



# OPEN Lightweight enhanced YOLOv8n underwater object detection network for low light environments

Jifeng Ding<sup>1✉</sup>, Junquan Hu<sup>1,2</sup>, Jiayuan Lin<sup>1,2</sup> & Xiaotong Zhang<sup>1,2</sup>

In response to the challenges of target misidentification, missed detection, and other issues arising from severe light attenuation, low visibility, and complex environments in current underwater target detection, we propose a lightweight low-light underwater target detection network, named PDSC-YOLOv8n. Firstly, we enhance the input images using the improved Pro MSRCR algorithm for data augmentation. Secondly, we replace the traditional convolutions in the backbone and neck networks of YOLOv8n with Ghost and GSConv modules respectively to achieve lightweight network modeling. Additionally, we integrate the improved DCNv3 module into the C2f module of the backbone network to enhance the capability of target feature extraction. Furthermore, we introduce the GAM into the SPPF and incorporate the CBAM attention mechanism into the last layer of the backbone network to enhance the model's perceptual and generalization capabilities. Finally, we optimize the training process of the model using WIoUv3 as the loss function. The model is successfully deployed on an embedded platform, achieving real-time detection of underwater targets on the embedded platform. We first conduct experiments on the RUOD underwater dataset. Meanwhile, we also utilized the Pascal VOC2012 dataset to evaluate our approach. The mAP@0.5 and mAP@0.5:0.95 of the original YOLOv8n algorithm on RUOD dataset were 79.6% and 58.2%, respectively, and the PDSC-YOLOv8n algorithm mAP@0.5 and mAP@0.5:0.95 can reach 86.1% and 60.8%. The number of parameters of the model is reduced by about 15.5%, the detection accuracy is improved by 6.5%. The original YOLOv8n algorithm was 73% and 53.2% mAP@0.5 and mAP@0.5:0.95 on the Pascal VOC dataset, respectively. The PDSC-YOLOv8n algorithm mAP@0.5 and mAP@0.5:0.95 were 78.5% and 57%, respectively. The superior performance of PDSC-YOLOv8n indicates its effectiveness in the field of underwater target detection.

**Keywords** Underwater target detection, Low-light, Deformable convolution, Attention mechanisms

With the development of society, terrestrial resources are becoming increasingly insufficient to meet the needs of human production and daily life<sup>1</sup>. Therefore, the exploration and exploitation of marine resources have become more and more important. Underwater target detection involves the identification and localization of various targets in the underwater environment, such as marine life, shipwrecks, and resources like oil, which can be utilized for the development and management of marine resources<sup>2</sup>. Traditional underwater target detection primarily relies on manual underwater environment assessment. But this method is time-consuming, labor-intensive, and poses high risks, and is no longer applicable. With the continuous development of target detection technology in deep learning, intelligent robots equipped with target detection algorithms have gradually replaced manual detection methods for underwater environments.

However, in underwater environments, there are significant fluctuations in light and uneven distribution of light intensity. This non-uniformity leads to some areas being darker while others are brighter, resulting in differences in the visibility of targets at different locations<sup>3</sup>.

With increasing water depth and the influence of factors such as suspended particles, visible light gradually diminishes, resulting in numerous low-light regions. Under these low-light conditions, underwater targets typically exhibit a blurry and distorted appearance, making it difficult for detection models to correctly identify the shape, contours, and textures of the targets<sup>4</sup>. This visual blurring increases the possibility of erroneous target detection, especially in complex underwater scenes, making target detection more challenging<sup>5,6</sup>. The YOLO

<sup>1</sup>College of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, China. <sup>2</sup>These authors contributed equally: Junquan Hu, Jiayuan Lin and Xiaotong Zhang. ✉email: 202211051048@stu.dlnu.edu.cn

(You Only Look Once) series of target detection algorithms, renowned for their detection accuracy, are indeed the preferred choice for underwater target detection.

In this work, we propose a lightweight low-light underwater target detection network, named PDSC-YOLOv8n, based on an improved YOLOv8n. The main contributions are as follows:

- A new data augmentation method, the Pro MSRCR(Pro Multi-Scale Retinex with Color Restoration) algorithm, is proposed, which is added at the input end of the model for image preprocessing. This algorithm enhances each of the three color channels of the input blurry underwater images separately to improve the color and clarity of the target images.
- A new DCNv3(Deformable Convolutional Networks v3) module is proposed and added to the C2f(CSP Bottleneck with 2 Convolutions) module of the backbone network to enhance the feature extraction capability of target regions. Additionally, Conv(traditional convolutions) in the YOLOv8n backbone and neck networks are replaced with Ghost(Ghost Convolution) and GSConv(Generalized Split Convolution) convolutional modules, respectively, to achieve model lightweighting.
- A new SPPF-G module is proposed, incorporating the GAM(Global Attention Mechanism) attention mechanism into the SPPF(Spatial Pyramid Pooling-Fast) module, to enhance its focus on features at different scales. Additionally, the CBAM(Convolutional Block Attention Module) attention mechanism is introduced into the last layer of the backbone network to enhance the model's perception and generalization capabilities.
- The WIoUv3 (Wise-IoUv3) loss function is introduced as a new loss function to enhance the prediction accuracy of bounding boxes, accelerating network convergence, optimizing the model training process, and deploying the proposed model on embedded devices for validation.

Experimental results demonstrate that our proposed detection model exhibits superior performance.

### Related work

Currently, YOLOv8 is one of the top-performing models in the field of computer vision, renowned for its exceptional accuracy and speed in object detection, which far surpasses other algorithms<sup>7</sup>. In recent years, underwater target detection has emerged as a popular research direction, with many researchers dedicated to applying object detection algorithms to the underwater tasks. Some researchers are devoted to improving the network models of algorithms for underwater target detection. Cheng et al.<sup>8</sup> utilized a diffusion model to generate underwater acoustic data and employed an augmented dataset to train an enhanced YOLOv7 model for underwater small-sample object detection. Their proposed enhanced YOLOv7 model achieved a 2% increase in mAP@0.5 on underwater datasets. Huo et al.<sup>9</sup> introduced an underwater biological detection model called G-YOLOv5, based on the principle of feature reuse. They designed an Improved-Ghost module with feature reuse. Achieving a mAP@0.5 of 74.2% on the 2021URPPC dataset, it demonstrates certain practical applicability. Zhao et al.<sup>10</sup> proposed the YOLOv7-CHS model for underwater target detection. This model replaces the Efficient Layer Aggregation Network (ELAN) with the High-Order Spatial Interaction (HOSI) module as the backbone of the model, and integrates the Context Transformer (CT) module into the model head, effectively enhancing the model's capability to detect small targets. On the DUO dataset, the model achieved higher average precision, with improvements of 4.5% and 4.2%, respectively. Additionally, real-time detection speed of 32 frames per second was achieved.

Some researcher's incorporate attention mechanism into YOLO framework to improve the detection performance. Chen et al.<sup>11</sup> improved YOLOv5 by introducing the multi-head self-attention (MHSA) mechanism into its backbone network. They further implemented the algorithm onto unmanned underwater vehicles, enabling the recognition and tracking of detected underwater targets. Bao et al.<sup>12</sup> utilized the High-Resolution Network (HRNet) to enhance target feature representation and improved the attention module (A-CBAM) to capture complex feature distributions. On the URPC2020 and URPC2018 datasets, they achieved detection accuracies of 77.02% and 80.9%, respectively, demonstrating the effectiveness of the proposed network. Ge et al.<sup>13</sup> improved the detection accuracy of YOLOv5s by incorporating Coordinate Attention (CA) modules and Squeeze-and-Excitation (SE) modules. They conducted experiments on the data from the 2019 China Underwater Robot Contest, and the average detection accuracy of the enhanced YOLOv5s network increased by 2.4%. Tang et al.<sup>14</sup> proposed a Skip Residual C3 (SRC3) module, applied in the YOLOv5 framework, and introduced the CBAM attention module for target feature extraction, effectively improving the detection accuracy of small underwater targets. Wang et al.<sup>15</sup> employed FasterNet-L as the backbone network model in the YOLOv7 framework. Additionally, they introduced the lightweight cross-modal Transformer Attention (CoTA tention) mechanism to enhance the extraction of target features, achieving excellent detection accuracy. Some researchers are also dedicated to improving certain convolutional modules to enhance the accuracy of underwater target detection algorithms. Zeng et al.<sup>16</sup> conducted research on the detection of underwater sound waves using millimeter-wave radar. They utilized the Continuous Wavelet Transform method for underwater target detection, achieving the capability to recognize underwater targets. Rakesh Joshi<sup>17</sup> proposed a time-domain signal detection algorithm based on integrated dual-functional deep learning for underwater target detection and classification, as well as three-dimensional integrated imaging (InIm) under degraded conditions. They utilized reconstructed 3D data as input for the YOLOv4 neural network for target detection, thereby improving the detection accuracy of the 3D InIm network. Kai et al.<sup>18</sup> proposed an underwater target detection network called G-Net, which integrates the use of the FML (Foggy Module Learning) module, capable of effectively extracting defogging features. Wang et al.<sup>19</sup> proposed an IG-YOLOv5 model, which utilizes the Improved-Ghost module to reconstruct the CSPDarknet structure of YOLOv5 and introduces the optimized WIoUv3 as the loss function. The model performed well on the 2021 URPPC dataset. Fu et al.<sup>20</sup> replaced the MP module in the neck network with an SPD-MP module structure and introduced a new NWD loss function. They proposed an enhanced YOLOv7

network for underwater target detection. With a mAP@0.5 value of 82.3% on the URPC dataset, it is suitable for underwater marine biological target detection. Zhu et al.<sup>21</sup> enhanced the feature extraction capability of the YOLOv8 backbone network using the InceptionNeXt module, achieving a detection accuracy of 77.9% for marine organisms. Liu et al.<sup>22</sup> proposed an improved YOLOv5-Lite model, which dynamically adopts transformer modules and utilizes distillation techniques to optimize the algorithm. Compared to YOLOv5, this model achieves a 6.6% increase in detection accuracy and is capable of performing underwater target detection efficiently. Wang et al.<sup>23</sup> optimized the network model in YOLOv5s by incorporating the CFnet structure. Additionally, they introduced a small object detection (SD) layer, proposing a novel detection model named YOLOv5-FCDSSE, which demonstrates excellent detection performance for underwater small targets. Chen et al.<sup>24</sup> replaced some ordinary convolutions in the YOLOv7 backbone network with PConv convolutions to reduce the parameter amount. Guo et al.<sup>25</sup> designed a lightweight YOLOv8 network using the FasterNet module and improved the C2f module. On the UOD and UTDAC2020 datasets, the mAP@0.5 values were 84.7% and 85.2%, respectively, demonstrating excellent detection performance.

The above research indicates that in the field of underwater target detection, the YOLO series algorithms perform excellently. Therefore, optimizing these algorithms to improve the accuracy of underwater target detection is crucial.

## Proposed methods

### Network model

YOLOv8 adopts a framework similar to YOLOv5, consisting of a backbone network, neck network, and head network. However, it modifies the C2f module, which consists of convolutional layers, batch normalization layers, activation functions, and cross-stage parts<sup>26</sup>. The C2f module utilizes residual connections and cross-stage feature fusion to effectively extract image features, thereby improving its performance and efficiency. Additionally, YOLOv8 employs an anchor-free model, allowing each branch to focus on its specific task<sup>27</sup>.

YOLOv8 employs the Complete Intersection Over Union (CIoU) and Distribution Error Loss (DEL) as its loss functions, and binary cross-entropy for classification. Figure 1 illustrates the YOLOv8 network model.

To address the challenges of underwater target detection, particularly in detecting blurred targets, we propose a lightweight low-light underwater target detection algorithm, named PDSC-YOLOv8n, based on the improved YOLOv8n.

First, we enhance the input image by applying the improved Pro MSRCR algorithm at the input of the model. Next, YOLOv8n adopts the Ghost and GConv convolutional modules for its backbone and neck networks, respectively, achieving lightweighting of the network model. Additionally, we integrate the improved DCNv3 module into the C2f module of the backbone network to enhance target feature extraction capabilities. Then, we introduce the GAM into the SPPF module to enhance the model's focus on features of different scales, improving its ability to recognize detected targets. Furthermore, we introduce the CBAM attention mechanism into the last layer of the backbone network to enhance model perception and generalization capabilities. Finally, we utilize WIoUv3 as the loss function to accelerate network convergence and optimize the model training process. The improved PDSC-YOLOv8n network model is shown in Fig. 2.

The proposed PDSC-YOLOv8n network primarily preprocesses the dataset at the input module to enhance the information of input image. In the backbone section, it focuses on extracting features from the images, followed by fusion processing in the neck part. The head section primarily detects objects in the fused feature map to output the detection results<sup>28</sup>.

## Module architecture

### Data augmentation module

Image data augmentation is a key technology for improving the clarity of underwater images<sup>29</sup>. Most existing image enhancement methods involve transforming and distorting original images to generate new training samples, thereby expanding the training dataset. While such strategy helps models better learn image features, the training time and cost increase as the training dataset grows<sup>30</sup>. In our work, we integrate an improved Pro MSRCR algorithm at the input of the network model. By preprocessing the input underwater target images and enhancing the R(Red), G(Green), and B(Blue) color channels, we aim to improve the clarity of the target images. This approach not only saves training costs but also reduces training time.

Firstly, the Pro MSRCR algorithm is an improved version of the MSRCR<sup>31</sup> algorithm. The MSRCR algorithm aims to restore color-distorted and poorly defined images by introducing color restoration factors, thereby achieving image data enhancement. The inference formula for MSRCR is as follows:

$$R_M(x, y) = \exp \left( \sum_{i=1}^r w_i \cdot \log(L(x, y) \cdot G_i(x, y) + 1) \right) \cdot \left( \frac{L(x, y)}{\sum_{i=1}^r w_i \cdot \log(L(x, y) \cdot G_i(x, y) + \mu)} \right) \eta \cdot (a(x, y)^\eta, b(x, y)^\eta)$$

where  $r$  represents the number of scales, typically set to 3.  $R_M(x, y)$  stands for MSRCR algorithm,  $w_i$  denotes the weights,  $L(x, y)$  represents the logarithm of image brightness,  $G_i(x, y)$  represents the Gaussian filter,  $\mu$  denotes a small positive number used to avoid division by zero.  $a(x, y)$  and  $b(x, y)$  represent the color components of the image, and  $\eta$  represents the color balance parameter.

The MSRCR algorithm adopts the Retinex theory, where Gaussian filtering is applied to the logarithmic domain of the image to compute the Gaussian weighted average of pixels in the neighborhood of the center point.

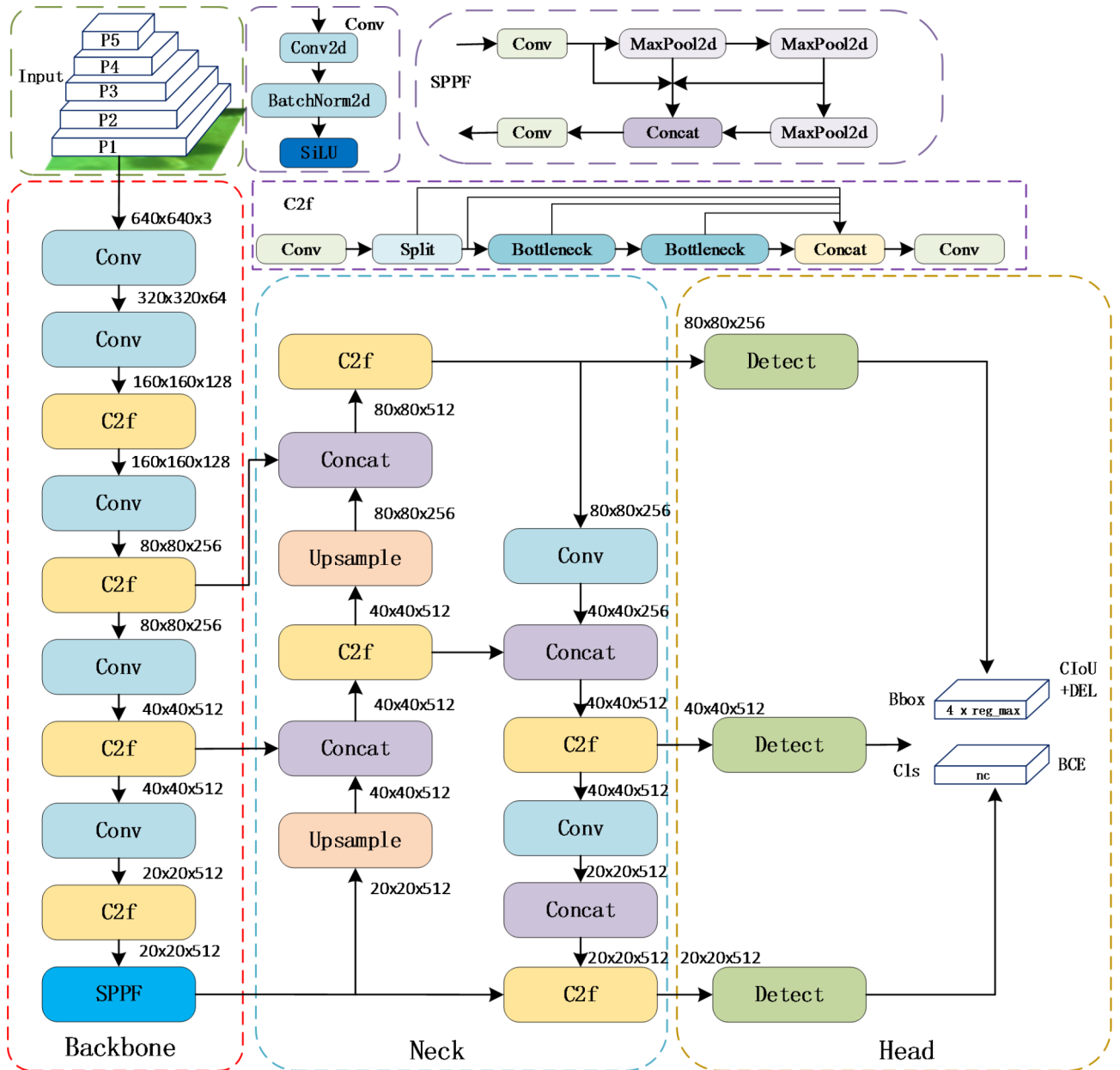


Fig. 1. Structure of YOLOv8.

This process enhances details in the image. However, this method does not consider the similarity between pixels or whether pixels are located at boundaries. It would lead to blurred boundary information, thereby affecting the extraction of image feature and reducing detection accuracy.

The Pro MSRCR algorithm combines bilateral filtering with the MSRCR algorithm for image processing<sup>32</sup>. It can further remove noise, refine image feature, and enhance the accuracy of target detection while maintaining clear image boundaries. The computational formula for the Pro MSRCR algorithm is as follows:

$$R_{PM}(x, y) = \exp \left( \sum_{i=1}^r w_i \cdot \log(L(x, y) \cdot G_i(x, y) + 1) \right) \cdot \left( \frac{\log(L(x, y) + 1)}{\sum_{i=1}^r w_i \cdot \log(L(x, y) \cdot G_i(x, y) + \mu) \cdot \exp \left( -\frac{(I(x, y) - I(x', y'))^2}{2 \cdot \sigma^2} \right)} \right) \cdot (a(x, y)^\gamma, b(x, y)^\gamma)$$

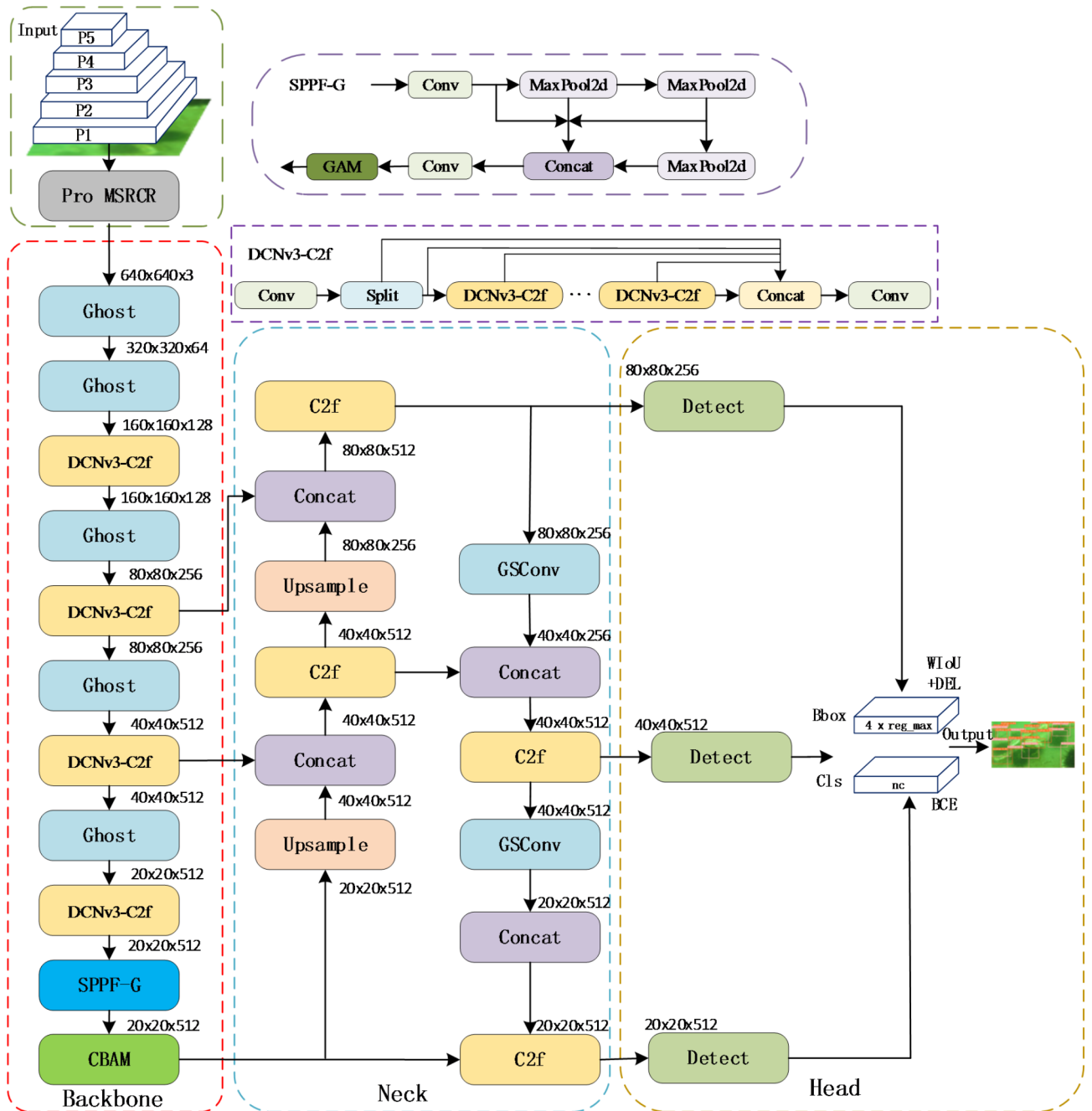


Fig. 2. Architecture of PDSC-YOLOv8n network.

where  $\exp\left(-\frac{(I(x,y)-I(x',y'))^2}{2\cdot\sigma^2}\right)$  represents the weight function of bilateral filtering.  $R_{PM}(x, y)$  represents the improved Pro MSRCR algorithm,  $\gamma$  represents the additional color balance parameter introduced by the Pro MSRCR algorithm, and  $\sigma$  represents the spatial domain standard deviation used to calculate the weights in bilateral filtering.

The Pro MSRCR algorithm introduces bilateral filtering to maintain the relative spatial relationships of image pixels. Furthermore, the bilateral filtering component can adjust weights based on the brightness differences between pixels. Additionally, an extra color balance parameter is introduced to further adjust the color balance of the enhanced image. This approach helps to alleviate issues such as overly dark original images and excessive enhancement caused by the MSRCR algorithm. As a result, the boundaries and feature information of target images become more prominent<sup>33</sup>. The enhancement results of Pro MSRCR and MSRCR are shown in Fig. 3, which demonstrates the effectiveness of the Pro MSRCR algorithm.

Figure 3a displays the original images from the dataset, Fig. 3b shows the image processed by the MSRCR algorithm, which reveals issues such as color imbalance, over-enhancement, and blurriness. In contrast, in Fig. 3c, the images have undergone processing with the Pro MSRCR method. The Pro MSRCR introduces



Fig. 3. Comparison of the proposed Image Enhancements method and MSRCR.

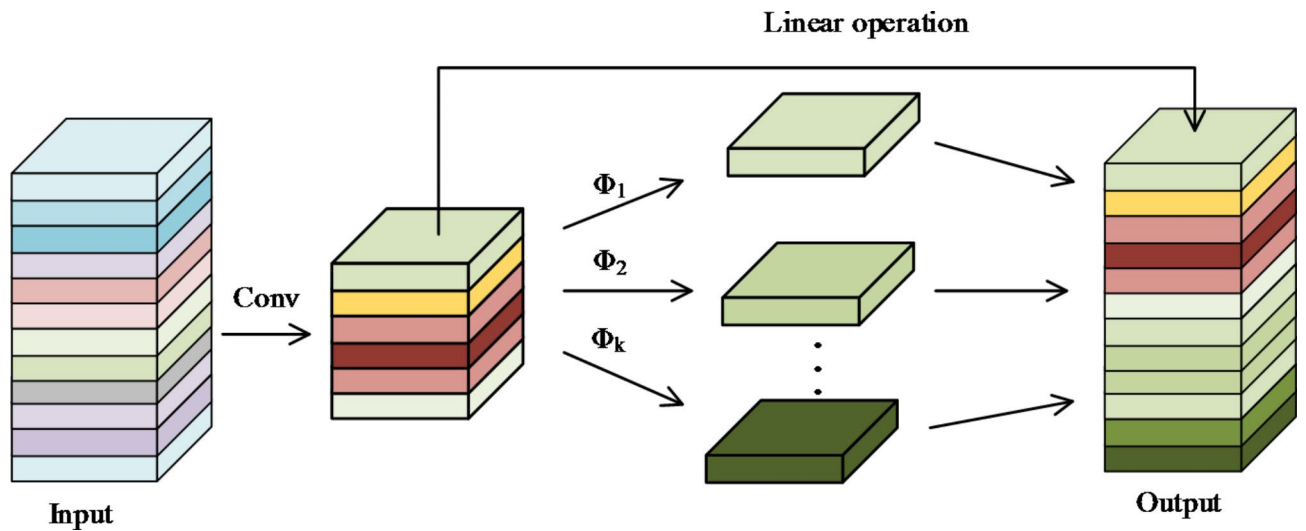


Fig. 4. Structure of the Ghost module.

additional color balance parameters, addressing both the issue of excessive enhancement caused by the MSRCR algorithm and the problem of color imbalance in the original images. It compensates for boundary defects, refines the feature information of target images, and significantly improves image clarity, then, the enhanced images are fed into the backbone network of the model for further processing.

**Ghost convolution module**

Underwater target detection is primarily applied in embedded underwater devices. Thus, the model must meet the operational requirements of embedded hardware devices. In the YOLOv8n model, traditional convolutional layers primarily extract feature information by convolving across all channels of the input images. However, it contains a large amount of redundant information, resulting in an oversized model, making it difficult to deploy on embedded devices. In our work, Ghost convolution is used to replace the Conv in the YOLOv8n backbone network. Ghost convolution first generates  $m$  original feature maps using a small number of traditional convolutions, and then utilizes these  $m$  original feature maps to generate  $s$  Ghost feature maps through linear operations<sup>34</sup>. Ghost convolution ultimately outputs  $n = m \cdot s$  feature maps,  $\Phi$  represents the  $k$ -th linear convolution, as shown in Fig. 4.

When outputting  $n$  feature maps simultaneously, the number of network parameters for traditional convolution and Ghost convolution are denoted by  $F_1$  and  $F_2$ , respectively, as calculated in Eqs. (3) and (4).

$$F_1 = n \cdot c \cdot k \cdot k \tag{3}$$

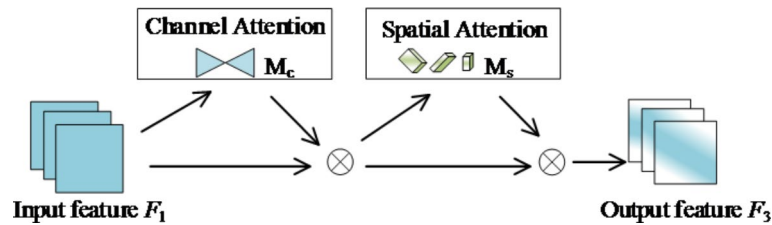


Fig. 5. Schematic diagram of the GAM module.

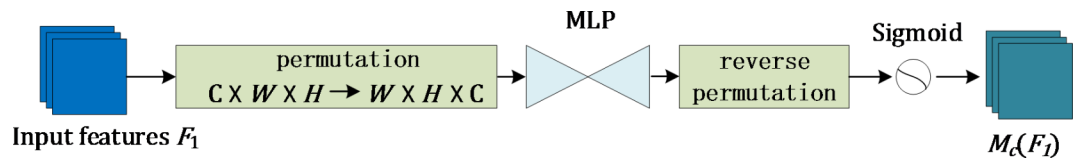


Fig. 6. Channel attention submodule.

$$F_2 = \frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d \quad (4)$$

The ratio of parameters between  $F_1$  and  $F_2$  is represented by Eq. (5).

$$\frac{F_1}{F_2} = \frac{n \cdot c \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot k \cdot k + (s-1) \cdot \frac{n}{s} \cdot d \cdot d} \approx \frac{s \cdot c}{s+c-1} \approx S \quad (5)$$

The computational complexity of models using traditional convolution and Ghost convolution is denoted by  $F'_1$  and  $F'_2$ , respectively, as shown in Eqs. (6), (7).

$$F'_1 = n \cdot c \cdot h' \cdot w' \cdot k \cdot k \quad (6)$$

$$F'_2 = \frac{n}{s} \cdot c \cdot h' \cdot w' + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d \quad (7)$$

The ratio of computational complexity between them can be expressed as Eq. (8).

$$\frac{F'_1}{F'_2} = \frac{n \cdot c \cdot h' \cdot w' \cdot k \cdot k}{\frac{n}{s} \cdot c \cdot h' \cdot w' + (s-1) \cdot \frac{n}{s} \cdot h' \cdot w' \cdot d \cdot d} \approx \frac{s \cdot c}{s+c-1} \approx s \quad (8)$$

Here,  $c$  represents the number of channels in the input image, and  $s$  is significantly smaller than  $c$ ,  $k \cdot k$  represents the kernel size implemented by traditional convolution,  $h'$  and  $w'$  respectively denote the height and width of the original feature maps generated by Ghost convolution, and  $d \cdot d$  represents the kernel size of the linear operation, when  $k = d$ , Ghost convolution occupies only  $1/s$  of the parameters and computational complexity of traditional convolution. Therefore, using Ghost convolution can make the model more lightweight.

### The SPPF-G module

In the YOLOv8 network, the SPPF module extracts features from different receptive fields by performing spatial pyramid pooling at various scales on the images processed by the convolutional layers. This helps the model capture features of objects at different scales, thereby enhancing detection accuracy. However, the SPPF module may lose some detailed information when performing spatial pyramid pooling for complex backgrounds or low-resolution objects, which poses limitations<sup>35</sup>.

To address this issue, we introduce the GAM attention mechanism in SPPF. The structure of the GAM attention mechanism is shown in Fig. 5.

GAM attention mechanism is a global attention mechanism, which aims to effectively reduce the dispersion of information in the global dimension and enhance the interaction between global features. By introducing the sequential channel-space attention mechanism, it endows the model with stronger adaptability and expressiveness. For the input feature map, firstly, the dimension transformation is carried out, and the feature map after the dimension transformation is input to MLP (Multi-Layer Perceptron), and then transformed into the original dimension, and the Sigmoid processing is carried out to output, where  $M_c(F_1)$  represents the feature map after dimension conversion, as shown in Fig. 6.

After convolution, the number of channels is reduced by convolution with a convolution kernel of 7, and the calculation amount is reduced. After a convolution operation with a convolution kernel of 7, the number of channels is increased and the number of channels is kept consistent. Finally, Sigmoid outputs the feature map after convolution processing,  $M_s(F_2)$  represents the output result after convolution processing, as shown in Fig. 7.

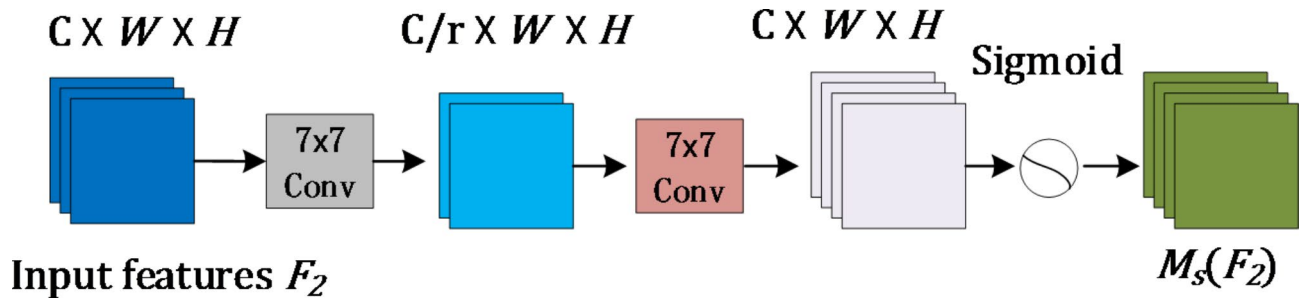


Fig. 7. Channel attention submodule.

The GAM attention mechanism can enhance the focus of the SPPF module on features of different scales, thereby improving the model's recognition capability of objects. Moreover, GAM typically requires only a small number of parameters to focus on global features. Combining GAM with the SPPF module can enhance model performance without significantly increasing the number of parameters. Figure 8 illustrates the schematic diagram of the improved SPPF-G module.

Combining the GAM attention mechanism with the SPPF module can enhance the model's detection ability for objects at different scales and strengthen its focus on global features, thus improving the accuracy of target detection.

### The DCNv3-C2f module

In the YOLOv8 backbone network, C2f is a feature extraction module responsible for extracting target features from the images processed by the convolutional layers. However, in underwater scenarios with multiple targets, dim lighting conditions, and objects of various scales, there may be instances of missing target feature extraction. To overcome this challenge, we introduce an improved DCNv3<sup>36</sup> into the C2f module, allowing the model to better adapt to the structure and size of objects, thereby enhancing robustness, especially in dim underwater environments<sup>37</sup>.

Firstly, DCNv3 is a type of deformable convolution, which differs from traditional convolutions in several ways. DCNv3 initially adopts the idea of separable convolution, separating the original convolution weights into depth-wise and point-wise components, thus achieving weight sharing among convolutional neurons. Secondly, DCNv3 introduces a multi-branch mechanism, enhancing the network's feature learning capability by introducing multiple branches and parameter groups<sup>38</sup>. This enables the network to more effectively capture target features and achieve better performance in complex scenes. Finally, by using  $k$  modulation scalars to change sigmoid normalization to softmax normalization, where the sum of modulation scalars is constrained to 1, the training process of the model at different scales becomes more stable.

The inference process of DCNv3 is represented by Eq. (9).

$$y(P_0) = \sum_{g=1}^G \sum_{k=1}^K W_g m_{gk} X_g(P_0 + P_k + \Delta P_{gk}) \quad (9)$$

Here,  $G$  represents the total number of aggregation groups,  $K$  represents the total number of sampling points,  $k$  represents the enumerated sampling points,  $W_g$  represents the position-independent projection weight of the group,  $m_{gk}$  represents the offset of the  $k$ -th sampling point in the  $g$ -th group,  $X_g$  represents the sliced feature map, and  $\Delta P_{gk}$  is the offset corresponding to the grid sampling position  $P_k$  in the  $g$ -th group. Traditional DCNv3, to some extent, can improve the model's receptive field and its ability to model object deformations.

However, it is challenging to extract features from some blurry targets. Therefore, we apply the SE attention mechanism before the convolutional output of DCNv3. First, we compute the global average pooling in the SE attention mechanism. Then, we use a small *MLP* to learn the importance of each channel, resulting in an activation vector, as shown in Eqs. (10), (11).

$$Z = AvgPool(X) \quad (10)$$

$$f = ReLU(FC(Z)) \quad (11)$$

where  $Z$  represents the feature map obtained through global average pooling,  $f$  denotes the activation vector, *FC* is the fully connected layer, and *ReLU* is the activation function. Next, we scale the activation vector to obtain attention weights. Then, we reshape the attention weights to the same shape as the feature map, and apply them to the original feature map by weighting the original feature map according to the attention weights, as shown in Eqs. (12), (13).

$$s = Sigmoid(FC(f)) \quad (12)$$

$$X_{SE} = X \cdot Reshape(s, (N, C, 1, 1)) \quad (13)$$

where  $X_{SE}$  represents the weighted feature map,  $s$  denotes the attention weights, *Reshape*( $s, (N, C, 1, 1)$ ) reshapes the attention weights to the same shape as the feature map,  $X$  denotes the original input feature map with a size

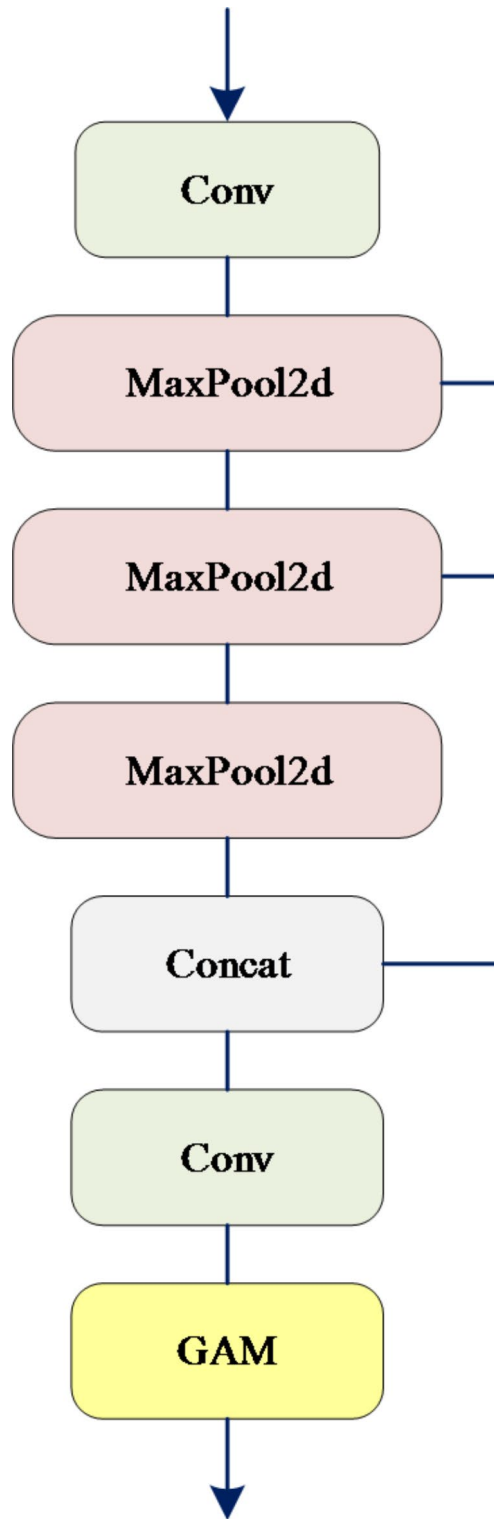


Fig. 8. SPPF-G module.

of  $(N, C, H, W)$ , where  $N$  is the batch size,  $C$  is the number of channels, and  $H$  and  $W$  are the height and width of the feature map, respectively, both of which are 1 in this case. Therefore, so it can be directly multiplied by  $X$ , it is equivalent to weighting the feature maps of each channel. This results in the weighted feature maps for each channel in the SE attention mechanism. Finally,  $X_{SE}$  is substituted into the original DCNv3, yielding our improved DCNv3, as shown in the inference formula (14).

$$Y(P_0) = \sum_{g=1}^G \sum_{k=1}^K W_g m_{gk} X_{(g, X_{SE})}(P_0 + P_k + \Delta P_{gk}) \tag{14}$$

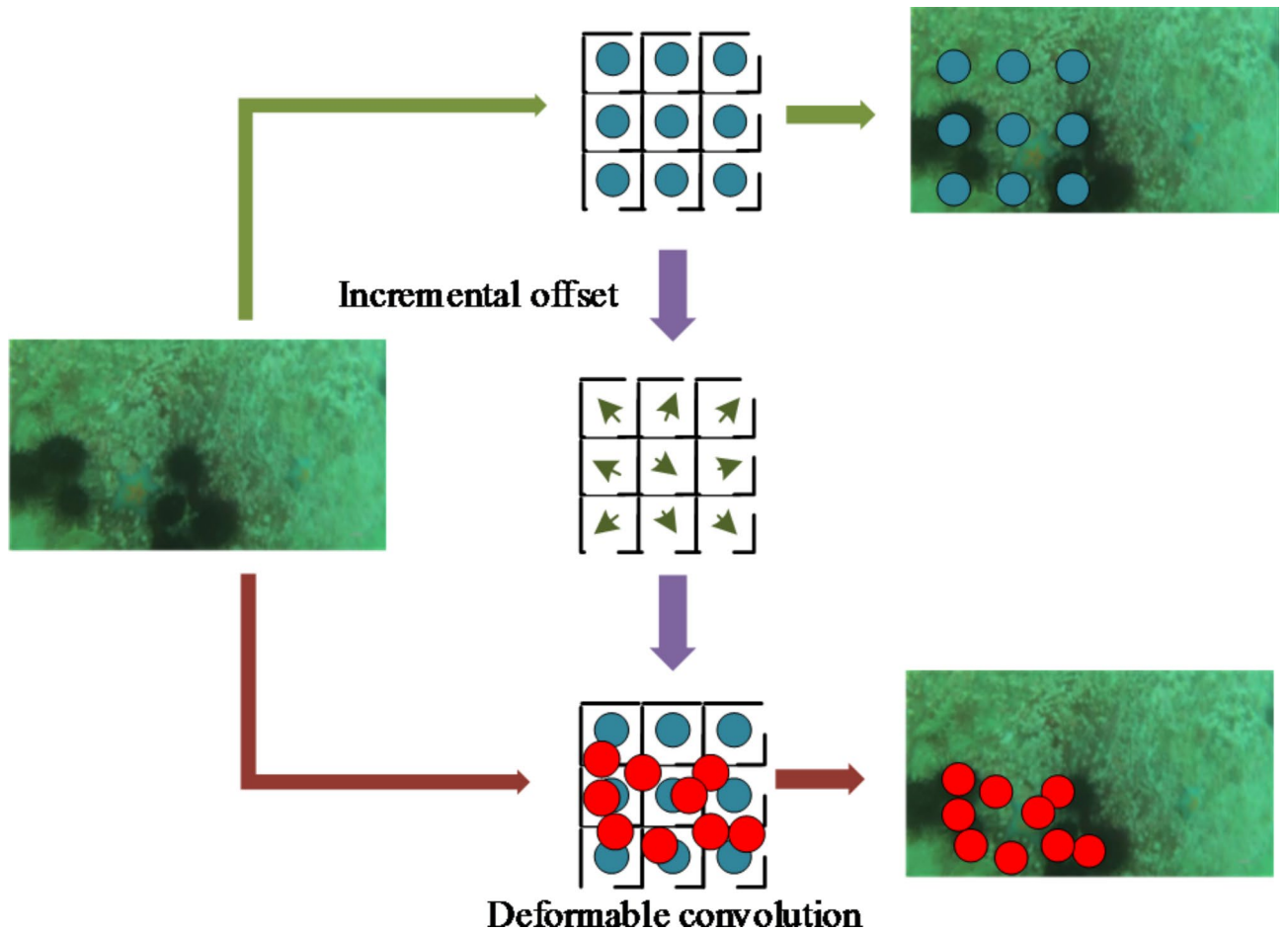


Fig. 9. Schematic diagram of the improved DCNv3 applied to underwater images.

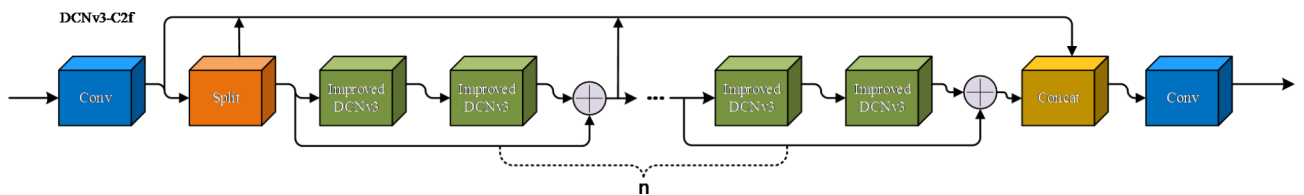


Fig. 10. Structure diagram of DCNv3-C2f module.

where  $Y(P_0)$  represents the improved DCNv3,  $X_{(g, X_{SE})}$  denotes the feature map of the  $g$ -th channel weighted by the SE attention mechanism. The integration of the SE attention mechanism into DCNv3 drives the model focus more on the significant features of the targets, enhancing the model’s perceptual ability towards important information. Moreover, it assists the model in suppressing redundant features, thus allowing the model to handle input data more efficiently.

Figure 9 illustrates the schematic diagram of the improved DCNv3 applied to underwater images.

Then, we incorporate the improved DCNv3 into the C2f module to enhance the feature extraction capability for target images, as illustrated in Fig. 10.

The DCNv3-C2f module employs  $1 \times 1$  convolutions to modify input feature channels, while also using  $1 \times 1$  convolutions for functional splitting. Then, multiple deformable convolution modules are stacked to expand the receptive field. This approach reduces parameter diversity while extracting more multi-scale features.

**CBAM module**

CBAM is an attention mechanism that improves the perceptual ability of the model by introducing Channel Attention Mechanism (CAM) and Spatial Attention Mechanism (SAM) into CNN, thus improving the performance without increasing the network complexity. SAM extracts feature information from input images, while CAM sets weights based on the importance of each feature channel. We introduce the CBAM into the

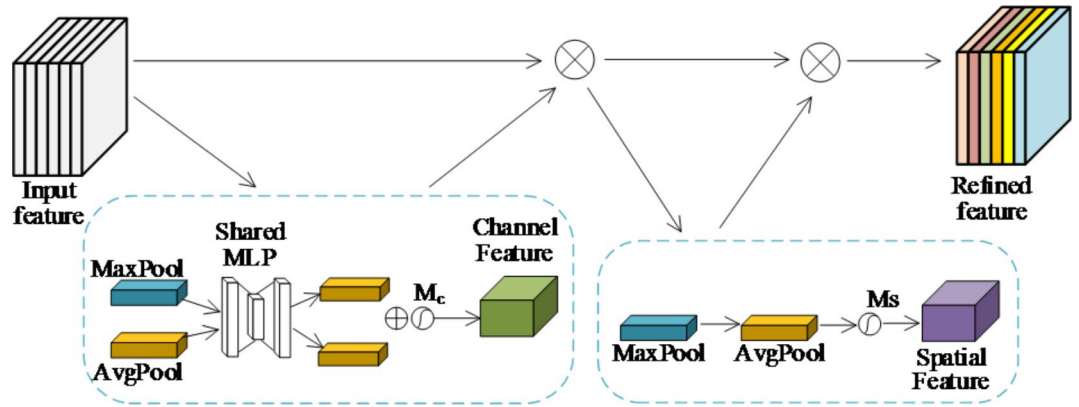


Fig. 11. Structure of the CBAM module.

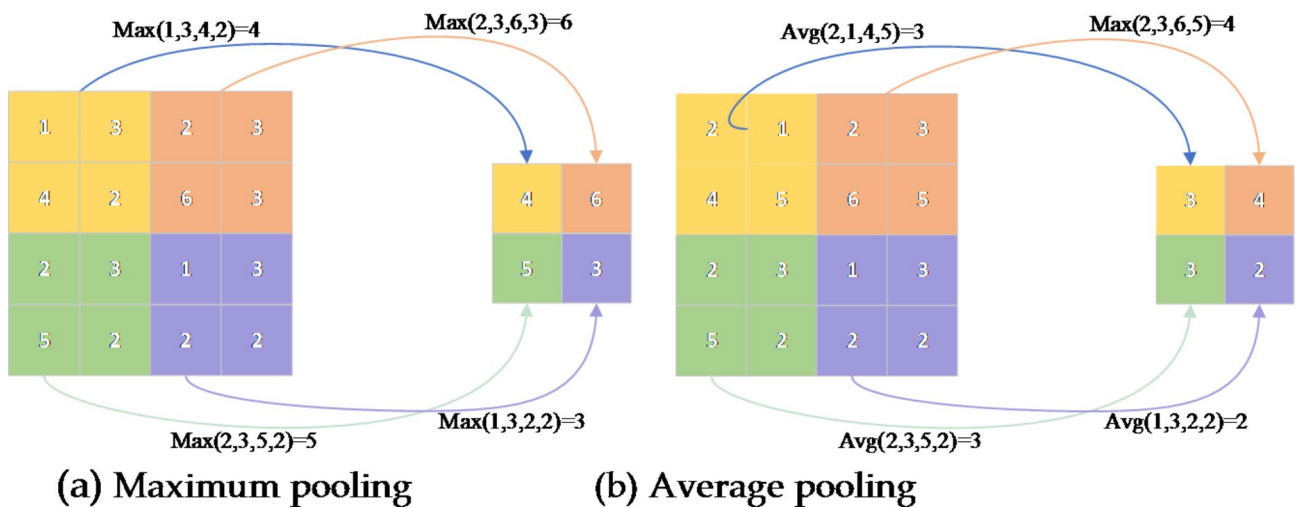


Fig. 12. Pooling process.

backbone network of the model to enhance its perception and generalization abilities, as well as its focus on important target features. The structure of the CBAM module is shown in Fig. 11.

In the CBAM attention mechanism, the channel attention mechanism is first utilized to reconstruct the input feature map<sup>39</sup>. As shown in Fig. 12, the input feature map  $F$  undergoes global average pooling and global maximum pooling to obtain two one-dimensional feature vectors.

These two feature vectors then pass through an  $MLP$  with shared weights. This  $MLP$  is used to learn the attention weight of each channel. Through learning, the network can adaptively decide which channels are more important for the current task. Intersecting the global maximum feature vector and the average feature direction to obtain the final attention weight vector, followed by a weighted summation to compute the attention submodule, as described in Eq. (15).

$$M_c(F) = \tau (MLP(AvgPool(F)) + MLP(MaxPool(F))) \tag{15}$$

where  $M_c(F)$  represents the channel attention module,  $\tau$  represents the Sigmoid function,  $AvgPool(F)$  and  $MaxPool(F)$  respectively denote average pooling and max pooling.

Then, the reconstructed feature map undergoes further processing using the SAM. The spatial attention module primarily conducts average pooling and max pooling simultaneously along the channel axis, and generates a spatial attention map through convolutional layers, as described in Eq. (16).

$$M_s(F') = \tau (f^{7 \cdot 7}([AvgPool(F'); MaxPool(F')])) \tag{16}$$

where  $M_s(F')$  represents the spatial attention module, where  $F'$  is the input feature map.  $f^{7 \cdot 7}$  denotes a  $7 \cdot 7$  convolution operation, and  $AvgPool(F')$  and  $MaxPool(F')$  respectively indicate average pooling and max pooling.

### GSConv module

In the YOLOv8n network model, the role of the neck network layer is to extract feature pyramids. Feature pyramids aid the model in detecting objects of different scales, thereby improving the model's accuracy. The role of the neck network layer in YOLOv8n is to merge feature maps from different levels to improve the accuracy of target feature extraction. The neck network layer in YOLOv8n utilizes traditional convolution, which increases the model's parameter count. Although depth-wise separable convolution modules can reduce the parameter count, they tend to overlook the relationships between channels, leading to information fragmentation.

To preserve more multi-channel information and better represent the intrinsic characteristics of images, we adopt GSConv instead of traditional convolution for up-sampling and down-sampling in the neck network layer. The GSConv module is a combination of traditional convolution, depth-wise separable convolution, and Shuffle convolution<sup>40</sup>. The module achieves this by employing a Shuffle mixing strategy to permeate the feature information generated by standard convolution into each module generated by depth-wise separable convolution, this maintains the output effect of convolution while reducing computational costs. Figure 13 illustrates the structure of the GSConv module.

When the input and output channels are  $C_1$  and  $C_2$  respectively, the process begins with a traditional convolution, which reduces the number of channels in the input feature map to  $C_2/2$ . Then, it undergoes a depth-wise separable convolution, keeping the number of channels unchanged. Finally, the results of the two convolutions are concatenated and shuffled. This process evenly shuffles the channel information, enhancing the extracted semantic information and improving the expressive power of image features. Using the GSConv module allows for maximum sampling effectiveness without increasing model parameters or computational costs, thus ensuring detection accuracy.

### Design of loss function

In the YOLOv8n model, CIoU is used to calculate the loss for bounding boxes. However, CIoU does not consider the issue of sample balance, which can affect the detection results.

In the Fig. 14,  $A$  represents the ground truth box,  $B$  represents the predicted box,  $C$  represents the minimum enclosing box of  $A$  and  $B$ ,  $\rho^2(A, B)$  represents the squared distance between the center points of  $A$  and  $B$ ,  $(x, y)$  and  $(x_{gt}, y_{gt})$  represent the coordinates of the center points,  $w^A, H^A, w^B, H^B$  represent the widths and heights of the two boxes. Compared to the CIoU loss function, the  $H$  and  $W$  in WIoU are separated from the computational graph and focus on the distance between the center points of  $A$  and  $B$  when the boxes overlap. This enables the model to pay more attention to the common-quality anchors and improves the overall performance of the detector.

In our work, we select WIoU v3, which includes a dynamic non-monotonic mechanism. Through reasonable gradient gain allocation, it reduces the occurrence of large gradients or harmful gradients in extreme samples, thereby improving the overall performance of the model.

WIoU v3 is derived by constructing a non-monotonic focusing factor using the degree of abnormality and applying it to WIoU v1. Firstly, distance attention is built based on distance metrics, resulting in WIoU v1 with two layers of attention mechanism, as derived in Eqs. (17)–(19).

$$L_{WIoUv1} = R_{WIoU} \times L_{IoU} \quad (17)$$

$$R_{WIoU} = \exp\left(\frac{(x - x_{gt})^2 + (y - y_{gt})^2}{(W_g^2 + H_g^2)^*}\right) \quad (18)$$

$$L_{IoU} = 1 - IoU \quad (19)$$

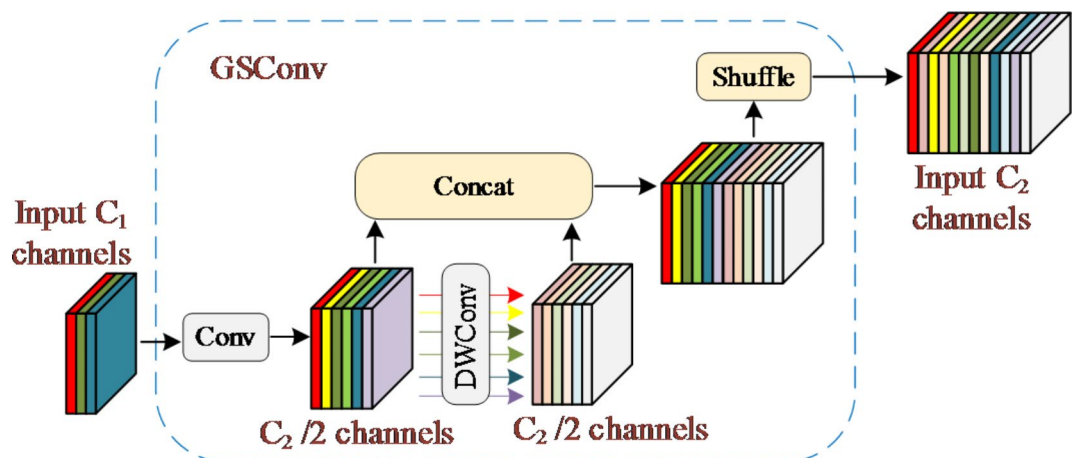


Fig. 13. Structure of the GSConv module.

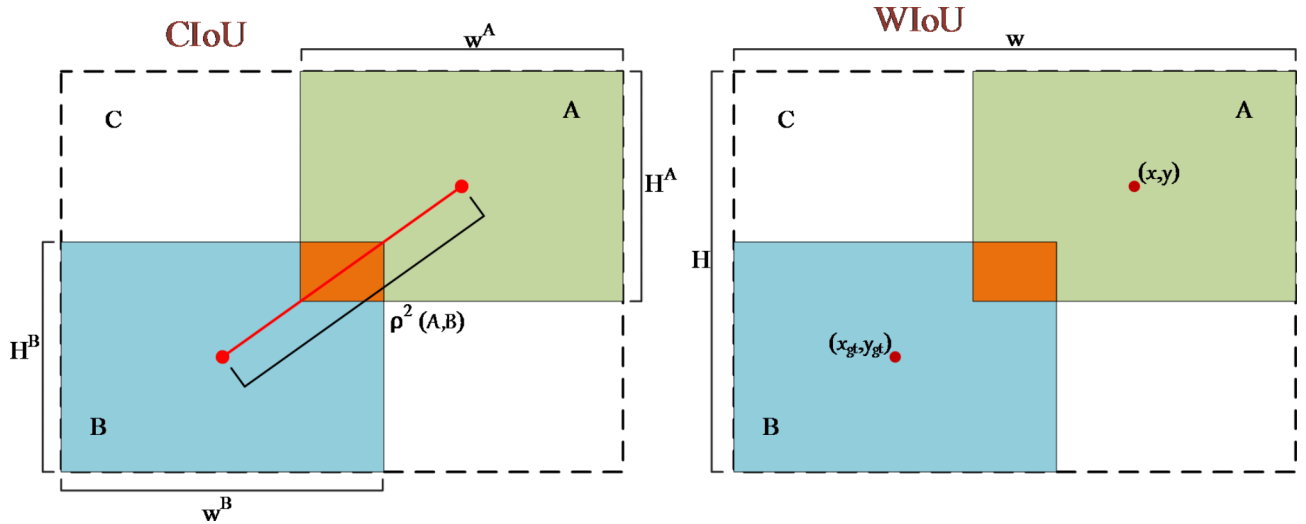


Fig. 14. Schematic diagram of the loss function.

Here,  $(x, y)$  and  $(x_{gt}, y_{gt})$  represent the coordinates of the anchor box and the target box center, respectively.  $W_g$  and  $H_g$  denote the dimensions of the minimum bounding box. IoU quantifies the overlap between a predicted bounding box and a ground truth bounding box. However, WIoU v1 does not introduce the concept of class weights, leading to insufficient attention to minority classes. Moreover, it does not consider the IoU distribution of each class, which may cause some classes to be overly optimized or ignored. Therefore, utilizing  $\beta$  (Abnormal degree) to construct a non-monotonic focusing coefficient and applying it to WIoU v1 yields WIoU v3 with dynamic non-monotonic FM, as shown in Eqs. (20, 21).

$$L_{WIoUv3} = L_{WIoUv1} \times \frac{\beta}{\delta \alpha^{\beta-\delta}} \tag{20}$$

$$\beta = \frac{L^*_{IoU}}{L_{IoU}} \in [0, +\infty) \tag{21}$$

Where \* prevents hinderance of convergence speed.  $\alpha$  and  $\delta$  are hyperparameters, with values set to 1.8 and 3 in our work. From Eq. (20), when  $\frac{\beta}{\delta \alpha^{\beta-\delta}}$  equals 1, WIoU v3 becomes v1 version. Furthermore, because v3 evaluates anchor box quality using a dynamic non-monotonic mechanism, the model can dynamically adjust the gradient gain allocation strategy.

### Embedded device deployment

To validate the feasibility of the proposed low-light underwater target detection model, a detection system based on the RV1126 embedded hardware platform was designed. RV1126 is an intelligent vision chip developed by Rockchip, featuring a quad-core ARM Cortex-A7 processor and a built-in NPU (Neural Processing Unit) with 2 TOPS computing power. It is more suitable for testing lightweight network models. Hence it is chosen as the hardware porting platform for this study.

Typically, the deep learning parameters trained on a PC are saved in a specified model and cannot be directly applied to a hardware platform. Transferring the model to the RV1126 platform requires parameter extraction and format conversion. As shown in Fig. 15, the schematic diagram illustrates the process of deploying the proposed model to the RV1126 platform.

Firstly, the trained weight file is converted into an open neural network model (onnx), and then the model is quantized. Finally, the quantized model is converted into an executable file of hardware equipment in Linux environment. The inference process is depicted in Fig. 16.

To validate the superiority of the model on hardware platforms, experiments were conducted on three platforms: GPU 3080Ti, X86 CPU, and RV1126. A comparison was conducted regarding computational load, memory usage, inference speed, and power consumption. The results are presented in Table 1.

When deploying the model to the RV1126 hardware device, we first quantized the model to reduce computational load and system memory usage. Additionally, RV1126 is a low-power hardware platform, consuming less power compared to CPU and GPU. Although GPU boasts the highest inference rate, it also consumes the highest power. In contrast, while the inference rate of RV1126 may not match that of GPU and CPU, its computational load, memory usage, and power consumption are all superior to them. This validates that the designed low-light underwater target detection system has a comprehensive advantage over other platforms.

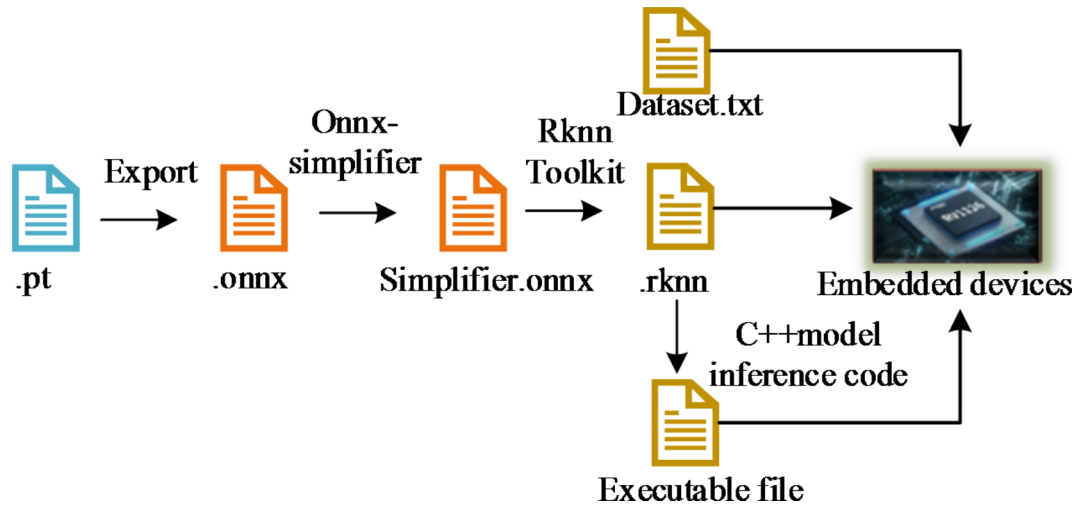


Fig. 15. Embedded device deployment process.

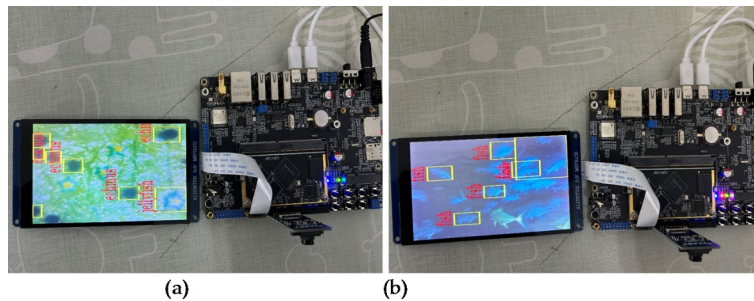


Fig. 16. Inference on the RV1126 Hardware Platform.

Equipment	FLOPs/G	Memory/MB	Frame rate/FPS	Power/W
GPU	7.0	1501	187.9	250
CPU	7.0	2208	297.8	64
RV1126	6.7	138	48.2	4.6

Table 1. Analysis of results on different devices.

## Experiment Dataset

Underwater target detection belongs to the category of target detection in special scenarios. Compared to general target detection, underwater imaging is complex and faces numerous challenges in detection. In simple terms, underwater target detection faces two major challenges: on one hand, underwater detection deals with many complex marine targets. Social marine targets (such as fish and coral) often live in groups, leading to issues like mutual occlusion. Moreover, marine targets often exhibit characteristics like small size, diverse postures, and various forms of camouflage. Additionally, marine targets frequently exhibit large intra-class variations and similar inter-class shapes, which further increases the difficulty of underwater detection. On the other hand, underwater detection faces complex environmental challenges. Environmental challenges mainly include issues such as fog effects, color bias, and light interference. Then we analyze the existing underwater datasets and draw a table, as shown in Table 2.

RUOD underwater target dataset is the only one that covers general underwater scenes and various underwater detection challenges. There are 14,000 pieces of this dataset. The dataset is divided into training set and test set according to the ratio of 49:21. The labeled target categories include: fish, diver, starfish, coral, turtle, sea urchin, sea cucumber, scallop, squid and jellyfish. The specific values of each data set category are shown in Table 3.

Dataset	Image			Data			Split				Challenges			
	#Lab	Bac	#Cat.	#Tran	#Tes	#Tot	HLE	CC	LL	CMO	HLE	CC	LL	CMO
URPC2018	22,688	S	4	2901	800	3701	√	√	×	×	√	√	×	×
URPC2019	36,100	S	4	4757	1029	5786	√	√	×	×	√	√	×	×
URPC2020	46,287	S	4	6575	2400	8975	√	√	×	×	√	√	×	×
URPC2021	54,238	S	4	7478	1200	8678	√	√	×	×	√	√	×	×
UDD <sup>41</sup>	15,022	S	4	1827	400	2227	√	√	×	×	√	√	×	×
DUO <sup>42</sup>	74,515	S	4	6671	1111	7782	√	√	×	×	√	√	×	×
UODD <sup>43</sup>	19,212	S	4	2688	506	3194	√	√	×	×	√	√	×	×
RUOD	74,903	G	10	9800	4200	14,000	√	√	√	√	√	√	√	√

**Table 2.** Current underwater dataset.

Experiment Set	Train	Validation	Total								
Quantity of images	9800	4200	14,000								
Category	Fish	Diver	Starfish	Coral	Turtle	Sea urchin	Sea cucumber	Scallop	Squid	Jellyfish	
Quantity of samples	2890	1250	1100	1520	1350	1210	950	1400	1080	1250	

**Table 3.** Quantity of images and samples in underwater target detection dataset.



**Fig. 17.** Sample dataset.

The RUOD dataset also includes three test sets of environmental challenges, namely fog effect, color cast and light interference. This dataset can comprehensively evaluate the performance of this model. Sample images of the dataset are shown in Fig. 17.

Firstly, before training the model, we conducted an inspection of the training set used in the experiment. The inspection results are shown in Fig. 18.

In Fig. 18, subFigs from left to right represent the following: The number of instances for each class in the training set; The size and number of bounding boxes; The position of sample centroids relative to the image and the aspect ratio of target objects relative to the image.

### Experimental setup

To ensure the comparability of the training results, the software and hardware used in this article are all experimented under the same environment. The parameter configuration used in the experiments is shown in Table 4.

During training, the following relevant parameters were used: epochs were set to 300, batch\_size was set to 40, the input size of the images was  $640 \times 640$ , stochastic gradient descent (SGD) was used for model optimization.

### Model evaluation metrics

In this paper, we use the following evaluation metrics: parameter count, FLOPS, FPS, mAP@0.5, and mAP@0.5:0.95. The relevant calculation formulas are shown in equations (22) to (25).

$$Precision = \frac{TP}{TP + FP} \quad (22)$$

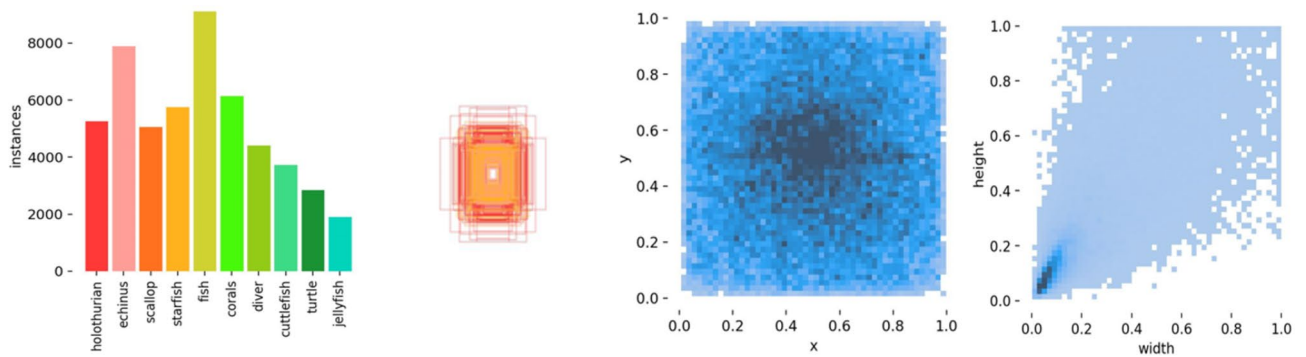


Fig. 18. Analysis of underwater object dataset.

	Environment	Version
Software environment	Operating system	Windows 11
	CUDA version	11.7
	CPU	12th Gen Intel(R) Core(TM) i5-12500
	GPU	Nvidia GeForce RTX 3080Ti
	Video memory	12G
	Python version	python3.8
	Deep learning framework	Pytorch-1.13.0

Table 4. Experimental setup.

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

$$AP = \int_0^1 P(r) dr \tag{24}$$

$$mAP = \frac{1}{num(class)} \int_0^1 P(r) dr \tag{25}$$

In the equations,  $TP$  and  $FP$  represent the proportions of true positive and false positive predictions respectively, while  $FN$  denotes the number of incorrectly predicted samples in the negative class.  $AP$  stands for average precision for a single class, and  $mAP@0.5$  is the mean average precision for all classes. The value of  $mAP@0.5$  ranges from 0 to 1, with a higher value indicating better detection performance.

## Results

### RUOD dataset results

To validate the performance of the improved PDSC-YOLOv8n model, we conducted experiments using YOLOv8n as the baseline. Firstly, we trained the YOLOv8n model before and after improvement under the same environment. The training results are shown in Fig. 19.

The specific parameters obtained by training for each category are shown in Table 5.

From Table 5, we can know that compared with YOLOv8n, the improved PDSC-YOLOv8n model improves the detection accuracy of each dataset category to varying degrees, which shows that PDSC-YOLOv8n has a better effect. The experimental results demonstrate that our proposed PDSC-YOLOv8n algorithm exhibits superior performance and more precise detection accuracy compared to the baseline YOLOv8n. Additional specific data is shown in Table 6.

From Table 6, we can see that the parameter count for YOLOv8n is 6.65 M, with a computational load of 8.71GFlops. The improved PDSC-YOLOv8n has a parameter count of 5.62 M and a computational load of 8.96GFlops. Although the computational load has increased by 0.25GFlops, the parameter count has decreased by 1.03 M compared to YOLOv8n, representing a reduction of approximately 15.5%. This demonstrates that our proposed model is more lightweight. YOLOv8n achieves  $mAP@0.5$  and  $mAP@0.5:0.95$  of 79.6% and 58.2%, respectively, while using the improved PDSC-YOLOv8n yields  $mAP@0.5$  and  $mAP@0.5:0.95$  of 86.1% and 60.8%, respectively. Compared to YOLOv8n, there is an improvement of 6.5% and 2.6% in  $mAP@0.5$  and  $mAP@0.5:0.95$ , respectively, indicating that our improved PDSC-YOLOv8n algorithm has better performance.

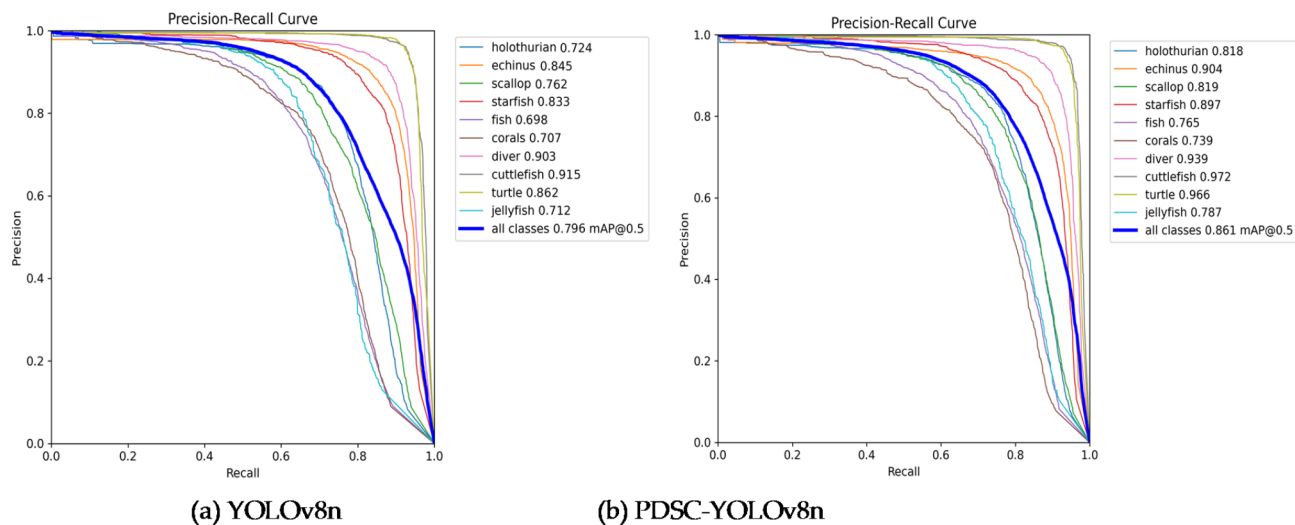


Fig. 19. Comparison of training results of improved YOLOv8n.

Model	Holothurian	Echinus	Scallop	Starfish	Fish	Corals	Diver	Cuttlefish	Turtle	Jellyfish
YOLOv8n	0.724	0.845	0.762	0.833	0.698	0.707	0.903	0.915	0.862	0.712
PMGS-YOLOv8n	0.818	0.904	0.819	0.897	0.765	0.739	0.939	0.972	0.996	0.787
Accuracy variation	+0.094	+0.059	+0.057	+0.064	+0.067	+0.032	+0.036	+0.057	+0.104	+0.075

Table 5. Training results of each category.

Model	Parameter/M	GFlops	mAP@0.5	mAP@0.5:0.95	FPS
YOLOv8n	6.65	8.71	79.6	58.2	318.7
PDSC-YOLOv8n	5.62	8.96	86.1	60.8	297.8

Table 6. Comparisons between our proposed model and its baseline.

### Experimental results of pascal VOC2012 dataset migration

At the same time, we also conducted transfer learning experiments on the Pascal VOC2012<sup>44</sup> dataset using the YOLOv8n model before and after improvements. The Pascal VOC2012 dataset is a standard dataset released as part of the PASCAL VOC challenge in 2012. It contains 20 categories, covering common objects from daily life, such as humans, animals, vehicles, and indoor items. These categories are not only diverse but also effectively reflect the complexity of real-world application scenarios. From the R-CNN series to models like YOLO and SSD, many object detection models have been developed based on the Pascal VOC dataset and its challenges. This dataset has played a significant role in advancing the fields of machine learning and deep learning, serving as a valuable benchmark for evaluating and promoting the performance of algorithms in object recognition, classification, object detection, and other visual detection tasks. Therefore, in addition to using the RUOD dataset, this paper incorporates the Pascal VOC dataset for supplementary experiments. As shown in Fig. 20, the results of transfer learning experiments using the YOLOv8n model before and after improvements on the Pascal VOC2012 dataset are presented.

Other experimental parameters are shown in Table 7.

From the data in Table 7, it can be seen that the mAP@0.5 obtained using the improved PDSC-YOLOv8n on the Pascal VOC2012 dataset is 78.5%, which represents an improvement of 5.5% compared to the original version. The FPS also increased by 51.8. This demonstrates that our improved PDSC-YOLOv8n performs well on new datasets.

### Ablation experiment

To understand how each improvement algorithm affects the test results differently, we conducted ablation experiments for comparison. We still used the RUOD dataset for this purpose. Experiment 1 involved using YOLOv8n as the baseline to obtain the experimental data.

Experiment 12 represents the experimental data obtained using our proposed PDSC-YOLOv8n. Experiments 2 to 11 represent the experimental data obtained within the framework of YOLOv8n, with the presence of  $\checkmark$  indicating whether the improvement modules were added. The experimental results are presented in Table 8.

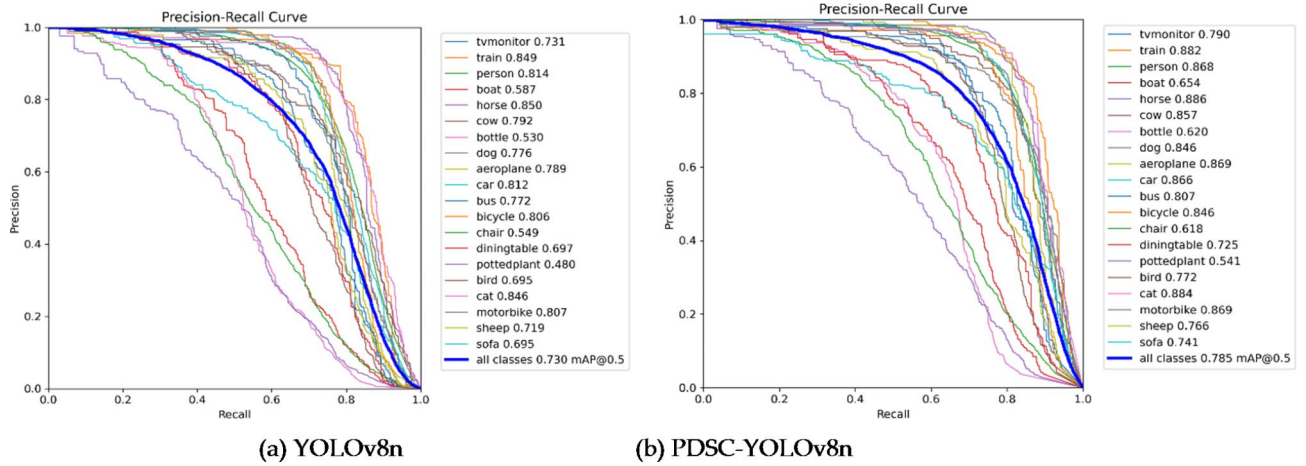


Fig. 20. Experimental results of Pascal VOC2012 dataset.

Model	Parameter/M	GFlops	mAP@0.5	mAP@0.5:0.95	FPS
YOLOv8n	7.76	9.56	73%	53.2	234.7
PDSC-YOLOv8n	5.89	8.23	78.5	57	286.5

Table 7. Transfer learning experiments on the Pascal VOC2012 dataset.

Experiment	Pro MSRCR	DCNv3-C2f	SPPF-G	CBAM	WIoUv3	Parameter/M	GFlops	mAP@0.5	mAP@0.5:0.95
1						6.65	8.71	79.6	58.2
2	√					6.69	8.74	80.5	58.1
3		√				6.76	8.69	81.3	59.2
4			√			6.87	8.21	80.9	57.9
5				√		6.68	8.72	82.7	58.4
6					√	6.75	8.77	81.6	58.8
7	√	√	√			6.73	8.97	83.4	58.9
8	√		√	√		6.86	9.12	84.4	60.1
9	√		√		√	6.74	8.89	83.7	59.8
10		√	√	√		6.82	8.80	84.0	60.3
11		√	√		√	6.67	9.01	84.9	59.8
12			√	√	√	6.84	8.87	85.2	60.4
13	√	√	√	√	√	5.62	8.96	86.1	60.8

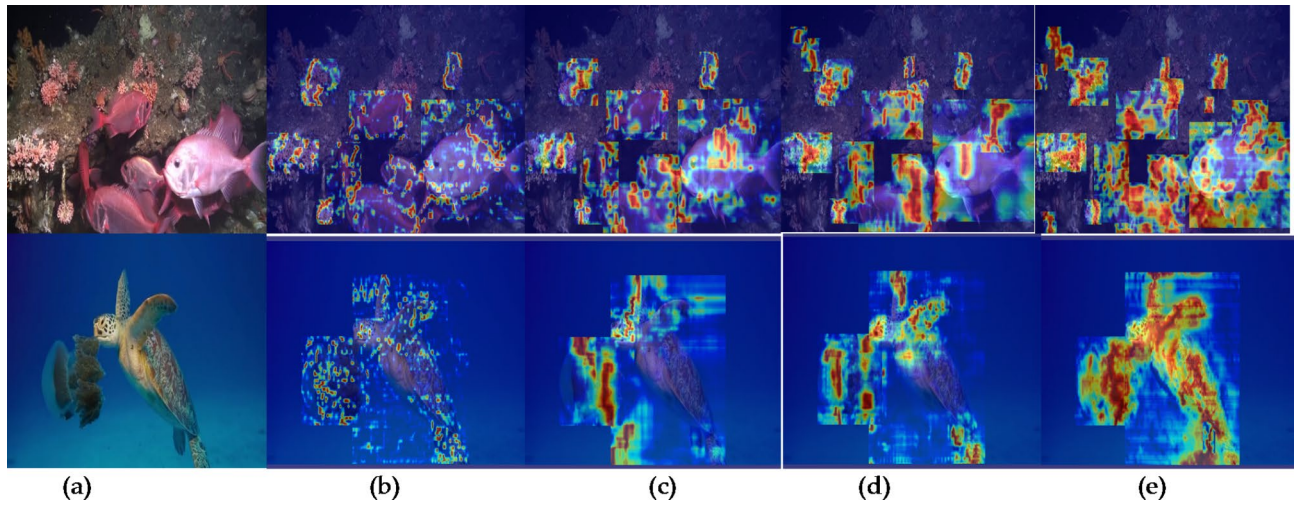
Table 8. Results of ablation experiments.

Based on Experiment 1 from Table 8, Pro MSRCR, DCNv3-C2f, SPPF-G, CBAM and WIoUv3 modules are used as the basis of ablation experiments respectively. The experimental results show that the PDSC-YOLOv8n proposed in this paper has better performance.

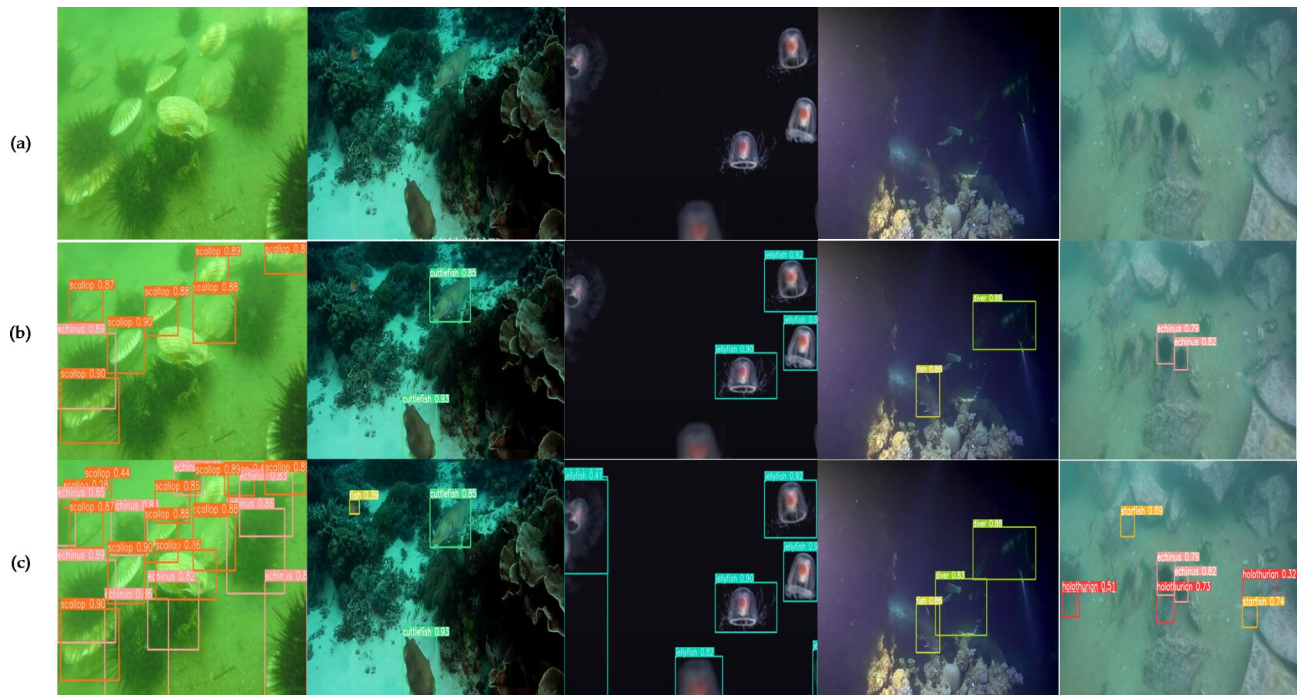
Meanwhile, as DCNv3-C2f and CBAM are both added to the model's backbone network, this section not only evaluates using traditional evaluation metrics but also provides heatmaps with the addition of DCNv3-C2f and CBAM attention mechanisms. Fish and turtle images are used as representative examples, as shown in Fig. 21.

As shown in Fig. 21, (a) represents the original image from the dataset, (b) represents the heatmap of YOLOv8n, (c) represents the heatmap with the addition of the improved DCNv3 module, (d) represents the heatmap with the addition of the improved CBAM module, and (e) represents the final output heatmap of the backbone network for the PDSC-YOLOv8n model. From Fig. 21b, it can be observed that the heatmap obtained from YOLOv8n without any added improvement modules is insensitive to the detection of positive samples. However, after adding the improvement modules, the heatmap gradually becomes more sensitive to the features of positive samples to be detected. It effectively eliminates the features of non-class objects to be detected, focusing more attention on the potential positive samples to be detected. This leads to more accurate and clearer object detection in the input image<sup>45</sup>.

As shown in Fig. 22, the results of underwater object detection under low light and blurry conditions using YOLOv8n and PDSC-YOLOv8n are presented.



**Fig. 21.** Heatmaps generated by the proposed PDSC-YOLOv8n.



**Fig. 22.** Sample detection results.

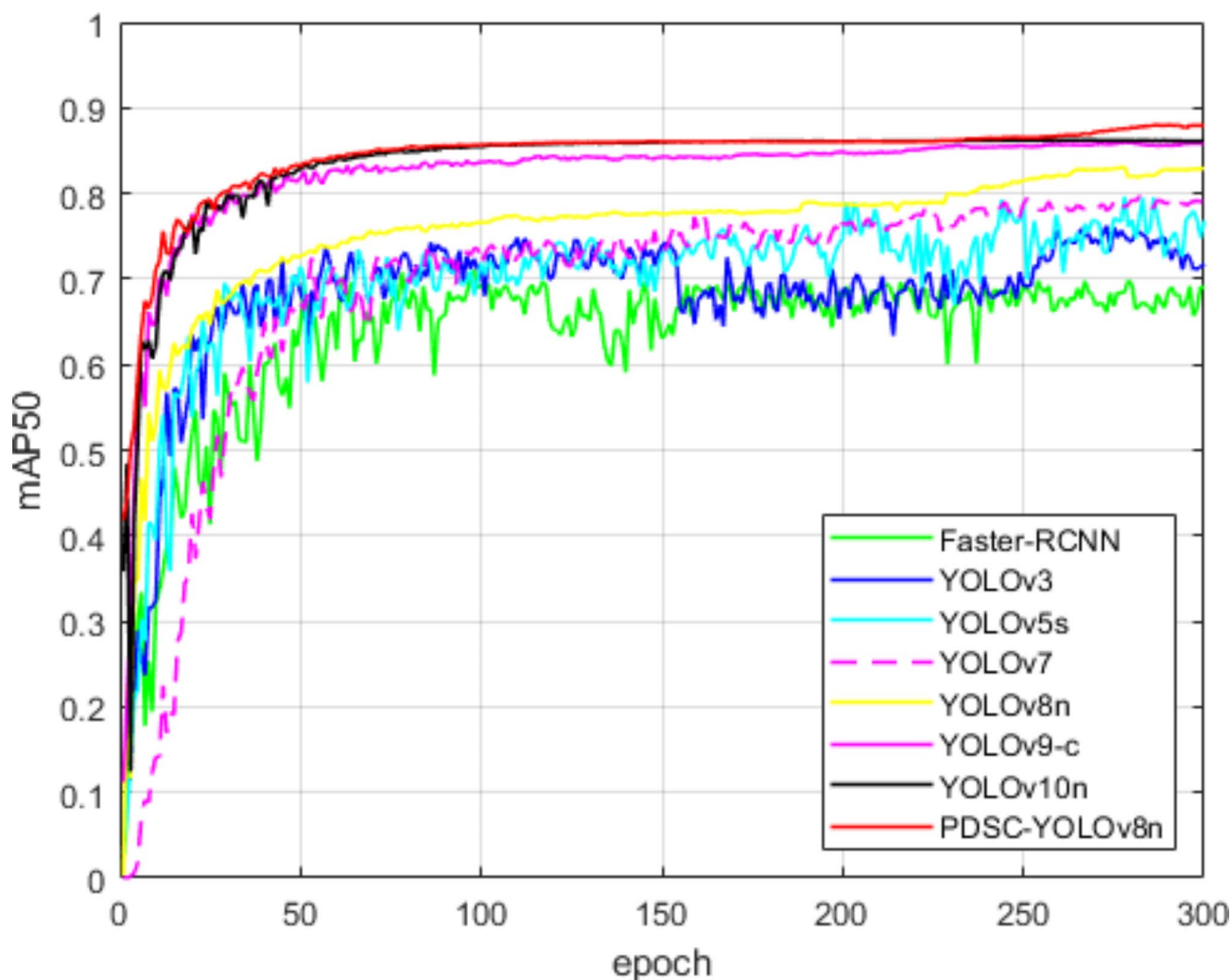
In addition, in order to verify that the algorithm proposed in this paper is superior to other detection models in underwater target detection, we use Faster-RCNN, YOLOv3, YOLOv5s, YOLOv7, YOLOv8n and the latest YOLOv9-c and YOLOv10n in this field for comparative experiments. The experimental results are shown in Table 9.

From Table 9, using different algorithms to experiment on the RUOD dataset. The mAP@0.5 obtained using the Faster-RCNN algorithm is 67.4%, while using the YOLOv3 algorithm yields an mAP@0.5 of 69.6%. The YOLOv5s algorithm achieves an average detection precision mAP@0.5 of 73.8%, and the YOLOv7 algorithm achieves an mAP@0.5 of 74.9%. Using YOLOv8n results in an mAP@0.5 of 79.6%. The mAP@0.5 of YOLOv9-c and YOLOv10n are 84.4% and 85.9% respectively.

In contrast, our proposed PDSC-YOLOv8n algorithm achieves an average detection precision mAP@0.5 of 86.1%, which is higher than the results obtained by the other algorithms. Therefore, our proposed PDSC-YOLOv8n algorithm is more suitable for underwater object detection. To visually compare the detection results of each algorithm more intuitively, We plotted a comparison graph of the average precision for each algorithm, as shown in Fig. 23.

Model	Parameter/M	GFlops	mAP@0.5	mAP@0.5:0.95	FPS
Faster-RCNN	13.72	18.12	67.4	44.3	75.6
YOLOv3	12.44	28.91	69.6	46.8	68.9
YOLOv5s	26.58	16.80	73.8	54.7	305.8
YOLOv7	35.69	103.22	74.9	56.5	215.7
YOLOv8n	6.65	8.71	79.6	58.2	318.7
YOLOv9-c	25.3	108.3	84.4	60.7	197.9
YOLOv10n	7.86	17.6	85.9	56.6	286.8
Our	5.62	8.96	86.1	60.8	297.8

**Table 9.** Comparisons between our proposed model and the other YOLO models.



**Fig. 23.** Comparison of results from different algorithms.

Through the comparison in Fig. 23, it can be observed that our proposed algorithm exhibits superior performance and detection accuracy.

### Conclusions and future works

The paper firstly discusses the challenges posed by the complex underwater environment, characterized by severe light attenuation, resulting in blurred and low-quality underwater target images. These factors contribute to difficulties in underwater target detection. To address this issue, we propose a lightweight low-light underwater target detection algorithm, PDSC-YOLOv8n, based on the improved YOLOv8n model.

Firstly, we introduce the improved Pro MSRCR algorithm at the input of the model to enhance the information of low-light and blurred target images in the dataset. Secondly, Ghost and GConv convolutional modules

are adopted for the backbone and neck networks of YOLOv8n, respectively, to achieve model lightweighting. Additionally, we incorporate the improved DCNv3 module into the C2f module of the backbone network to enhance the feature extraction capability of targets. Then, we enhance the SPPF module by introducing the GAM attention mechanism to improve the model's recognition ability for objects across different scales. Furthermore, we introduce the CBAM attention mechanism at the last layer of the backbone network. Finally, we utilize WIoUv3 as the loss function to accelerate network convergence and optimize the model training process. The proposed PDSC-YOLOv8n algorithm has been successfully deployed on an embedded platform, achieving real-time detection of underwater targets. Experiments on underwater dataset in RUOD and Pascal VOC2012 dataset migration show that our PDSC-YOLOv8n model has higher detection accuracy.

However, our work also has some shortcomings. There is still ample room for improvement in the overall network structure of the model, and further research is needed. Our next step will focus on further improving the network model and designing more sensors, such as LiDAR, infrared obstacle avoidance, and remote control, to make it more powerful in the field of underwater target detection.

## Data availability

The data presented in this study are available on request from the corresponding author due to original data is still doing more in-depth research.

Received: 2 June 2024; Accepted: 6 November 2024

Published online: 13 November 2024

## References

- Zhao, M., Zhou, H. & Li, X. YOLOv7-SN: underwater target detection algorithm based on improved YOLOv7. *Symmetry* **16**, 514 (2024).
- Zhu, J. et al. YOLOv8-C2f-Faster-EMA: an improved underwater trash detection model based on YOLOv8. *Sensors* **24**, 2483 (2024).
- Wang, H., Zhang, P., You, M. & You, X. A method for underwater biological detection based on improved YOLOs. *Appl. Sci.* **14**, 3196 (2024).
- Zhang, X., Zhu, D. & Gan, W. YOLOv7t-CEBC network for underwater litter detection. *J. Mar. Sci. Eng.* **12**, 524 (2024).
- Hu, S. & Liu, T. Underwater rescue target detection based on acoustic images. *Sensors* **24**, 1780 (2024).
- Xi, J. & Ye, X. Sonar image target detection based on simulated stain-like noise and shadow enhancement in optical images under zero-shot learning. *J. Mar. Sci. Eng.* **12**, 352 (2024).
- Valido, M. R., Bendicho, P. F., Martín Reyes, M. & Rodríguez-Juncá, A. Software application for automatic detection and analysis of biomass in underwater videos. *Appl. Sci.* **13**, 10870 (2023).
- Cheng, C., Hou, X., Wen, X., Liu, W. & Zhang, F. Small-sample underwater target detection: a Joint Approach utilizing diffusion and YOLOv7 model. *Remote Sens.* **15**, 4772 (2023).
- Jialu, H. & Qing, J. IG-YOLOv5-based underwater biological recognition and detection for marine protection. *Open. Geosci.* **15**, 1 (2023).
- Zhao, L. et al. YOLOv7-CHS: an emerging model for underwater object detection. *J. Mar. Sci. Eng.* **11**, 10 (2023).
- Chen, T. & Qi, Q. Research on the cooperative target state estimation and tracking optimization method of Multi-UUV. *Sensors* **23**, 7865 (2023).
- Bao, Z. et al. Underwater target detection based on parallel high-resolution networks. *Sensors* **23**, 7337 (2023).
- Wen, G. et al. YOLOv5s-CA: a modified YOLOv5s network with coordinate attention for underwater target detection. *Sensors* **23**, 3367 (2023).
- Tang, P. et al. Real-world underwater image enhancement based on attention U-Net. *J. Mar. Sci. Eng.* **11**, 662 (2023).
- Wang, Z., Chen, H., Qin, H. & Chen, Q. Self-supervised pre-training joint framework: assisting lightweight detection network for underwater object detection. *J. Mar. Sci. Eng.* **11**, 604 (2023).
- Zeng, Y., Shen, S. & Xu, Z. Water surface acoustic wave detection by a millimeter wave radar. *Remote Sens.* **15**, 4022 (2023).
- Joshi, R. et al. Underwater object detection and temporal signal detection in turbid water using 3D-integral imaging and deep learning. *Opt. Express* **32**(2), 1789–1801 (2024).
- Liu, K. et al. Underwater target detection based on improved YOLOv7. *J. Mar. Sci. Eng.* **11**, 677 (2023).
- Wang, Z. et al. Diseased fish detection in the underwater environment using an improved YOLOv5 network for intensive aquaculture. *Fishes* **8**, 169 (2023).
- Fu, J. & Tian, Y. U. Target detection based on Improved YOLOv7. *IAENG Int. J. Comput. Sci.* **51**, 4 (2024).
- Zhu, Y. et al. Detection of underwater targets using polarization laser assisted Echo detection technique. *Appl. Sci.* **13**, 3222 (2023).
- Liu, K., Peng, L. & Tang, S. Underwater object detection using TC-YOLO with attention mechanisms. *Sensors* **23**, 2567 (2023).
- Wang, J. et al. An underwater dense small object detection model based on YOLOv5-CFDSSE. *Electronics* **12**, 15 (2023).
- Chen, L. et al. Lightweight underwater target detection Algorithm based on dynamic sampling transformer and knowledge-distillation optimization. *J. Mar. Sci. Eng.* **11**, 426 (2023).
- An, G., Kaiqiong, S. & Ziyi, Z. A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection. *J. Real-Time Image Process.* **21**, 2 (2024).
- Yao, H., Gao, T., Wang, Y., Wang, H. & Chen, X. Mobile\_ViT: underwater acoustic target Recognition Method based on local-global feature fusion. *J. Mar. Sci. Eng.* **12**, 589 (2024).
- Yin, F. et al. Weak underwater acoustic target detection and enhancement with BM-SEED algorithm. *J. Mar. Sci. Eng.* **11**, 357 (2023).
- Shi, Y., Li, S., Liu, Z., Zhou, Z. & Zhou, X. MTP-YOLO: you only look once based Maritime tiny person detector for emergency rescue. *J. Mar. Sci. Eng.* **12**, 669 (2024).
- Chen, L. et al. Underwater target detection lightweight algorithm based on multi-scale feature fusion. *J. Mar. Sci. Eng.* **11**, 320 (2023).
- Gao, Y., Liu, W., Chui, H. C. & Chen, X. Large span sizes and irregular shapes target detection methods using variable convolution-improved YOLOv8. *Sensors* **24**, 2560 (2024).
- Aguirre-Castro, O. A. E. E. et al. Evaluation of underwater image enhancement algorithms based on Retinex and its implementation on embedded systems. *Neurocomputing* **494**, 148–159 (2022).
- Xu, W., Zheng, X., Tian, Q. & Zhang, Q. Study of underwater large-target localization based on Binocular Camera and Laser Rangefinder. *J. Mar. Sci. Eng.* **12**, 734 (2024).
- Li, J., Liu, C., Lu, X. & Wu, B. CME-YOLOv5: an efficient object detection network for densely spaced fish and small targets. *Water* **14**, 2412 (2022).

34. Sun, C., Wei, Y., Wang, W., Wu, Z. & Li, Y. Water level inversion detection method for water level Images without a scale in complex environments. *Water* **16**, 1176 (2024).
35. Dinakaran, R., Zhang, L., Li, C. T., Bouridane, A. & Jiang, R. Robust and fair undersea target detection with automated underwater vehicles for biodiversity data collection. *Remote Sens.* **14**, 3680 (2022).
36. Li, H. et al. *DCNv3: Towards Next Generation Deep Cross Network for CTR Prediction* (2024).
37. Wang, M., Xu, C., Zhou, C., Gong, Y. & Qiu, B. Study on underwater target Tracking Technology based on an LSTM–Kalman Filtering Method. *Appl. Sci.* **12**, 5233 (2022).
38. Yuan, X., Guo, L., Luo, C., Zhou, X. & Yu, C. A survey of target detection and recognition methods in underwater turbid areas. *Appl. Sci.* **12**, 4898 (2022).
39. Pan, T. et al. Experimental study on Bottom-Up detection of underwater targets based on polarization imaging. *Sensors* **22**, 2827 (2022).
40. Lei, F., Tang, F. & Li, S. Underwater target detection Algorithm based on improved YOLOv5. *J. Mar. Sci. Eng.* **10**, 310 (2022).
41. Wang, Z. et al. *UDD: An Underwater Open-sea Farm Object Detection Dataset* (For Underwater Robot Picking arXiv, 2020).
42. Liu, C. et al. *A Dataset And Benchmark Of Underwater Object Detection For Robot Picking* (2021).
43. Liu, C. et al. *A New Dataset, Poisson GAN and AquaNet for Underwater Object Grabbing* (Institute of Electrical and Electronics Engineers (IEEE), 2021).
44. Everingham, M. et al. The Pascal Visual object classes (VOC) challenge. *Int. J. Comput. Vis.* **88**, 303–338 (2010).
45. Zu, Y., Zhang, L., Li, S., Fan, Y. & Liu, Q. EF-UODA: underwater object detection based on enhanced feature. *J. Mar. Sci. Eng.* **12**, 729 (2024).

## Acknowledgements

We express our gratitude to the editor and the esteemed reviewers for the invaluable feedback provided on our manuscript.

## Author contributions

The research was designed by H and L. H, L and Z carried out experiments, analyzed data, and drafted the manuscript. The manuscript was revised by D. All authors re-viewed and approved the final version of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to J.D.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024