frontiers | Frontiers in Marine Science

TYPE Original Research
PUBLISHED 23 January 2025
DOI 10.3389/fmars.2024.1513740

OPEN ACCESS

EDITED BY
Narayanamoorthi R.,
SRM Institute of Science and Technology, India

REVIEWED BY
Zhaoqiang Xia,
Northwestern Polytechnical University, China
Hao Wang,
China University of Petroleum, China
Mingzhi Chen,
University of Shanghai for Science and Technology, China

*CORRESPONDENCE
Cong Lin
lincong@gdou.edu.cn

RECEIVED 19 October 2024
ACCEPTED 30 December 2024
PUBLISHED 23 January 2025

CITATION
Liu M, Wu Y, Li R and Lin C (2025) LFN-YOLO: precision underwater small object detection via a lightweight reparameterized approach. *Front. Mar. Sci.* 11:1513740. doi: 10.3389/fmars.2024.1513740

LFN-YOLO: precision underwater small object detection via a lightweight reparameterized approach

Mingxin Liu[1,2], Yujie Wu[3], Ruixin Li[3] and Cong Lin[1,2]*

[1]School of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang, China, [2]Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, Zhanjiang, China, [3]College of Naval Architecture and Shipping, Guangdong Ocean University, Zhanjiang, China

Underwater object detection plays a significant role in fisheries resource assessment and ecological environment protection. However, traditional underwater object detection methods struggle to achieve accurate detection in complex underwater environments with limited computational resources. This paper proposes a lightweight underwater object detection network called LightFusionNet-YOLO (LFN-YOLO). First, we introduce the reparameterization technique RepGhost to reduce the number of parameters while enhancing training and inference efficiency. This approach effectively minimizes precision loss even with a lightweight backbone network. Then, we replaced the standard depthwise convolution in the feature extraction network with SPD-Conv, which includes an additional pooling layer to mitigate detail loss. This modification effectively enhances the detection performance for small objects. Furthermore, We employed the Generalized Feature Pyramid Network (GFPN) for feature fusion in the network's neck, enhancing the network's adaptability to features of varying scales. Finally, we design a new detection head, CLLAHead, which reduces computational costs and strengthens the robustness of the model through cross-layer local attention. At the same time, the DFL loss function is introduced to reduce regression and classification errors. Experiments conducted on public datasets, including URPC, Brackish, and TrashCan, showed that the mAP@0.5 reached 74.1%, 97.5%, and 66.2%, respectively, with parameter sizes and computational complexities of 2.7M and 7.2 GFLOPs, and the model size is only 5.9 Mb. Compared to mainstream vision models, our model demonstrates superior performance. Additionally, deployment on the NVIDIA Jetson AGX Orin edge computing device confirms its high real-time performance and suitability for underwater applications, further showcasing the exceptional capabilities of LFN-YOLO.

KEYWORDS

underwater object detection, lightweight detector, small object, marine resources, multi-scale feature fusion

1 Introduction

Underwater object detection plays a crucial role in fisheries resource assessment and the ecological environment protection. As global attention on sustainable development increases, accurately monitoring the state of underwater ecosystems and resources becomes particularly important (Grip and Blomqvist, 2020). With challenges in complex underwater environments, including insufficient lighting, clutter interference, and limited computational resources, traditional underwater object detection methods struggle to achieve optimal detection accuracy (Er et al., 2023) (Liu et al., 2023b). The above issues limit effective resource management and ecological monitoring. Therefore, developing efficient and reliable underwater object detection technology not only helps improve the accuracy of fisheries resource assessments but also provides scientific evidence for ecological protection, ensuring the sustainable development of marine ecosystems (Zhou et al., 2024).

In recent years, deep learning-based object detection technology has been widely applied in various fields. Deep learning-based object detection algorithms are generally categorized into two-stage and onestage detection algorithms. The former involves generating candidate regions first and then classifying and localizing these regions, which leads to high detection accuracy but with the downside of complex structures and low real-time performance, Notable examples include Faster R-CNN (Ren et al., 2017), R-FCN (Dai et al., 2016), and Mask R-CNN (He et al., 2017). The latter completes object detection in a single forward propagation without generating candidate regions, resulting in a simplified structure and a more lightweight model that effectively balances accuracy and speed. These methods perform well in various scenarios, with YOLO (Redmon et al., 2016), SSD (Liu et al., 2016), and RetinaNet (Ross and Dollar, 2017) are notable examples. YOLO, proposed by Joseph Redmon in 2016, transformed object detection into a single regression problem and achieved real-time object detection by dividing images into grids.

YOLOv8, the most representative algorithm in the YOLO series, strikes a good balance between accuracy and model size, making it more suitable for industrial applications. However, YOLOv8 was not specifically designed for underwater environments, leaving room for improvement in underwater detection tasks. Building upon YOLOv8, we propose the Light Fusion Net YOLO (LFN-YOLO) model to enhance the lightweight characteristics and performance of underwater target detection models. Experimental results demonstrate that the model performs exceptionally well on the URPC (Zhanjiang, 2021 China Underwater Robot Professional Contest) dataset, with a 2.2% increase in mAP@0.5 and a 19.1% reduction in GFLOPs, achieving only 7.2 GFLOPs. The parameters were reduced by 15.6%, down to 2.6M. Furthermore, LFN-YOLO demonstrated excellent performance on the Brackish dataset, achieving the highest accuracy and the smallest model size in comparison experiments with other mainstream one-stage detection algorithms. This demonstrates that LFN-YOLO strikes a better balance between accuracy and model complexity, making it suitable for underwater target detection tasks on platforms with limited hardware capabilities. The main contributions of this paper are as follows:

1) To reduce the number of network parameters and computational complexity while enhancing training and inference efficiency, a reparameterization approach is employed in the backbone network to facilitate feature reuse. Furthermore, SPD-Conv is utilized in the feature extraction process to enhance the ability to capture small object features effectively.

2) To improve the network's ability to adapt to features of varying sizes, the Generalized Feature Pyramid Network was applied for feature fusion, which effectively fuses geometric detail information from lowlevel features with semantic information from high-level features, allowing better feature extraction for underwater objects of varying sizes.

3) A lightweight detection head, CLLAHead, was designed in this paper, which incorporates a cross-layer local attention mechanism. This design reduces unnecessary computations and enhances the model's robustness in underwater environments. Additionally, the Distribution Focal Loss was introduced to minimize both regression and classification losses in target detection.

4) The proposed LFN-YOLO demonstrates superior performance in detection accuracy, network lightweight, and adaptability to underwater environments. Additionally, this paper presents an efficient underwater deployment solution. With the optimized network architecture, LFN-YOLO shows improved detection accuracy and higher FPS in real underwater scenarios.

The paper is organized as follows. Section 2 reviews the development of underwater object detection research, along with related work on lightweight networks and small object detection. Section 3 provides a detailed introduction to the network structure of LFN-YOLO, covering the overall design and the internal principles of each module. Section 4 describes the experimental setup, including datasets, evaluation metrics, equipment, and software. Section 5 presents the experimental results and analysis, including ablation and comparative experiments, as well as underwater deployment experiments. In Section 6, the generality and robustness of the LFN-YOLO model are evaluated using the TrashCan dataset. Finally, Section 7 concludes the paper.

2 Related work

2.1 Underwater object detection

In recent years, deep learning-based underwater object detection models have rapidly evolved. Many researchers have focused on developing algorithms to tackle the challenges of underwater images, which often suffer from high noise, low contrast, and color distortion (Zhang et al., 2024b). Wang et al. (2023) proposed a reinforcement learning paradigm for underwater visual enhancement, which simultaneously optimizes the target detection and visual enhancement tasks. However, the variability of underwater environments poses limitations for the visual enhancement algorithm. To address this, Wang et al. (2024) introduced a new underwater image enhancement method that can select an enhancement technique and configuration parameters based on the degree of image degradation, thereby improving the effectiveness of the

enhancement for practical applications. Additionally, underwater environments present unique challenges for object detection, such as background interference, dense object distribution, and occlusion. These issues contrast sharply with those in conventional detection scenarios, highlighting the complexity and necessity of robust underwater detection methods (Jian et al., 2021). For instance, Wang et al. (2022) proposed an enhanced YOLO network without anchor points. They utilized Retinex theory to eliminate impurities in underwater images and subsequently performed multi-scale feature fusion in the YOLO network. This approach reduced the inference time for regression and classification tasks while improving the accuracy of underwater object detection. Yan et al. (2023) proposed a dual adversarial contrastive learning enhancement network for underwater images. This network transforms degraded waters into high-quality waters and builds an inverse circulation net mapping in a self-learning manner, reducing dependency on training data and significantly enhancing the quality of underwater images. Liu et al. (2023a) proposed the YOLOv7-AC network for underwater object detection, which replaces the YOLOv7 convolution module with an ACmixBlock and incorporates global attention in the backbone network. Additionally, the K-means algorithm was employed to optimize the anchor box selection, improving both average precision and inference speed. Zhao et al. (2023) introduced the YOLOv7-CHS model, which integrates a non-contextual transformer module with parameter-free attention to learn spatial and channel relationships, resulting in enhanced detection performance. Zhang et al. (2024a) proposed the FasterNetT0 as the backbone network, reducing the number of parameters and computational complexity. They further added a small object detection head to improve accuracy for small targets, and used Deformable ConvNets and channel attention mechanisms in the neck to handle irregularly shaped and occluded objects. A comprehensive qualitative comparison of underwater object detection methods developed in recent years, as shown in Table 1.

TABLE 1 Comprehensive qualitative comparison of underwater object detection methods developed in recent years.

Method	Dataset	Backbone	Method highlights
Enhanced YOLO (Wang et al., 2022)	LED water tank image	Resnet	Retinex theory
YOLOv7-AC (Liu et al., 2023a)	URPC, Brackish	Darknet53	K-means algorithm for anchor box generation
YOLOv7-CHS (Zhao et al., 2023)	Starfish, DUO	HOSI-Darknet53	High-order spatial interaction, Contextual transformer
YOLOv8 improved (Zhang et al., 2024a)	UTDAC2020, Pascal VOC	FasterNet-T0	Deformable ConvNets
CHE-YOLO (Feng and Jin, 2024)	DUO, UTDAC2020	Darknet-53	High-order deformable attention, Enhanced spatial pyramid pooling-fast
LFN-YOLO (Ours)	URPC, Brackish	RepGhostNet	Cross-Level Local Attention, Detecting head

However, these underwater detection models primarily focus on accurate object identification, without fully considering the need for lightweight models that can be efficiently deployed in real world scenarios.

Our research aims to improve the accuracy of object detection models in underwater environments while reducing network parameters and computational complexity to enable deployment on hardware with varying performance levels. To accomplish this, we designed CLLAHead, which incorporates a cross-layer local attention mechanism (Tang and Li, 2020) and introduced the Distribution Focal Loss (DFL) (Li et al., 2023). These improvements have enhanced the model's ability to accurately identify and locate objects in underwater environments, while also reducing unnecessary computational overhead and hardware resource requirements.

2.2 Lightweight network

In recent years, the rapid development of Graphics Processing Units (GPUs) has accelerated the growth of deep neural networks (DNNs) across various fields. Simultaneously, the deployment of DNN models on resource-limited devices, such as mobile and edge devices, has become increasingly common. These devices often have constrained computational power and storage, posing challenges for DNN deployment. Balancing high accuracy with reduced model size and computational complexity is a key challenge that needs to be addressed (Xu et al., 2023a).

Currently, lightweight models are primarily achieved through two approaches: network architecture design and model compression (Lin et al., 2024). The former involves designing a more efficient network structure to reduce the number of parameters and floating-point operations (FLOPs). Popular networks in this category include MobileNet (V1, V2, V3) (Howard, 2017) (Sandler et al., 2018) (Howard et al., 2019), EfficientNet (Tan and Le, 2019), GhostNet (Han et al., 2020), and FasterNet (Chen et al., 2023a), which have significantly contributed advancing to deep learning on mobile and edge devices. On the other hand, model compression techniques—such as pruning, quantization, and knowledge distillation—focus on reducing parameters and complexity while maintaining performance.

In object detection, network architecture design is a commonly used method for lightweight. Cheng et al. (2023b) proposed replacing the YOLOv4 feature extraction backbone with the lightweight MobileViT network, effectively extracting both local and global features of objects while reducing model complexity. Shang et al. (2023) suggested using ShuffleNetv2 to replace the YOLOv5 backbone, which reduces memory access costs and convolution operations, leading to a smaller model size and faster detection speeds. Zhang et al. (2024a) introduced the FasterNet network to replace the YOLOv8 backbone for lightweight underwater object detection, aiming to reduce parameters and computational complexity while maintaining accuracy. Although these methods contribute to lightweight, they often fail to achieve a satisfactory level of accuracy. Therefore, we leverage SPD-Conv (Sunkara and Luo, 2022) for feature extraction and incorporate

RepGhost for reparameterized feature reuse (Chen et al., 2024) in the backbone network, ensuring that the underwater object detection model achieves improved accuracy while maintaining a lightweight backbone structure.

2.3 Small object detection

In object detection tasks, deep neural networks (DNNs) typically recognize objects by capturing edge features and geometric cues. However, underwater images often present significant challenges due to occlusion and the presence of small objects, making underwater object detection particularly difficult. Improving the model's ability to detect small objects is crucial for practical applications in underwater object detection.

Small objects are generally categorized into two types based on their definition: relative and absolute small objects (Tong and Wu, 2022). Relative small objects refer to targets whose area is less than 1% of the image area, while absolute small objects are defined based on fixed size thresholds. For instance, in the MS-COCO dataset, absolute small objects are defined as those with dimensions smaller than 32×32 pixels (Krishna and Jawahar, 2017). These definitions provide a basis for evaluating the performance of object detection models in various contexts, especially in scenarios with complex underwater environments.

Effective multi-scale feature fusion can significantly enhance the model's ability to detect small objects. Multi-scale feature fusion involves combining geometric details and positional information from low-level feature maps with rich semantic information from high-level feature maps. Notable methods include the Feature Pyramid Network (FPN), the Asymptotic Feature Pyramid Network (AFPN), and the Bidirectional Feature Pyramid Network (BiFPN). For example, Zhai et al. (2023) introduced a Global Attention Mechanism (GAM) into the neck of YOLOv8, enabling the network to improve the interaction of global dimension features and fuse key features, thereby increasing the speed and accuracy of small object detection. Bao et al. (2023) employed a Double Dimensional Mixed Attention (DDMA) mechanism to fuse local and non-local attention information in the YOLOv5 network, reducing the missed detections caused by densely packed small objects. Ma et al. (2024) used the Enhanced Spatial Feature Pyramid Network (ESFPN) to combine high-resolution and low-resolution semantic information, creating additional high-resolution pyramid layers to improve small object detection capabilities. However, these methods are not well-suited for the unique conditions of underwater environments. To address the challenge of misdetections and missed detections caused by the varying scales of underwater objects, we propose employing the Generalized Feature Pyramid Network (GFPN) (Jiang et al., 2022) for feature fusion. This approach effectively utilizes the feature information of small objects, enhancing the robustness of the model in detecting small objects.

3 Materials and methods

In this paper, we propose a network specifically designed for underwater object detection, which improves the detection performance of small objects while maintaining accurate detection of normal-sized objects, and reduces the model's parameter count and computational complexity. The structure of the proposed network is shown in Figure 1. First, We introduce the

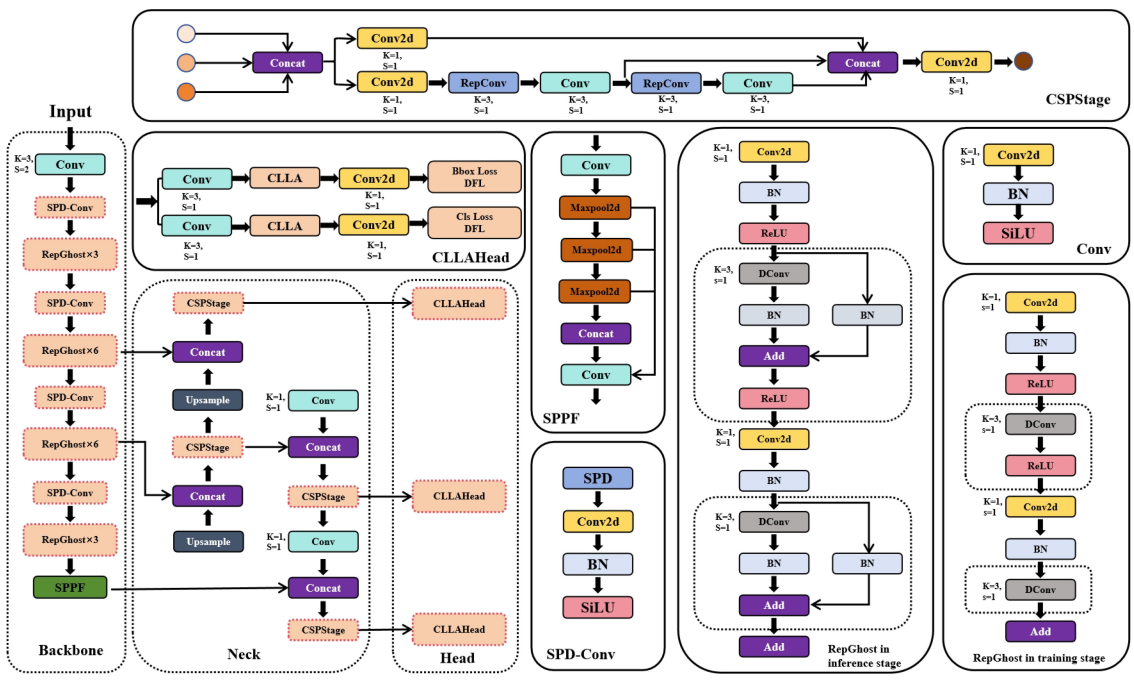


FIGURE 1
Illustration of the network structure of LFN-YOLO.

reparameterization module RepGhost into the backbone of the detection network to achieve efficient feature reuse. Second, in the feature extraction network, we replace the standard depthwise convolution with SPD-Conv to prevent the loss of detail. Then, we employed GFPN to enhance the fusion of high-level semantic information and low-level spatial information in the neck of the network. Finally, we propose a new detection head, CLLAHead, which integrates cross-layer local attention mechanisms with Distribution Focal Loss (DFL) to improve object recognition and localization, thereby forming the LFN-YOLO underwater object detection model.

3.1 RepGhost reparameterization module

Feature reuse plays an essential role in lightweight convolutional neural networks (Minaee et al., 2022). Existing feature reuse methods often use concatenation operations to reuse feature maps from different layers, which helps maintain a larger number of channels but results in a higher computational cost on hardware devices, posing challenges for real-world applications. To address this issue, we propose the RepGhost module, which uses structural reparameterization techniques to achieve feature reuse, eliminating the need for computationally expensive concatenation operations.

The RepGhost module is a lightweight convolutional module that replaces the concatenation operation used in Ghost modules with an additional operation, which is more efficient in terms of computation. The ReLU activation layer is moved behind the depthwise convolution and additional layers to conform to the rules of reparameterized structures. Lastly, a batch normalization (BN) branch is added during training, which is then fused with the depthwise convolution during inference, reducing floating-point operations. Figure 2 illustrates the reparameterization process of RepGhost.

By introducing the RepGhost module into the backbone network of YOLOv8, we can train the object detection model more efficiently. During the inference stage, this approach enhances detection speed while minimizing accuracy loss, achieving a balance between simplifying model complexity and ensuring high detection performance. This enables the model to meet the demands of object detection tasks in scenarios with limited hardware resources, making it suitable for industrial applications.

3.2 SPD-Conv

In object detection, especially when dealing with small objects, the amount of feature information is often limited. Standard stride convolutions and pooling can lead to a loss of detail, which is a major factor contributing to the low detection efficiency for small objects (Cheng et al., 2023a). To mitigate this issue, we introduce the SPD-Conv method, which replaces the standard convolution layers in the feature extraction network of YOLOv8.

The SPD-Conv is composed of a Space-to-Depth layer followed by a non-strided convolution layer. The Space-to-Depth layer downsamples the original feature map while preserving the

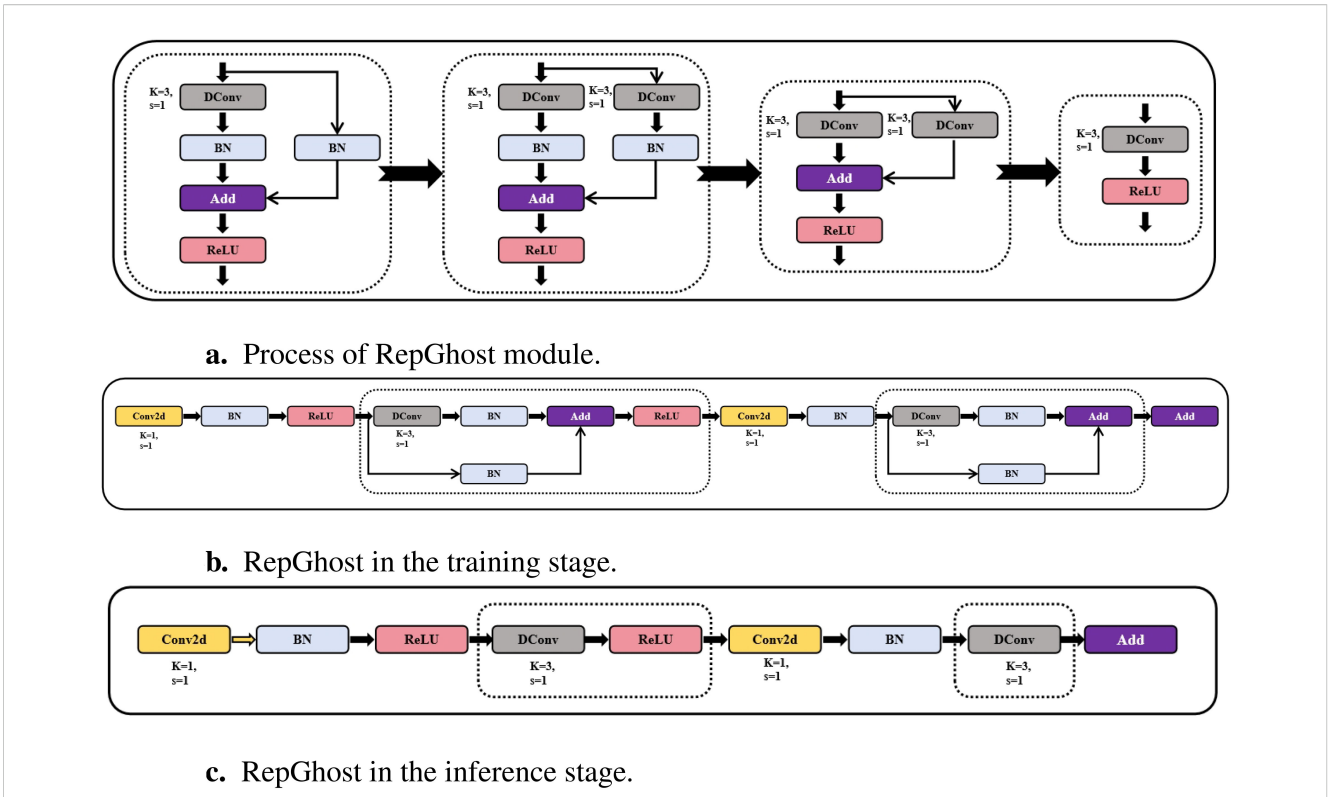


FIGURE 2
(A–C) RepGhost module details.

information in the channel dimension, with this downsampling involving only a rearrangement of the data along the channel dimension, avoiding information loss. For any intermediate feature map X of size $S \times S \times C_1$, we can generate a series of sub-feature maps according to Equation 1.

$$\begin{aligned} f_{0,0} = \mathbf{X}[0:S:\ \text{scale}\ ,0:S:\ \text{scale}\ ], f_{1,0} = \mathbf{X}[1:S:\ \text{scale}\ ,0:S:\ \text{scale}\ ],\ldots, \\ f_{\text{scale}-1,0} = \mathbf{X}[\text{scale}-1:S:\ \text{scale}\ ,0:S:\ \text{scale}\ ]; \\ f_{0,1} = \mathbf{X}[0:S:\ \text{scale}\ ,1:S:\ \text{scale}\ ], f_{1,1},\ldots, \\ f_{\text{scale}-1,1} = \mathbf{X}[\ \text{scale}-1:S:\ \text{scale}\ ,1:S:\ \text{scale}\ ]; \\ \vdots \\ f_{0,\ \text{scale}-1} = \mathbf{X}[0:S:\ \text{scale}\ ,\ \text{scale}-1:S:\ \text{scale}\ ], f_{1,\ \text{scale}-1},\ldots, \\ f_{\text{scale}-1,\ \text{scale}-1} = \mathbf{X}[\ \text{scale}-1:S:\ \text{scale},\ \text{scale}-1:S:\ \text{scale}\ ] \end{aligned} \tag{1}$$

These feature sub-maps $f_{x,y}$ are composed of all elements $\mathbf{X}(j+i)$, which are divisible by both $i+x$ and $j+i$. Therefore, each sub-map is obtained by downsampling the original feature map $\mathbf{X}$ by a scaling factor. These sub-maps are then concatenated along the channel dimension to form a new feature map $\mathbf{X}'$, where the space and dimensions are reduced by the scaling factor, and the channel dimension is increased by the square of the scaling factor. In other words, Space-to-Depth transforms $\mathbf{X}(S, S, C_1)$ into an intermediate feature map $\mathbf{X}'(S\ \text{Scale}, S\ \text{Scale}, \text{Scale}^2C_1)$. Figure 3 illustrates the process of Space-to-Depth conversion when the scaling factor is set to 2.

After applying the Space-to-Depth transformation, a non-strided (i.e., stride=1) convolution layer with C_2 filters is added, where $C_2 < \text{scale}^2C_1$. The feature map is then further transformed from $\mathbf{X}'(\frac{S}{\text{Scale}}, \frac{S}{\text{Scale}}, \text{Scale}^2C_1) \rightarrow \mathbf{X}''(\frac{S}{\text{Scale}}, \frac{S}{\text{Scale}}, \text{Scale}^2C_2)$. The reason for using non-strided convolution is to retain as much discriminative information as possible; otherwise, using stride=3 (as with a 3×3 filter) would downsample the feature map but only sample each pixel once. If stride=2 were used, asymmetric sampling would occur, with different rows or columns being sampled at different times. Generally, strides greater than 1 leads to a loss of discriminative information. Although it may appear that this process downsamples the feature map from $\mathbf{X}(S, S, C_1) \rightarrow \mathbf{X}''(\frac{S}{\text{Scale}}, \frac{S}{\text{Scale}}, \text{Scale}^2C_2)$, it fails to preserve the discriminative features of $\mathbf{X}'$.

3.3 GFPN

In the feature extraction layers, the shallow layers have small receptive fields and limited ability to represent semantic information, but they are better at capturing geometric details with high-resolution feature maps, making them suitable for perceiving position and geometric details. In contrast, deeper layers have larger receptive fields and stronger semantic representation capabilities, but they are weaker at capturing geometric information and have lower resolution feature maps (Chen et al., 2023b). Therefore, enhancing the exchange of high-level semantic information with low-level spatial information is key to handling objects of varying scales (Xiao et al., 2025). To address this, we propose a novel cross-scale feature fusion method called the Generalized Feature Pyramid Network (GFPN). GFPN aggregates features from the same and adjacent levels to enable more efficient information transfer. It also employs skip connections to prevent gradient vanishing, improving the ability of features to propagate to deeper layers. While striking a balance between model size and performance, GFPN exhibits superior performance in feature fusion. The feature fusion structure of GFPN is illustrated in Figure 4.

Since the GFPN structure is more complex compared to other feature fusion networks, its complexity increases with the depth of the layers leading to the issue of gradient vanishing. Inspired by the reparameterized GFPN used in DAMO-YOLO (Xu et al., 2023b), we adopt CSPStage to implement skip connections to replace the C2f (Cross Stage Partial Network Fusion) and combine convolutional layers, allowing information sharing between features across different spatial scales and non-adjacent semantic

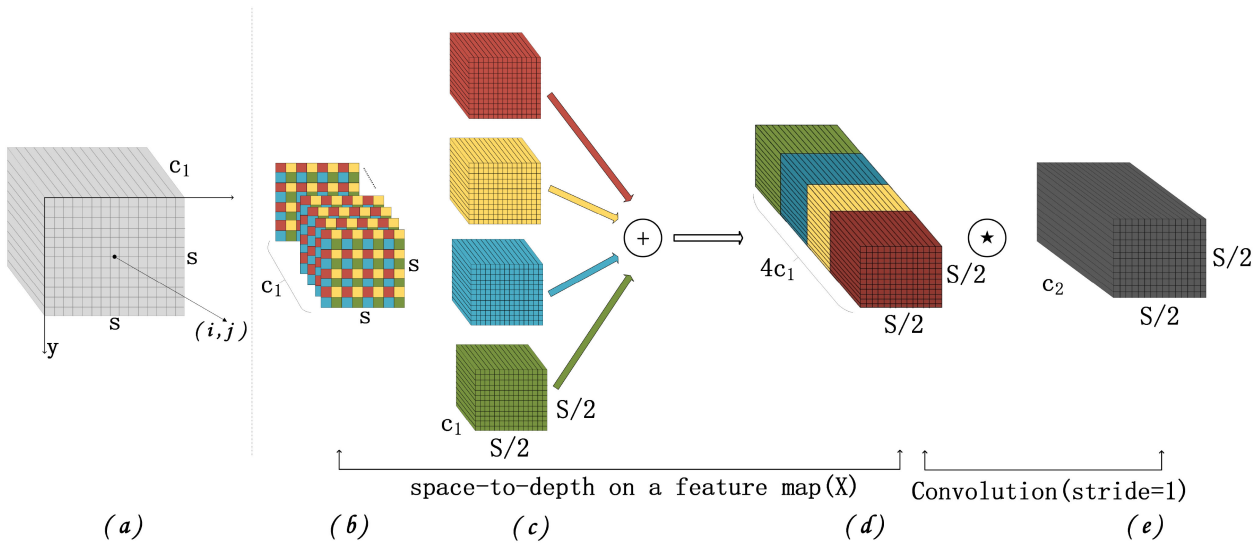


FIGURE 3
(A–E) Illustration of SPD-Conv when scale = 2.

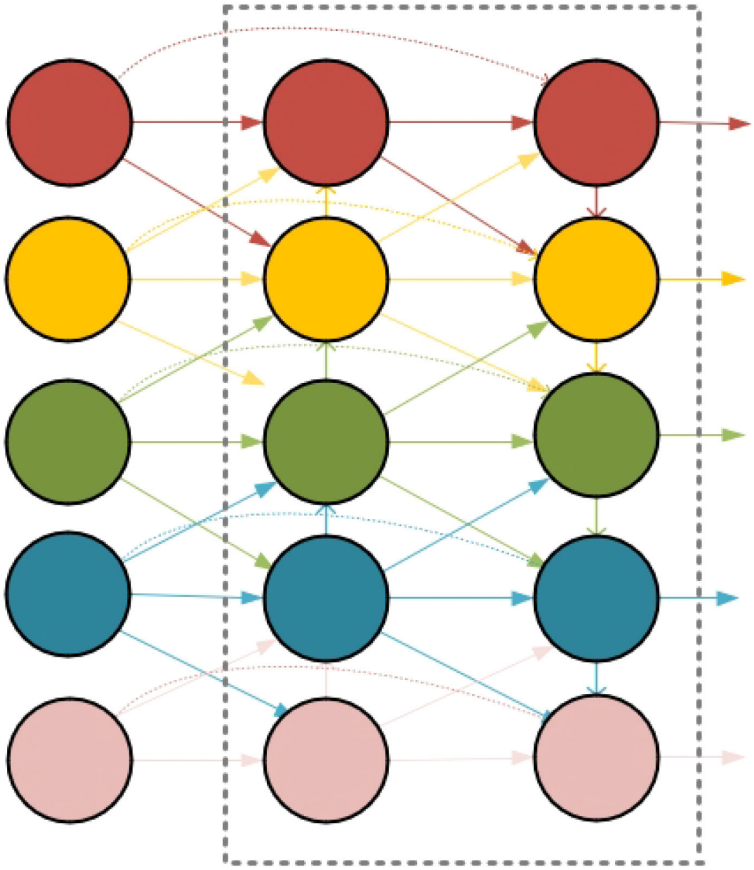


FIGURE 4
The structure of GFPN, which takes feature maps extracted from different depths as input and outputs a set of fused feature maps that encapsulate rich semantic and spatial information.

layers. This ensures that the network focuses on high-level semantic information while avoiding the loss of low-level spatial information.

The CSPStage module incorporates reparameterized convolutions (RepConv), which allow multiple computational branches to be fused during the inference phase, enhancing the efficiency and performance of the model. During training, RepConv uses multiple branches for convolution. During inference, the parameters from these branches are reparameterized into the main branch, thus reducing the computational load and memory requirements. By using CSPStage to implement skip connections, shallow feature information can be passed to deeper layers, minimizing the loss of features and enhancing the information exchange between shallow and deep layers. This improves the network's ability to adapt to targets of varying scales.

3.4 CLLAHead

In scenarios where small objects are densely packed, the original detection head of YOLOv8 struggles to meet the demands of efficient and accurate detection. Therefore, we combined the Cross-Level Local Attention (CLLA) mechanism with Distribution Focal Loss (DFL) to design CLLAHead, which enhances the model's ability to recognize and localize objects in images.

The goal of CLLA is to model the contextual relationships between cross-level features and aggregate multi-level features, as shown in Figure 5. Different levels of features typically contain different recognition information. To capture fine-grained contextual information across different feature levels and improve the detection accuracy and robustness, we embedded the CLLA module into the detection head.

The CLLA module models the relationships between the channels and spatial dimensions of high-level and low-level feature maps. Among them, $\mathbf{F}_1^{'}$ and $\mathbf{F}_2^{'}$ represent low-level and mid-level feature maps, containing shallow information (such as texture, edges, and color), while $\mathbf{F}_3^{'}$ contains valuable deep semantic information. Then, average pooling and 1×1 convolutions are applied to reduce the size of $\mathbf{F}_1^{'}$ and $\mathbf{F}_2^{'}$, unifying their spatial dimensions into new feature maps $\mathbf{F}_1^{\mathrm{l}}$ and $\mathbf{F}_2^{\mathrm{m}}$ (with the same dimensions as $\mathbf{F}_3^{'}$). Subsequently, three learnable parameters $\mathbf{W}^{\mathbf{Q}}$,$\mathbf{W}^{\mathbf{K}}$ and $\mathbf{W}^{\mathbf{V}}$ are used to project $\mathbf{F}_1^{\mathrm{l}}$, $\mathbf{F}^{\mathbf{m2}}$ into the spaces of **Q**, **K**, and **V**, as shown in the following equation:

$$\mathbf{Q} = \mathbf{F}_1^{l}\mathbf{W}^{\mathrm{Q}}, \mathbf{K} = \mathbf{F}_2^{m}\mathbf{W}^{\mathrm{K}}, \mathbf{V} = \mathbf{F}_2^{m}\mathbf{W}^{\mathrm{V}} \tag{2}$$

Next, the dot product and the softmax function are used to calculate the correlation weights between **Q** and **K**, followed by a dot product with **V** to form a new feature map. This new feature map is then added to $\mathbf{F}_3^{'}$ to finally aggregate into $\mathbf{F}_{\mathbf{M}}$, which can be

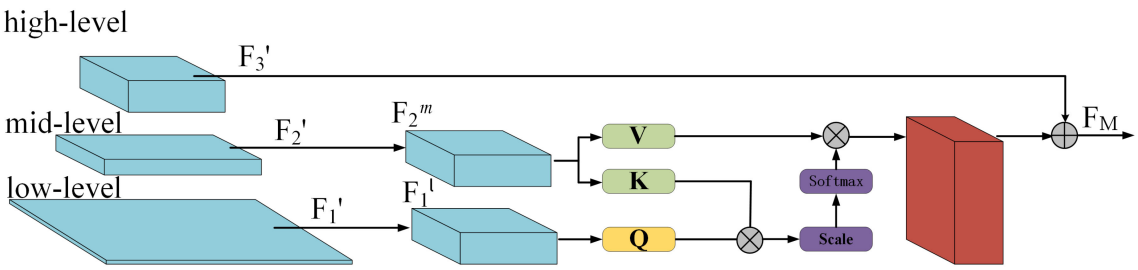


FIGURE 5
The proposed CLLA model structure.

expressed by the following equation:

$$\mathbf{F}_{\mathrm{M}} = \mathbf{F}_3^{'} + \mathrm{softmax}\left(\frac{(\mathbf{F}_1^{l}\mathbf{W}^{\mathrm{Q}})(\mathbf{F}_2^{m}\mathbf{W}^{\mathrm{K}})^{\mathrm{T}}}{\sqrt{d_k}}\right)(\mathbf{F}_2^{m}\mathbf{W}^{\mathrm{V}}) \quad (3)$$

Distribution Focal Loss (DFL) enables the network to focus on values near the target label quickly, maximizing the probability density around the label. This guides the model to pay attention to difficult-todetect targets, improving its ability to detect small objects. To optimize the probabilities at two positions near the label $y(y_i$ and $y_{i+1})$, DFL uses the cross-entropy function to concentrate the network's distribution around the target label value. The formula for DFL is given below:

$$\mathrm{DFL}_{(s_i,s_{i+1})} = -((y_{i+1} - y)\log(s_i) + (y - y_i)\log(s_{i+1})) \quad (4)$$

where s_i is the Sigmoid output of the network, and y_i and y_{i+1} represent the interval labels for $y(y_i \le y \le y_{i+1})$.

4 Experiment preparation

To evaluate and validate the detection performance of our proposed model architecture, we conducted experiments using two challenging underwater object detection datasets.

4.1 Dataset

To verify the effectiveness of the proposed method in this paper, the dataset used for the experiment contains two parts. One part is from the 2021 China Underwater Robot Professional Contest (URPC) dataset (Liu et al., 2021), which consists of 7,543 images in total. The dataset includes four categories of objects: holothurian, echinus, scallop, and starfish. The dataset presents challenges such as occlusions, overlapping objects, and small-sized underwater targets. In addition, it also includes significant color distortion caused by the absorption and scattering of light at different wavelengths underwater. This phenomenon results in a predominance of blue and green tones in the images, while red and other long-wavelength colors are heavily attenuated. These unique optical properties pose challenges in accurately recognizing and classifying underwater objects, making this dataset particularly valuable for testing methods aimed at enhancing robustness under such conditions. We randomly split the dataset into training, validation, and testing sets in a 7:2:1 ratio. Specifically, 5,280 images were used for training, 1,463 for validation, and 800 images were reserved for testing and performance evaluation. Figures 6A, B present the distribution of target counts across different categories in the URPC dataset, along with the height and width of bounding boxes. The analysis indicates that most underwater organisms within the dataset are relatively small, with the majority of bounding box dimensions falling within the range of (0-0.1, 0-0.1).

The other part is The Brackish dataset (Pedersen et al., 2019), which is a publicly available European underwater image dataset consisting of 11,205 images in total. It includes six categories of small marine organisms: crabs, normal-sized fish, small-sized fish, starfish, shrimp, and jellyfish. The Brackish dataset contains a significant number of small underwater targets. Moreover, the presence of numerous suspended particles in the water results in image blurring, reduced contrast, and even scattering artifacts, posing considerable challenges for accurate detection and recognition. The dataset also suffers from low image resolution, further complicating the detection and recognition of small marine organisms. These environmental factors make the Brackish dataset an essential benchmark for evaluating the performance of detection algorithms in turbid and low-visibility conditions. The dataset was randomly split into training, validation, and testing sets in an 8:1:1 ratio. Figures 6C, D illustrate the visual attributes of the Brackish dataset. Additionally, Figure 6E illustrates the representative underwater environments of the URPC and Brackish datasets.

4.2 Evaluation metrics

In this paper, we use Precision (P), Recall (R), mean Average Precision (mAP), Giga Floating-point Operations Per Second (GFLOPs), the number of parameters, and Frames Per Second (FPS) to evaluate the effectiveness of the model. Precision (P) reflects the accuracy of classifying positive samples, while Recall (R) indicates the effectiveness of identifying positive samples. mAP represents the mean precision across all categories. GFLOPs is a commonly used metric for measuring the computational complexity of a model, representing the number of floating-point operations executed per second. The number of parameters indicates the model's size. These metrics are widely

10.3389/fmars.2024.1513740

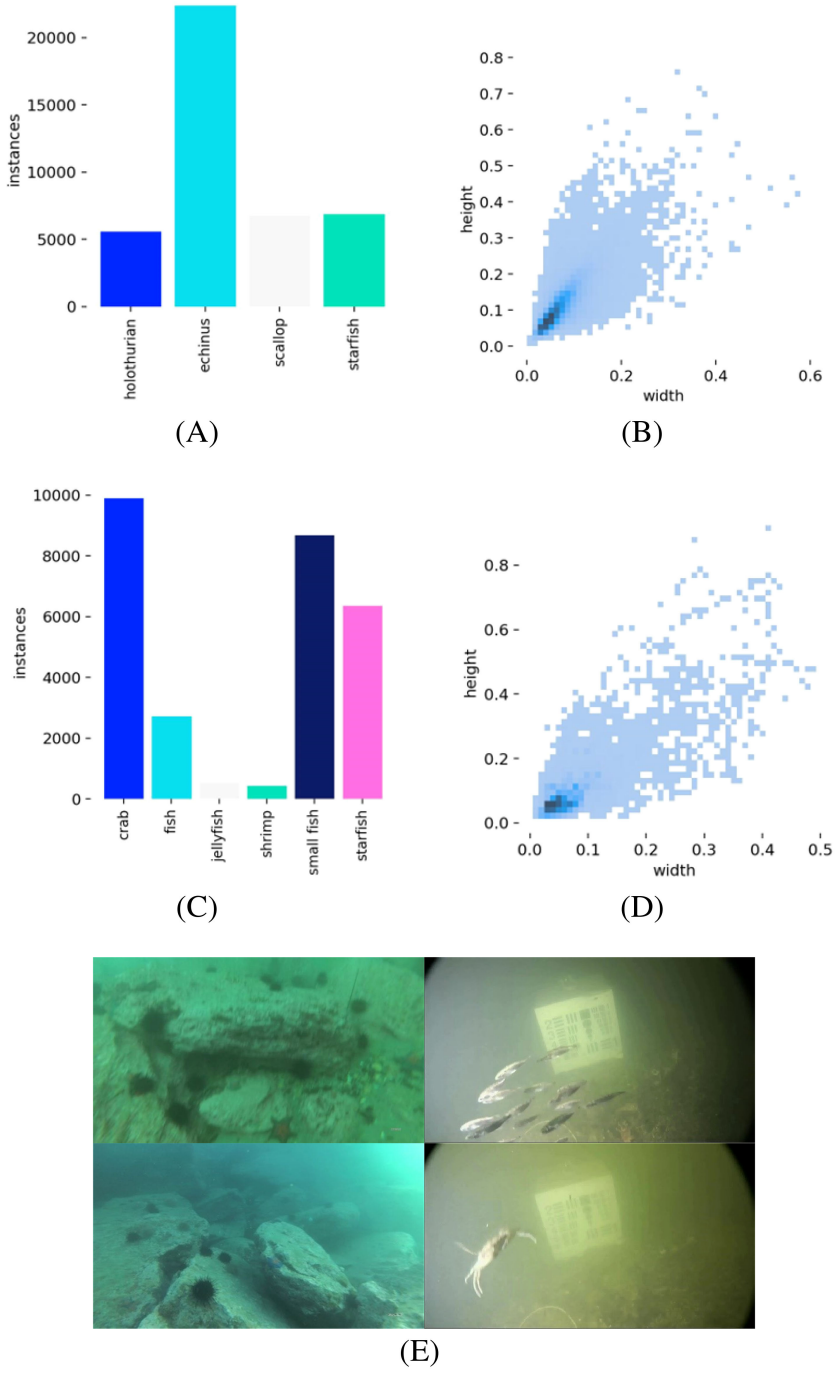


FIGURE 6
(A, B) represent the number of different species and the label size distribution in the URPC dataset, respectively. **(C, D)** show the number of different species and the label size distribution in the Brackish dataset, respectively. **(E)** illustrates the representative underwater environments of the URPC and Brackish datasets.

adopted for evaluating object detection tasks. The formulas are as follows (Equations 5–8):

$$\text{Precision} = \frac{\text{TP}}{\text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{AP} = \int_0^1 P(r)dr \tag{7}$$

$$\text{mAP} = \frac{\sum_{i=1}^{k} APi}{k} \tag{8}$$

Here, TP represents the number of true positive samples correctly predicted by the model, FP represents the number of

10.3389/fmars.2024.1513740

false positive samples, TN represents the number of true negative samples, and FN represents the number of false negative samples. P (r) represents the Precision-Recall curve, and k denotes the number of classes in the current recognition task. In object detection tasks, the mAP is determined by the selected Intersection over Union (IoU) threshold. mAP@0.5 refers to the mean Average Precision achieved by the model in object detection tasks when the IoU threshold is set to 0.5.

4.3 Experimental platform

All experiments in this study were conducted on the same computer, running the Windows 10 operating system, with an Intel® Xeon® Silver 4100 CPU, an NVIDIA GeForce RTX 2080Ti GPU, Python version 3.8, CUDA version 11.7, and PyTorch version 2.0.0. The experiments involved training the model for 150 epochs, with the batch size set to 16 and the learning rate set to 0.01. The model gradients were optimized using the SGD optimizer. The edge deployment device utilizes the NVIDIA Jetson AGX Orin with 32GB of RAM and runs the Ubuntu 20.04 Focal operating system. It uses Python 3.8, CUDA 11.4, and PyTorch v1.12.0 for GPU acceleration. For the camera, we employ the IPC5MPW underwater camera, which features a 5MP resolution, a 36mm lens, and a 2m focal length.

5 Experimental results and analysis

5.1 Ablation experiments

To validate the detection performance and model complexity of the proposed model in this study, as well as to explore the impact of specific network substructures on the model, we conducted ablation experiments based on YOLOv8n. The results are presented in Table 2, with the best results highlighted in bold.

The primary goal of the first set of experiments was to evaluate the detection capabilities of the original model. Subsequently, we conducted experiments to improve the model, both individually and collectively, using RepGhost, SPD-Conv, GFPN, and CLLAHead, to assess the effectiveness of these four enhancement techniques across the two datasets. The original model achieved precision rates of 79.7 and 96.3 on the URPC and Brackish datasets, respectively. After individually evaluating each improvement, we found that the introduction of the RepGhost module slightly reduced accuracy on the URPC dataset, but it significantly alleviated the issue of high model parameters and computational load. In the subsequent combined experiments, RepGhost had a positive impact on the performance of underwater object detection tasks. On the Brackish dataset, the RepGhost module improved both model accuracy and complexity. Additionally, we observed that GFPN resulted in noticeable performance gains on both datasets, further demonstrating the effectiveness and feasibility of GFPN's feature fusion approach for detecting small underwater objects.

Our proposed LFN-YOLO network achieved accuracy rates of 82.1% and 97.4% on the URPC and Brackish datasets, respectively, representing improvements of 2.4% and 1.1% compared to the original model. Furthermore, the parameter counts and GFLOPs

TABLE 2 Ablation study on detection performance and complexity.

Group	RepGhost	SPD-Conv	GFPN	CLLAHead	URPC(%)				Brackish(%)				Parameters/M	GFLOPs
					P	R	mAP@0.5	mAP@0.5:0.95	P	R	mAP@0.5	mAP@0.5:0.95		
1					79.7	64.2	71.9	39.6	96.3	94.2	96.9	78.5	3.2	8.9
2	✓				78.9	63.7	71.5	39.8	97.3	94.4	97.2	78.6	**2.6**	**7.0**
3		✓			80.0	65.4	72.8	40.6	97.1	94.8	97.4	78.8	2.8	7.6
4			✓		81.5	64.5	72.5	40.3	97.0	95.2	97.6	79.1	3.1	8.1
5				✓	81.1	64.8	72.3	40.8	97.1	94.4	96.8	78.9	3.0	7.7
6	✓	✓			80.3	65.1	72.6	40.7	96.8	94.9	97.1	79.2	2.9	7.5
7	✓	✓	✓		81.4	65.4	73.8	41.5	97.3	95.1	**97.7**	79.5	2.8	7.8
8	✓	✓	✓	✓	**82.1**	**65.7**	**74.1**	**42.1**	**97.4**	**95.4**	97.5	**79.8**	2.7	7.2

'✓' represents the introduced module; with the best results highlighted in bold.

were reduced by 15.6% and 19.1%, respectively. In addition, the network demonstrated an increase in recall and mAP@0.5 by 1.5% and 2.2% on the URPC dataset and by 1.3% and 0.6% on the Brackish dataset. These results highlight the adaptability and robustness of the LFN-YOLO model in different underwater environments. Figure 7 illustrates a comparison of the detection performance between the original YOLOv8 model and the improved LFN-YOLO model for underwater object detection.

The interactions between these improvements are realized through their complementary characteristics and advantages, working synergistically to reinforce each other and ultimately form a collaborative network for underwater object detection tasks. To validate the effectiveness of the proposed model in enhancing underwater detection performance, we compared the mAP@0.5 and box loss fitting curves of LFN-YOLO and YOLOv8n over 150 training epochs, as shown in Figures 8A-D. Additionally, Figures 8E, F presents a detailed comparison of detection precision for various organisms in the dataset between LFN-YOLO and YOLOv8n. Experimental results demonstrate that LFN-YOLO significantly outperforms YOLOv8n in detecting small objects, such as echinus, scallops, jellyfish, and small fish. This demonstrates LFN-YOLO's exceptional ability to capture fine

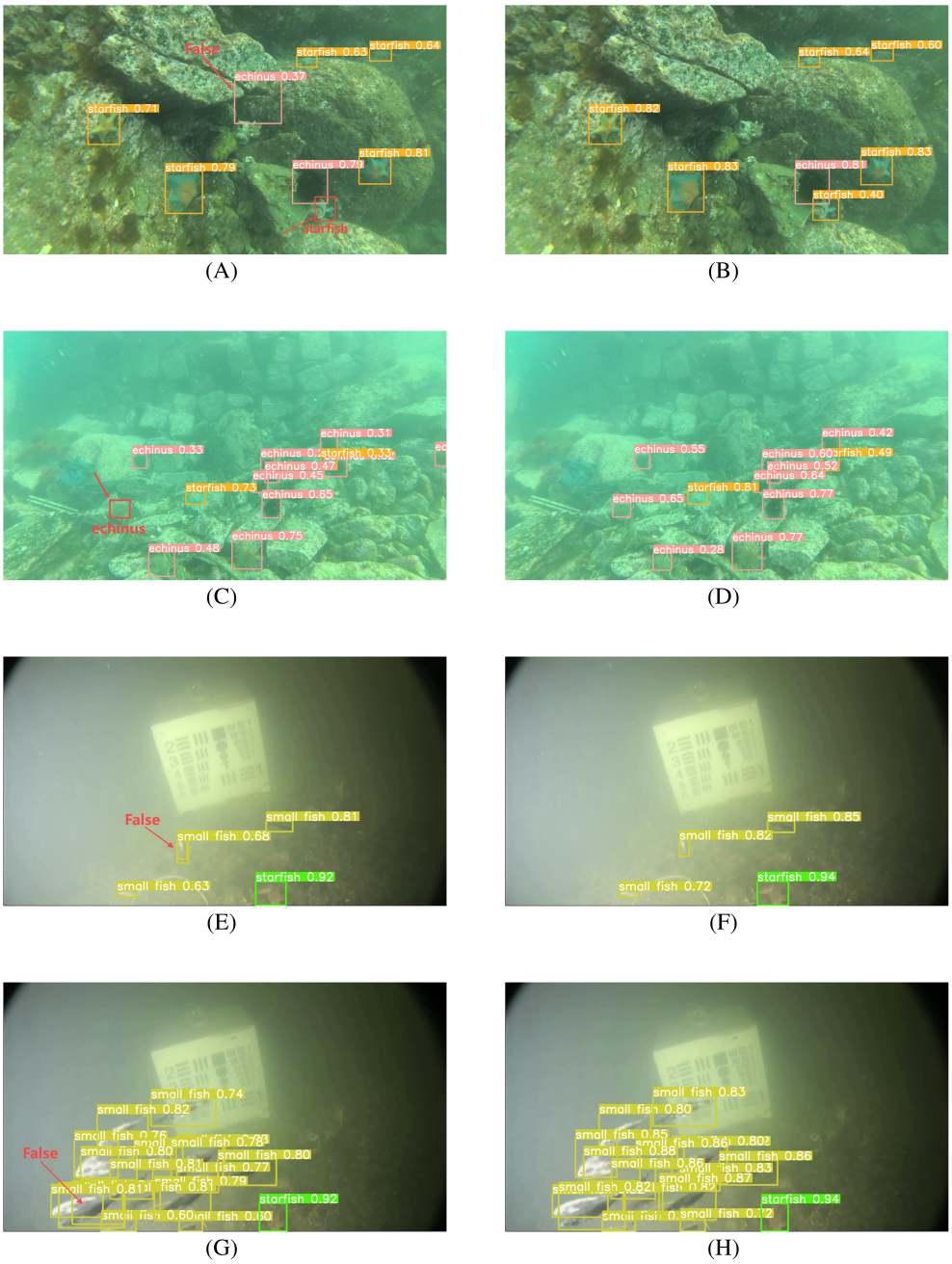


FIGURE 7
Comparison of detection results of different methods **(A, C, E, G)** results from YOLOv8, and **(B, D, F, H)** results from LFN-YOLO.

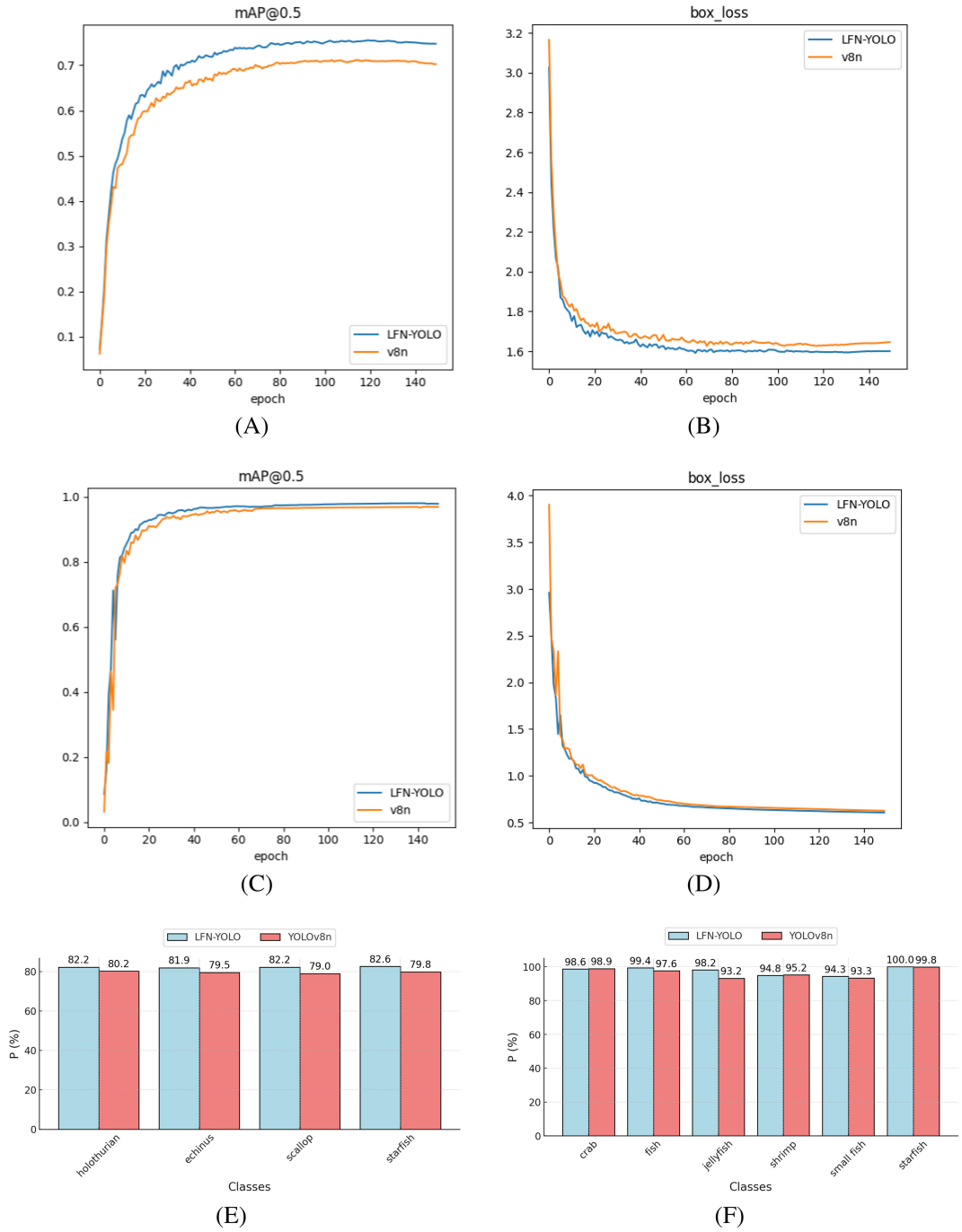


FIGURE 8
(A, B) represent the fitting curves during training on the URPC dataset, while **(C, D)** show the fitting curves for the Brackish dataset. **(E, F)** respectively illustrate the accuracy comparison for different species in the URPC and Brackish datasets.

details and detect small-scale underwater organisms, which is crucial for achieving accurate underwater object detection.

5.2 Comparative experiments

To further demonstrate that the LFN-YOLO model achieves a better balance between model complexity and accuracy, we conducted comparative experiments with eight other mainstream one-stage detection models. All experiments were carried out under the same settings, evaluating the models based on accuracy, recall, mAP@0.5, number of parameters, computational complexity, and model size. The detailed results are presented in Table 3. All the algorithms compared in the experiments meet the real-time monitoring requirements and LFN-YOLO exhibits higher detection accuracy while maintaining a lighter model complexity.

Based on the experimental results in Table 3, LFN-YOLO demonstrates outstanding performance on both the URPC and

Brackish datasets. On the URPC dataset, LFN-YOLO achieved an mAP@0.5 of 74.1%, surpassing YOLOv5n, YOLOv6-N, YOLOv8n, YOLOv10n, and YOLOv11 by 2.3%, 3.4%, 2.2%, 3.3%, and 0.5%, respectively. For the more rigorous mAP@0.5:0.95 evaluation metric, LFN-YOLO remains the best among the YOLO series models, showcasing its exceptional capability in detecting underwater targets. Additionally, LFN-YOLO achieves an inference speed of 58 FPS, highlighting its significant advantage in real-time detection tasks. In contrast, the SSD model, with VGG-16 as its backbone, demonstrated the best performance on the URPC dataset, owing to the unique distribution of objects and scene characteristics in the URPC dataset.; however, its poor performance on the Brackish dataset reveals a lack of generalization and robustness, which are essential qualities for underwater object detection models. Moreover, SSD, being relatively large among one-stage algorithms, is not suitable for underwater target recognition tasks on unmanned platforms. Notably, among the one-stage algorithms included in our comparative experiments, only RetinaNet requires higher hardware performance to meet the real-time demands of underwater detection tasks, as its lower FPS makes it unsuitable for real-time underwater target detection.

Figure 9 presents a comparison of the detection results for underwater objects using the seven models with the best performance from Table 3. We used the same RGB color for the detection boxes of all YOLO series algorithms for easier comparison. YOLOv5n exhibited lower confidence scores for detected objects, YOLOv11n showed instances of missed detections, while YOLOv7tiny, YOLOv8n, and SSD suffered from false positives. YOLOv9t experienced both false positives and missed detections. In contrast, LFN-YOLO not only accurately identified the underwater objects but also achieved higher confidence scores. Among the networks compared, LFN-YOLO boasts the most lightweight structure.

5.3 Model edge deployment

With the increasing computational capabilities of edge deployment devices, deep learning-based object detection tasks are now more feasible. In this study, we deploy LFN-YOLO on the NVIDIA Jetson AGX Orin edge computing device to further validate its applicability in underwater platforms. To this end, we selected fish fry as the target for recognition, with the recognition scenario being a 200L fish tank. Our algorithm is deployed and tested underwater, as shown in Figure 10A, which illustrates the hardware connection diagram.

First, we trained both the LFN-YOLO and YOLOv8n models using the Brackish dataset on a PC and transferred the trained models to the project directory on the NVIDIA Jetson AGX Orin. Subsequently, an underwater camera was placed in the tank and connected to the RJ45 port of the NVIDIA Jetson AGX Orin, and an external monitor was connected via the HDMI port on the Jetson AGX Orin. Finally, a pre-written Python script was used to capture the underwater camera feed as network input, run predictions, and display real-time data. The real-time detection results of LFN-YOLO are shown in Figures 10B, C.

TABLE 3 Comparison experiments of LFN-YOLO on URPC and Brackish datasets, with the best results highlighted in bold, and the second-best results highlighted in red.

Model	URPC(%)					Brackish(%)					Parameters/M	GFLOPs	Model size/Mb
	P	R	mAP@0.5	mAP@0.5:0.95	FPS	P	R	mAP@0.5	mAP@0.5:0.95	FPS			
YOLOv5n	77.9	64.3	71.8	39.7	44	96.1	94.1	96.9	77.3	51	2.7	7.8	5.0
YOLOv6-N	78.0	62.1	70.7	39.0	57	96.6	92.3	96.4	76.1	61	4.5	11.9	8.3
YOLOv7tiny	80.6	64.9	73.2	40.9	48	95.7	93.1	96.5	75.1	58	6.2	13.2	12.0
YOLOv8n	79.7	64.2	71.9	39.6	52	96.3	94.2	96.9	79.0	59	3.2	8.9	6.0
YOLOv9t	80.2	64.6	73.1	41.1	37	97.3	95.0	97.5	78.5	39	**2.0**	7.9	8.8
YOLOv10n	75.8	63.9	70.8	39.0	42	96.2	94.3	97.0	79.1	42	2.7	8.4	5.5
YOLOv11n	78.9	65.2	72.1	41.1	54	96.9	94.0	97.2	78.2	60	2.6	**6.5**	5.3
RetinaNet	75.2	66.8	73.4	41.4	18	94.8	94.6	96.5	75.9	22	36.2	206.0	80.1
RT-DETR	74.3	61.3	68.0	38.6	20	97.8	94.8	97.2	79.3	22	32.8	109.0	63.4
SSD	**84.4**	**68.4**	**75.4**	**42.3**	51	92.5	92.8	95.8	76.4	57	26.4	116.2	92.1
LFN-YOLO	82.1	65.7	74.1	42.1	**58**	**97.4**	**95.4**	**97.5**	**79.8**	**63**	2.7	7.2	5.7

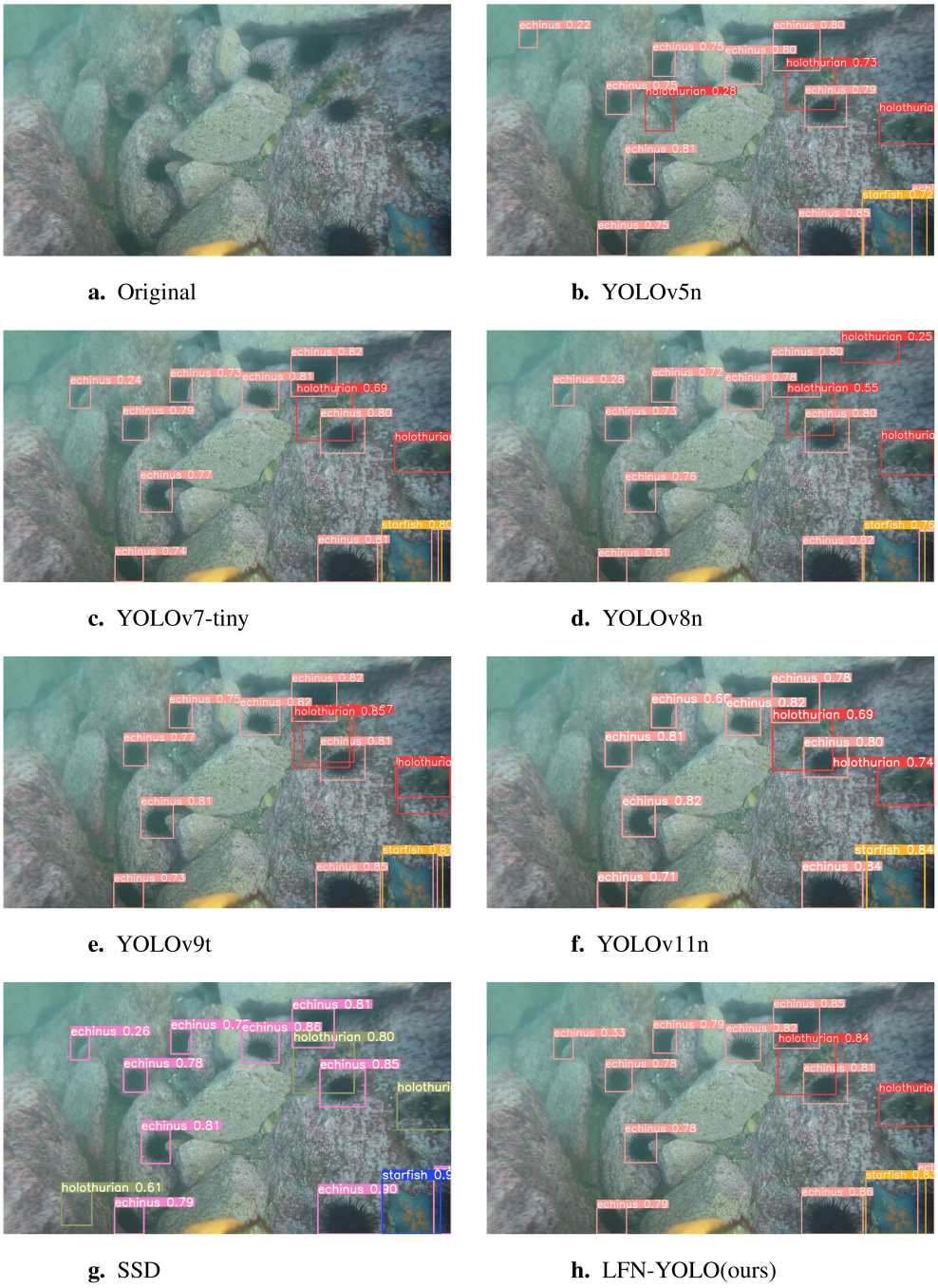


FIGURE 9
(A–H) Presentation of detection results from seven advanced models.

To evaluate the practical applicability of LFN-YOLO, we compared its detection performance against that of YOLOv8n. As illustrated in Figure 11, YOLOv8n exhibited notable deficiencies, including frequently missed detections and low confidence scores. In contrast, LFN-YOLO demonstrated superior performance, particularly in terms of real-time detection speed, where it outpaced YOLOv8n by a substantial margin. These results demonstrate that our method performs stably in underwater environments, enabling real-time detection and recognition of underwater targets, thus validating the effectiveness and practicality of the algorithm.

6 Discussion

The LFN-YOLO model excels in underwater object detection, particularly in terms of accuracy and lightweight design. This can be attributed to our specially designed deep learning-based object detection network, which seamlessly integrates various modules to address the inherent challenges of the underwater environment. Importantly, the architecture of the LFN-YOLO network was not specifically designed for our experimental dataset, highlighting the network's broad applicability and strong generalization capabilities

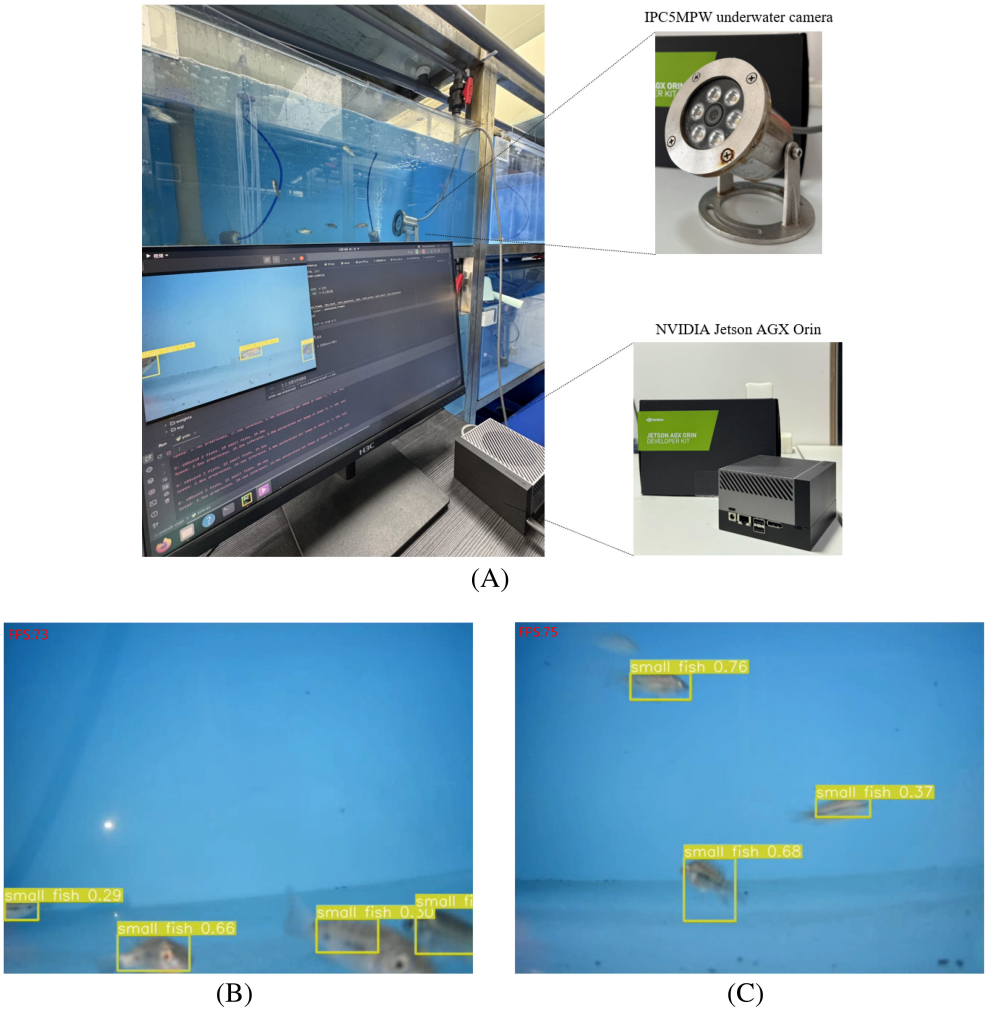


FIGURE 10
(A) is the hardware connection diagram. **(B, C)** present the recognition results of the LFNYOLO deployment.

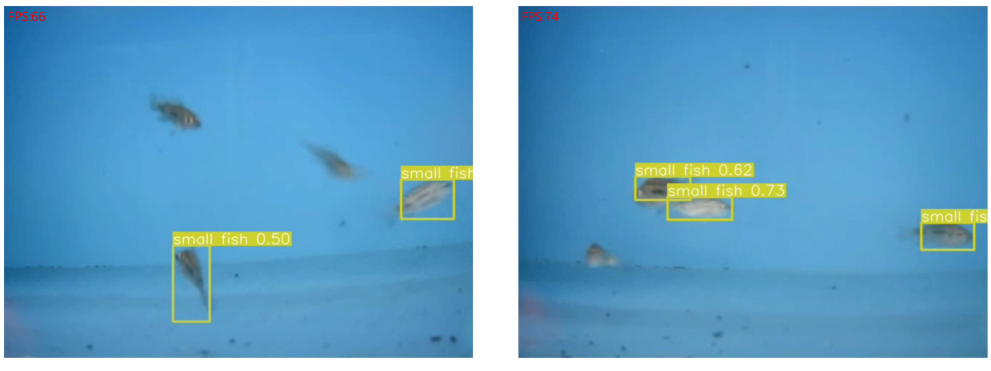


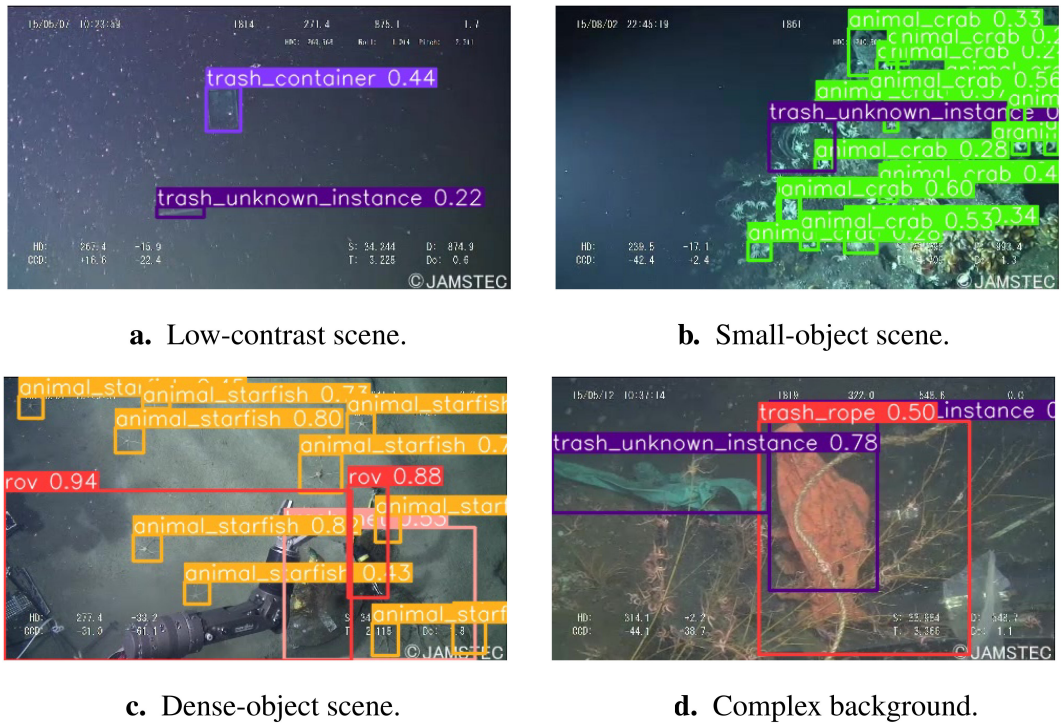
FIGURE 11
(A, B) Comparison of recognition results for edge deployment.

in underwater tasks. To further evaluate its effectiveness, we compared LFN-YOLO with other leading object detection algorithms using the TrashCan dataset (Hong et al., 2020). This dataset comprises 7,212 underwater images, featuring 22 categories of underwater objects such as debris, ROVs, and various marine species. The images were sourced from the J-EDI (JAMSTEC E-library of Deep-sea Images), managed by the Japan Agency for Marine-Earth Science and Technology (JAMSTEC).

Table 4 presents the comprehensive experimental results of LFN-YOLO compared to other models on the TrashCan dataset. Our model achieved a mAP@0.5 of 66.2%, demonstrating a significant advantage in accuracy over other algorithms, while also being more lightweight in terms of model parameters and GFLOPs compared to many real-time detection algorithms. Figure 12 shows the detection results of LFN-YOLO across four different underwater environments in the TrashCan dataset. In

TABLE 4 Experimental results of LFN-YOLO and other object detection models on the TrashCan dataset, with the best results highlighted in bold.

Model	Parameters/M	GFLOPs	mAP@0.5/%	mAP@0.5:0.95/%
Faster R-CNN	41.4	135	55.3	38.2
RT-DETR	32.8	109	61.4	44.2
YOLOv5n	2.7	7.8	61.7	43.7
YOLOv6-N	4.5	11.9	58.8	41.6
YOLOv7-tiny	6.2	13.2	65.9	45.0
YOLOv8n	3.2	8.9	64.1	45.8
YOLOv9t	**2.0**	7.9	63.9	45.6
YOLOv10n	2.7	8.4	61.0	43.6
YOLOv11n	2.6	**6.5**	63.4	45.3
SSD	26.3	116.2	58.1	40.4
LFN-YOLO	2.7	7.2	**66.2**	**47.1**



a. Low-contrast scene. **b.** Small-object scene.

c. Dense-object scene. **d.** Complex background.

FIGURE 12
(A–D) Detection results of LFN-YOLO in four different underwater environments.

particularly challenging conditions where human vision struggles to discern objects that blend almost seamlessly with their surroundings, LFN-YOLO can accurately and effectively detect these targets. As illustrated in Figures 12A, B, even in scenarios with extremely low contrast and densely clustered small objects, the model successfully identifies underwater targets. Similarly, in Figures 12C, D, LFN-YOLO demonstrates strong robustness in detecting objects in complex backgrounds with occlusions.

Although our model has made significant progress in advancing the deployment of underwater object detection systems, LFN-YOLO still faces challenges with false positives and missed detections in highly variable underwater environments. As shown in Figure 13, using the URPC dataset as an example, LFN-YOLO struggles with small object detection in complex backgrounds due to limitations in feature extraction, leading to missed detections. Additionally, under low-resolution conditions, such as those represented by the Brackish dataset, small object detection is easily affected by occlusion and insufficient resolution, resulting in inaccurate localization. Furthermore, in scenarios with large variations in object scale, such as the TrashCan dataset, LFN-YOLO still needs improvement in detecting targets with significant scale changes in underwater images.

7 Conclusion

The detection of small underwater organisms is of great significance for marine life sciences and resource exploration. This paper proposes a lightweight underwater object detection model based on deep learning, which achieves both lightweight design and high accuracy while demonstrating excellent generalization and robustness, essential qualities of a strong model. Firstly, the model introduces a lightweight re-parameterization technique, RepGhost, to achieve

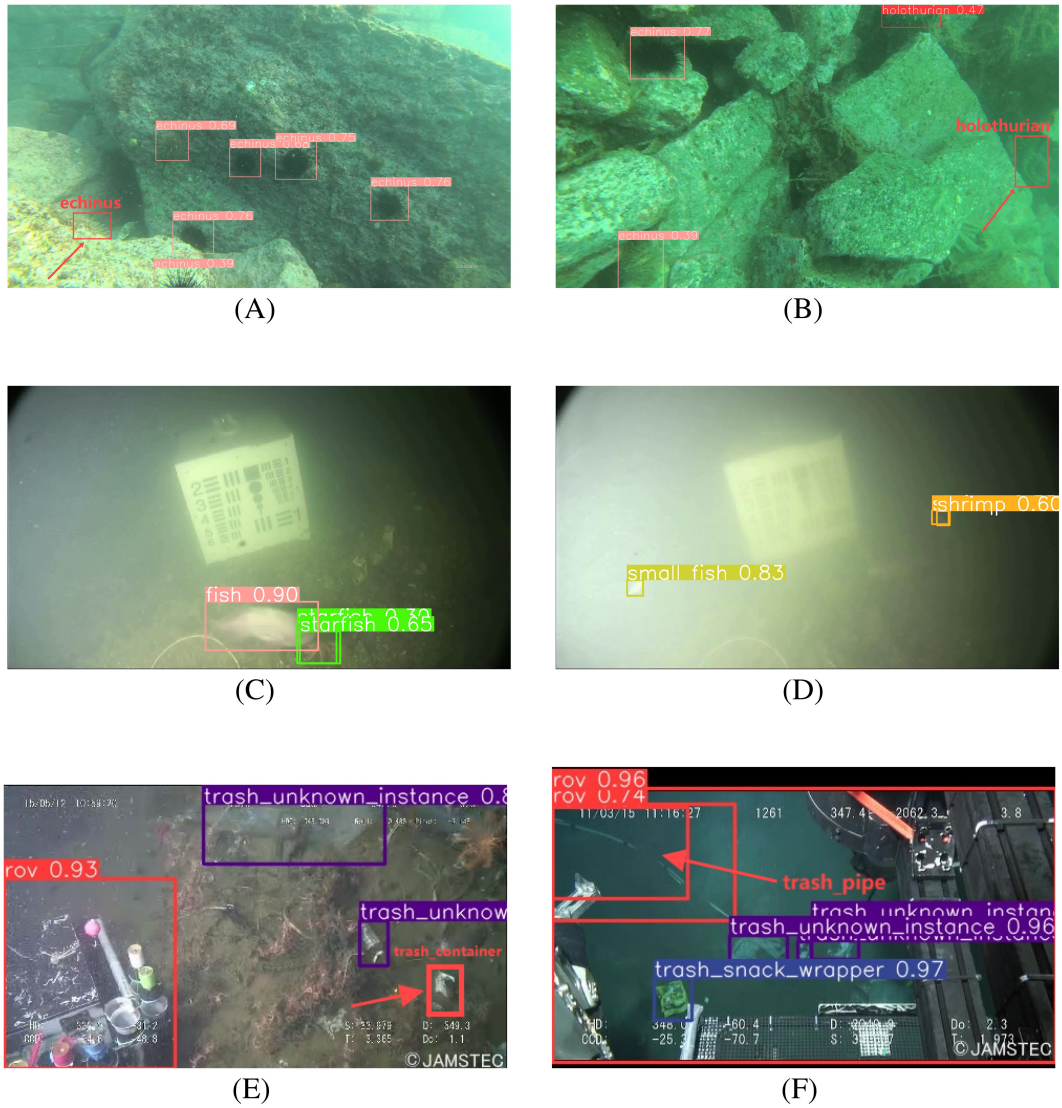


FIGURE 13
Presentation of representative failure cases. Panels **(A, B)** show missed detections in complex backgrounds from the URPC dataset. Panels **(C, D)** illustrate misdetections in low-resolution images from the Brackish dataset. Panels **(E, F)** depict missed and misdetections due to significant scale variations in the TrashCan dataset.

feature reuse, reduce the number of parameters, and improve both training efficiency and inference speed, minimizing the accuracy loss while maintaining a lightweight backbone network. The feature extraction network is further enhanced by incorporating SPD-Conv convolution modules, which improves the effective extraction of small object features. Secondly, to address challenges such as small object size, dense distribution, and blurry imaging in underwater visible light conditions, we propose a GFPN (General Feature Pyramid Network) for feature fusion, enabling effective extraction of features across varying object scales. Finally, cross-layer local attention mechanisms are added to the detection head to reduce unnecessary computations and enhance model robustness. A DFL (Distribution Focal Loss) is also introduced to minimize regression and classification losses. LFN-YOLO achieves strong detection results on the URPC, Brackish, and TrashCan datasets, with mAP@0.5 scores of 82.2%, 97.5%, and 66.2%, respectively, improving upon YOLOv8 by 2.6%, 1.2%, and 2.1%. Meanwhile, the model reduces parameters and GFLOPs by 15.6% and 19.1%, meeting the requirements for both lightweight design and high precision. This makes it suitable for small underwater object detection and marine species diversity surveys. In the future, we will explore underwater multi-source information fusion, specifically by integrating underwater visible light images with various underwater sensors, such as sonar, to enable the model to perform underwater exploration tasks in low-light or no-light conditions. This approach aims to further enhance the model's generalization capability and adaptability to diverse environments. At the same time, we will optimize the model end-to-end to improve its real-time detection capabilities. This will not only assist researchers in conducting more efficient marine resource surveys but also provide robust technological support for underwater ecological conservation.

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

ML: Conceptualization, Funding acquisition, Methodology, Software, Writing – original draft, Writing – review & editing. YW: Methodology, Software, Writing – original draft, Writing – review & editing. RL: Software, Writing – original draft, Writing – review & editing. CL: Conceptualization, Project administration, Writing – original draft, Writing – review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was partly supported by the National Natural Science Foundation of China (62171143), Guangdong Provincial University Innovation Team (2023KCXTD016), special projects in key fields of ordinary universities in Guangdong, Province (2021ZDZX1060), the Stable Supporting Fund of Acoustic Science and Technology Laboratory (JCKYS2024604SSJS00301), the Undergraduate Innovation Team Project of Guangdong Ocean University under Grant CXTD2024011, the Open Fund of Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching (Grant NO. 2023B1212030003).

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.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

References

Bao, W., Zhu, Z., Hu, G., Zhou, X., Zhang, D., and Yang, X. (2023). Uav remote sensing detection of tea leaf blight based on ddma-yolo. *Comput. Electron. Agric.* 205, 107637. doi: 10.1016/j.compag.2023.107637

Chen, C., Guo, Z., Zeng, H., Xiong, P., and Dong, J. (2024). Repghost: A hardware-efficient ghost module via re-parameterization. Available online at: https://arxiv.org/abs/2211.06088.

Chen, J., Kao, S.-h., He, H., Zhuo, W., Wen, S., Lee, C.-H., et al. (2023a). "Run, don't walk: Chasing higher flops for faster neural networks," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (Piscataway, NJ, USA: IEEE), 12021–12031. doi: 10.1109/CVPR52729.2023.01157

Chen, S., Zhao, J., Zhou, Y., Wang, H., Yao, R., Zhang, L., et al. (2023b). Info-fpn: An informative feature pyramid network for object detection in remote sensing images. *Expert Syst. Appl.* 214, 119132. doi: 10.1016/j.eswa.2022.119132

Cheng, G., Yuan, X., Yao, X., Yan, K., Zeng, Q., Xie, X., et al. (2023a). Towards large-scale small object detection: Survey and benchmarks. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 13467–13488. doi: 10.1109/TPAMI.2023.3290594

Cheng, Q., Li, X., Zhu, B., Shi, Y., and Xie, B. (2023b). Drone detection method based on mobilevit and ca-panet. *Electronics* 12, 223. doi: 10.3390/electronics12010223

Dai, J., Li, Y., He, K., and Sun, J. (2016). "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*,

vol. 29 . Eds. D. Lee, M. Sugiyama, U. Luxburg, I. Guyon and R. Garnett (Red Hook, NY, USA: Curran Associates, Inc).

Er, M. J., Chen, J., Zhang, Y., and Gao, W. (2023). Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review. *Sensors* 23, 1990. doi: 10.3390/s23041990

Feng, J., and Jin, T. (2024). Ceh-yolo: A composite enhanced yolo-based model for underwater object detection. *Ecol. Inf.* 82, 102758. doi: 10.1016/j.ecoinf.2024.102758

Grip, K., and Blomqvist, S. (2020). Marine nature conservation and conflicts with fisheries. *Ambio* 49, 1328–1340. doi: 10.1007/s13280-019-01279-7

Han, K., Wang, Y., 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 (CVPR)*. (Piscataway, New Jersey, USA: IEEE), 1577–1586. doi: 10.1109/CVPR42600.2020.00165

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (Piscataway, New Jersey, USA: IEEE).

Hong, J., Fulton, M., and Sattar, J. (2020). Trashcan: A semantically-segmented dataset towards visual detection of marine debris. Available online at: https://arxiv.org/abs/2007.08097.

Howard, A. G. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., et al. (2019). Searching for mobilenetv3. *Proc. IEEE/CVF Int. Conf. Comput. Vision (ICCV)*, 1314–1324. doi: 10.1109/ICCV43118.2019

Jian, M., Liu, X., Luo, H., Lu, X., Yu, H., and Dong, J. (2021). Underwater image processing and analysis: A review. *Signal Process.: Image Commun.* 91, 116088. doi: 10.1016/j.image.2020.116088

Jiang, Y., Tan, Z., Wang, J., Sun, X., Lin, M., and Li, H. (2022). Giraffedet: A heavy-neck paradigm for object detection. Available online at: https://arxiv.org/abs/2202.04256.

Krishna, H., and Jawahar, C. (2017). "Improving small object detection," in *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*. (New York, NY, USA: IEEE), 340–345. doi: 10.1109/ACPR.2017.149

Li, X., Lv, C., Wang, W., Li, G., Yang, L., and Yang, J. (2023). Generalized focal loss: Towards efficient representation learning for dense object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 3139–3153. doi: 10.1109/TPAMI.2022.3180392

Lin, C., Mao, X., Qiu, C., and Zou, L. (2024). Dtcnet: Transformer-cnn distillation for super-resolution of remote sensing image. *IEEE J. Select. Topics Appl. Earth Observ. Remote Sens.* 17, 11117–11133. doi: 10.1109/JSTARS.2024.3409808

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*. Eds. B. Leibe, J. Matas, N. Sebe and M. Welling (Springer International Publishing, Cham), 21–37. doi: 10.1007/978-3-319-46448-02

Liu, M., Jiang, W., Hou, M., Qi, Z., Li, R., and Zhang, C. (2023b). A deep learning approach for object detection of rockfish in challenging underwater environments. *Front. Mar. Sci.* 10, 1242041. doi: 10.3389/fmars.2023.1242041

Liu, C., Li, H., Wang, S., Zhu, M., Wang, D., Fan, X., et al. (2021). "A dataset and benchmark of underwater object detection for robot picking," in *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*. (New York, NY, USA: IEEE), 1–6. doi: 10.1109/ICMEW53276.2021.9455997

Liu, K., Sun, Q., Sun, D., Peng, L., Yang, M., and Wang, N. (2023a). Underwater target detection based on improved yolov7. *J. Mar. Sci. Eng.* 11, 677. doi: 10.3390/jmse11030677

Ma, P., He, X., Chen, Y., and Liu, Y. (2024). Isod: Improved small object detection based on extended scale feature pyramid network. *Visual Comput.* 40, 1–15. doi: 10.1007/s00371-024-03341-2

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2022). Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 3523–3542. doi: 10.1109/TPAMI.2021.3059968

Pedersen, M., Bruslund Haurum, J., Gade, R., and Moeslund, T. B. (2019). "Detection of marine animals in a new underwater dataset with varying visibility," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. (Long Beach, California, USA: IEEE), 18–26.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Piscataway, NJ, USA: IEEE), 779–788. doi: 10.1109/CVPR.2016.91

Ren, S., He, K., Girshick, R., and Sun, J. (2017). 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

Ross, T.-Y., and Dollar, G. (2017). "Focal loss for dense object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Piscataway, NJ, USA: IEEE), 2980–2988.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (Piscataway, NJ, USA: IEEE), 4510–4520. doi: 10.1109/CVPR.2018.00474

Shang, Y., Xu, X., Jiao, Y., Wang, Z., Hua, Z., and Song, H. (2023). Using lightweight deep learning algorithm for real-time detection of apple flowers in natural environments. *Comput. Electron. Agric.* 207, 107765. doi: 10.1016/j.compag.2023.107765

Sunkara, R., and Luo, T. (2022). "No more strided convolutions or pooling: A new cnn building block for lowresolution images and small objects," in *Machine Learning and Knowledge Discovery in Databases*. Eds. M.-R. Amini, S. Canu, A. Fischer, T. Guns, P.K. Novak and G. Tsoumakas (Springer Nature Switzerland, Cham), 443–459.

Tan, M., and Le, Q. (2019). "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 . Eds. K. Chaudhuri and R. Salakhutdinov (Cambridge, MA, USA: PMLR), 6105–6114.

Tang, L., and Li, B. (2020). "Class: Cross-level attention and supervision for salient objects detection," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*. (Heidelberg, Germany: Springer).

Tong, K., and Wu, Y. (2022). Deep learning-based detection from the perspective of small or tiny objects: A survey. *Image Vision Comput.* 123, 104471. doi: 10.1016/j.imavis.2022.104471

Wang, X., Jiang, X., Xia, Z., and Feng, X. (2022). "Underwater object detection based on enhanced yolo," in *2022 International Conference on Image Processing and Media Computing (ICIPMC)*. (New York, USA: IEEE), 17–21. doi: 10.1109/ICIPMC55686.2022.00012

Wang, H., Sun, S., Bai, X., Wang, J., and Ren, P. (2023). A reinforcement learning paradigm of configuring visual enhancement for object detection in underwater scenes. *IEEE J. Ocean. Eng.* 48, 443–461. doi: 10.1109/JOE.2022.3226202

Wang, H., Zhang, W., and Ren, P. (2024). Self-organized underwater image enhancement. *ISPRS J. Photogram. Remote Sens.* 215, 1–14. doi: 10.1016/j.isprsjprs.2024.06.019

Xiao, H., Chen, X., Luo, L., and Lin, C. (2025). A dual-path feature reuse multi-scale network for remote sensing image super-resolution. *J. Supercomput.* 81, 1–28. doi: 10.1007/s11227-024-06569-w

Xu, S., Ji, Y., Wang, G., Jin, L., and Wang, H. (2023a). "Gfspp-yolo: A light yolo model based on group fast spatial pyramid pooling," in *2023 IEEE 11th International Conference on Information, Communication and Networks (ICICN)*. (New York, USA: IEEE), 733–738. doi: 10.1109/ICICN59530.2023.10393445

Xu, X., Jiang, Y., Chen, W., Huang, Y., Zhang, Y., and Sun, X. (2023b). Damo-yolo: A report on real-time object detection design. Available online at: https://arxiv.org/abs/2211.15444.

Yan, M., Jiang, X., Ren, Y., Li, J., Dang, S., Feng, X., et al. (2023). "Dual adversarial contrastive learning for underwater image enhancement," in *2023 2nd International Conference on Image Processing and Media Computing (ICIPMC)*. (New York, USA: IEEE), 1–8. doi: 10.1109/ICIPMC58929.2023.00008

Zhai, X., Huang, Z., Li, T., Liu, H., and Wang, S. (2023). Yolo-drone: An optimized yolov8 network for tiny uav object detection. *Electronics* 12, 3664. doi: 10.3390/electronics12173664

Zhang, W., Wang, H., Ren, P., and Zhang, W. (2024b). Underwater image color correction via color channel transfer. *IEEE Geosci. Remote Sens. Lett.* 21, 1–5. doi: 10.1109/LGRS.2023.3344630

Zhang, M., Wang, Z., Song, W., Zhao, D., and Zhao, H. (2024a). Efficient small-object detection in underwater images using the enhanced yolov8 network. *Appl. Sci.* 14, 1095. doi: 10.3390/app14031095

Zhao, L., Yun, Q., Yuan, F., Ren, X., Jin, J., and Zhu, X. (2023). Yolov7-chs: An emerging model for underwater object detection. *J. Mar. Sci. Eng.* 11, 1949. doi: 10.3390/jmse11101949

Zhou, P., Bu, Y., Fu, G., Wang, C., Xu, X., and Pan, X. (2024). Towards standardizing automated image analysis with artificial intelligence for biodiversity. *Front. Mar. Sci.* 11, 1949. doi: 10.3389/fmars.2024.1349705