved algorithm in this paper has certain advantages in both speed and precision compared with

TABLE 6 Ablation experiment.

Order Number	Param(M)	GFLOPS(G)	Map(%)	FPS
1	7.03	16.0	84.0	64
2	4.59	12.0	83.14	80
3	4.90	10.6	83.26	72
4	3.99	8.5	82.94	88
5	4.31	9.3	84.24	88
6	4.31	9.3	85.10	85

TABLE 7 Comparative experiments on mainstream object detection algorithms.

Algorithm	Param(M)	GFLOPS(G)	mAP(50%)	FPS
SSD	26.3	62.8	79.08	23
Faster-RCNN	137.0	370.0	71.77	10
RetinaNet	37.97	170.1	58.28	23
YOLOv3	61.9	66.2	72.83	25
YOLOv4-tiny	6.1	7.0	63.26	55
YOLOv7-tiny	6.0	13.2	83.8	70
CenterNet	32.7	70.2	67.53	44
Ours	4.31	9.3	85.1	85

different object detection algorithms. The algorithm lays the foundation for deployment to underwater equipment. Meanwhile, the proposed improved algorithm improves the impact of low underwater image quality on object detection. It can perform high-precision object detection in blurry underwater images.

In order to more intuitively reflect the detection performance of the algorithm proposed in our methods, two representative images were selected, which respectively represent the cases of underwater small object aggregation and underwater image blurring. The detection results were compared with other algorithms, as shown in Figure 11. From Figure 11, it can be observed that other algorithms have more or less missed detections and false detections in the case of small objects aggregation or image blurring. The algorithm proposed in this article greatly reduces the occurrence of missed detections and false detections by using RepVGG modules and ECA attention mechanisms. This further proves that the proposed algorithm model can be applied to complex underwater environment with high speed and lighter weight.

5 Conclusions

This article proposes a lightweight marine biological object detection method based on YOLOv5, which solves the problem of slow detection speed caused by large model parameters and computational complexity, and improves the detection accuracy of small targets in underwater fuzzy images. The algorithm is based on YOLOv5s and replaces conventional convolution with depth separable convolution, and introduces a C3 module based on Ghost convolution to reduce the number of parameters and computation complexity and improve the inference speed. The RepVGG module is introduced to enhance the model's feature extraction capability while maintaining high detection speed during inference. Embedding the ECA attention mechanism to the RepVGG module improves the detection precision of small underwater object in blurry images. We conducted experiments on the URPC dataset, and the improved model significantly reduces the number of parameters and computational complexity compared to the original model, while increasing the mAP by 1.1% compared to

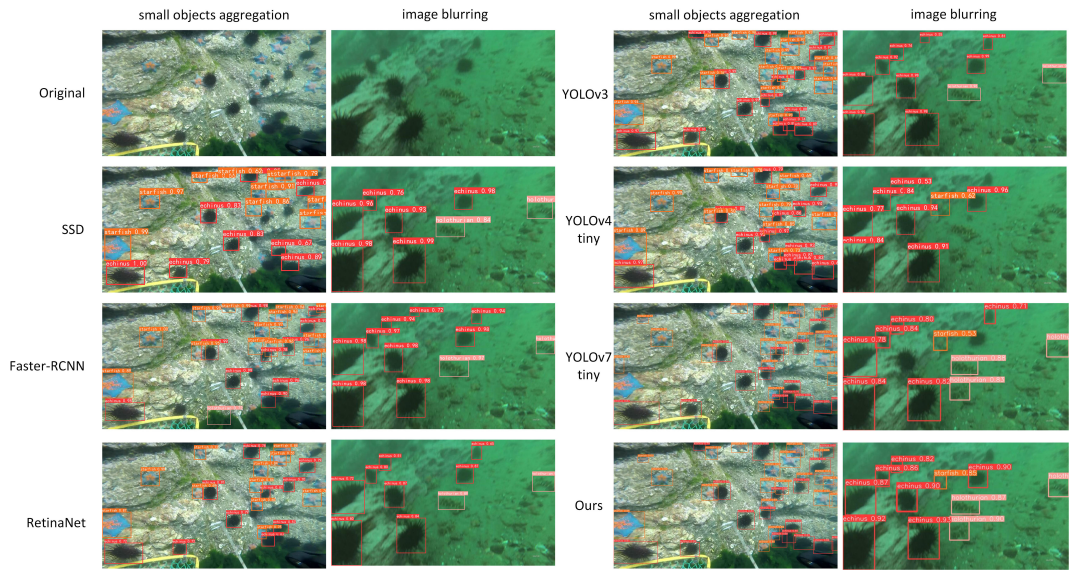

FIGURE 11
detection results of different algorithms.

the original model. We also conducted comparative experiments with other advanced algorithms, which demonstrated that our proposed algorithm is better suited to complex underwater environment than other algorithms.

Due to the complexity of the underwater environment, low and uneven underwater illumination, large amounts of suspended matter in the water, and the influence of weather, season, and sampling location, the captured underwater images are often blurry, of poor quality, and have significant color changes. Therefore, indepth research on image restoration methods is required to further improve the accuracy of object detection in underwater environments.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material. Further inquiries can be directed to the corresponding author.

Author contributions

HL, TS Designed research methods. HL wrote the original draft and organized database. TS Reviewed and edited the article. All authors contributed to manuscript revision, read, and approved the submitted version.

Funding

This research was funded by the 2019 Shandong Provincial Key R&D Program Project (grant number: 2019GGX101047).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Banan, A., Nasiri, A., and Taheri-Garavand, A. (2020). Deep learning-based appearance features extraction for automated carp species identification. *Aquac. Eng.* 89, 102053. doi: 10.1016/j.aquaeng.2020.102053

Bazeille, S., Quidu, I., and Jaulin, L. (2007). "Identification of underwater man-made object using a colour criterion," in *Proceedings of the Conference on detection and classification of underwater targets*, Edinburgh, UK. 18–19.

Bochkovskiy, A., Wang, C. Y., and Liao, H. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv*. doi: 10.48550/arXiv.2004.10934

Bonofiglio, F., De, L. F., Yee, C., Chatzievangelou, D., Aguzzi, J., and Marini, S. (2022). Machine learning applied to big data from marine cabled observatories: A case study of sablefish monitoring in the NE Pacific. *Front. Mar. Sci.* 9. doi: 10.3389/fmars.2022.842946

Chollet, F. (2017). "Xception: deep learning with depthwise separable convolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA. 1800–1807. doi: 10.1109/CVPR.2017.195

Chuang, M.-C., Hwang, J.-, and Williams, K. (2016). A feature learning and object recognition framework for underwater fish images. *IEEE Trans. Image Proc.* 25, 1862–1872. doi: 10.1109/TIP.2016.2535342

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. (2021). "Repvgg: Making vgg-style convnets great again," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13733–13742. doi: 10.1109/CVPR46437.2021.01352

Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). "CenterNet: Keypoint triplets for object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 6568–6577. doi: 10.1109/ICCV.2019.00667

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *Int. J. Comput. vision.* 88 (2), 303–338. doi: 10.1007/s11263-009-0275-4

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recogn. Lett.* 27, 861–874. doi: 10.1016/j.patrec.2005.10.010

Girshick, R. (2015). "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile. 1440–1448. doi: 10.1109/ICCV.2015.169

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA. 580–587. doi: 10.1109/CVPR.2014.81

Guan, Y., Liu, X., Yu, Z., Wang, Y., Zheng, X., Zhang, S., et al. (2023). Fast underwater image enhancement based on a generative adversarial framework. *Front. Mar. Sci.* 9. doi: 10.3389/fmars.2022.964600

Han, K., Wang, Y. H., Tian, Q., Guo, J., Xu, C., and Xu, C. (2020). "Ghostnet: More Features from cheap operations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA. 1580–1589. doi: 10.1109/CVPR42600.2020.00165

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 386–397. doi: 10.1109/TPAMI.2018.2844175

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778. doi: 10.1109/CVPR.2016.90

Hoeser, T., and Kuenzer, C. (2020). Object detection and image segmentation with deep learning on Earth observation data: A review-part I: evolution and recent trends. *Remote Sens.* 12, 1667. doi: 10.3390/rs12101667

Hou, Q., Zhou, D., and Feng, J. (2021). "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Nashville, TN, USA. 13708–13717. doi: 10.1109/CVPR46437.2021.01350

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-Excitation Networks. *2018 IEEE/CVF 545 Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, 7132–7141. doi: 546 10.1109/CVPR.2018.00745

Huang, A., Zhong, G., Li, H., and Choi, D. (2022). "Underwater object detection using restructured SSD," in *Proceedings of the CAAI International Conference on Artificial Intelligence*, Singapore. (Cham: Springer) 526–537. doi: 10.1007/978-3-031-20497-5_43

Liang, P., Dong, P., Wang, F., Ma, P., Bai, J., and Wang, B. (2022). Learning to remove sandstorm for image enhancement. *Vis. Comput.* 39, 1–24. doi: 10.1007/s00371-022-02448-8

Lin, T. Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). *Feature Pyramid Networks for Object Detection* (Washington, DC, USA: IEEE Computer Society). doi: 10.1109/CVPR.2017.106

Lin, T. Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*. 2980–2988. doi: 10.1109/ICCV.2017.324

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). "Microsoft coco: Common objects in context," in *European Conference on Computer*

Vision (Germany: Springer: Berlin/Heidelberg), 740–755. doi: 10.1007/978-3-319-10602-1_48

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., et al. (2016). "Ssd: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands. 21–37. doi: 10.1007/978-3-319-46448-0_2

Liu, S., Qi, L., Qin, H., Shi, J., and Jia, J. (2018). "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA. 8759–8768. doi: 10.1109/CVPR.2018.00913

Lopez-Vazquez, V., Lopez-Guede, J. M., Chatzievangelou, D., and Aguzzi, J. (2023). Deep learning based deep-sea automatic image enhancement and animal species classification. *J. Big Data* 10, 37. doi: 10.1186/s40537-023-00711-w

Marini, S., Fanelli, E., Sbragaglia, V., Azzurro, E., Joaquin Del Rio, F., and Aguzzi, J. (2018). Tracking fish abundance by underwater image recognition. *Sci. Rep.* 8, 13748. doi: 10.1038/s41598-018-32089-8

Qin, H., Li, X., Liang, J., Peng, Y., and Zhang, C. (2016). DeepFish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing* 187, 49–58. doi: 10.1016/j.neucom.2015.10.122

Redmon, J., and Farhadi, A. (2018). YOLOv3: an incremental improvement. *arXiv*. doi: 10.48550/arXiv.1804.02767

Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1137–1149. doi: 10.1109/TPAMI.2016.2577031

Sahoo, A., Dwivedy, S. K., and Robi, P. (2019). Advancements in the field of autonomous underwater vehicle. *Ocean Eng.* 181, 145–160. doi: 10.1016/j.oceaneng.2019.04.011

Shafiq, M., and Gu, Z. (2022). Deep residual learning for image recognition: A survey. *Appl. Sci.* 12, 8972. doi: 10.3390/app12188972

Shi, X., and Wang, H. (2023). Improved lightweight underwater target detection network based on YOLOV4 (you only look once v4). *J. Harbin Eng. University*. 44 (01), 154–160. doi: 10.11990/jheu.202111022

Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv*. doi: 10.48550/arXiv.2207.02696

Wang, J., Wei, J., Mei, S., and Wang, J. (2021). Improved YOLOv3 for small object detection in remote sensing images. *Comput. Eng. Applications.* 57 (20), 133–141. doi: 10.3778/j.issn.1002-8331.2012-0064

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Piscataway, NJ. 11531–11539. doi: 10.1109/CVPR42600.2020.01155

Woo, S., Park, J., Lee, J.-Y., and Kweon, I. S. (2018). "CBAM: convolutional block attention module," in *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, vol. 11211 . Eds. V. Ferrari, M. Hebert, C. Sminchisescu and Y. Weiss (Cham: Springer). doi: 10.1007/978-3-030-01234-2_1

Zaidi, S. S. A., Ansari, M. S., Aslam, A., Kanwal, N., Asghar, M., and Lee, B. A. (2021). A survey of modern deep learning based object detection models. *SAN DIEGO, USA. Digital Signal Process* 126, 1033514. doi: 10.1016/j.dsp.2022.103514

Zhao, Z., Zheng, P., Xu, S., and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Trans. Neural Networks Learn. Syst.* 30, 3212–3232. doi: 10.1109/TNNLS.2018.2876865