Unmanned Systems, Vol. 14, No. 4 (2026) 887–897

DOI: 10.1142/S230138502650024X

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us

Lightweight Detector for Real-Time Marine Organism Detection in Autonomous Underwater Vehicle

Na Cheng[*,§], Mingrui Li[†,¶], Xuehao Sun[‡,‖], Hongyu Wang[†,**]

[*]*School of Information Science and Engineering, Dalian Polytechnic University, Dalian 116034, P. R. China*

[†]*School of Information and Communication Engineering, Dalian University of Technology, Dalian 116024, P. R. China*

[‡]*Neusoft Group (Dalian) Co., Ltd., Dalian 116033, P. R. China*

Marine object detection plays a crucial role in enabling high-precision, real-time operations of autonomous underwater vehicles (AUVs) and facilitating efficient capture of marine organisms. However, the underwater environment presents challenges such as light absorption, scattering, interference, and the mutual occlusion of organisms, which hinder accurate object detection. Current approaches have made notable progress in addressing these challenges, but they often require significant computing resources, limiting their deployment on AUV platforms. To overcome this limitation, we propose a lightweight detector architecture optimized for edge GPU devices, enabling real-time inference performance. Our method builds upon the single-stage detector network and introduces large and selective kernel convolutions to enhance feature expression. Additionally, we introduce a lightweight task-specific context decoupling head to improve detection accuracy and speed. Experimental results reveal that our approach significantly enhances organism detection performance and speed in complex underwater environments, even in resource-limited conditions. Notably, when compared to advanced Yolov7-tiny object detectors, our method achieves a 1.6% AP improvement on the echinus marine organism, 6.3% AP improvement on the holothurian marine organism, 3.0% AP improvement on the scallop marine organism, and 2.0% AP improvement on the starfish marine organism without introducing additional GPU latency. To validate our approach, we developed a portable, cost-effective, and compact AUV equipped with an NVIDIA Jetson TX2 platform. Successful implementation in the Yellow Sea area demonstrates the effectiveness of our proposed method.

Keywords: Marine object detection; lightweight detector; autonomous underwater vehicles; edge GPU devices.

US

1. Introduction

With the rapid development of computer vision technology, underwater intervention tasks can now be accomplished with the aid of underwater detection equipment, such as autonomous underwater vehicles (AUVs). These technologies are employed in a variety of underwater-related sectors [1–6] such as aquaculture, marine species monitoring, marine fishing, and archeology. Apart from the accurate mapping and localization [7–12], high-precision and real-time underwater object detection are essential prerequisites for AUVs to achieve precise autonomous operations and successfully complete underwater intervention tasks. Nonetheless, several inescapable factors make marine organism detection a daunting challenge. First, the scattering effect of a large number of suspended particles on light alter its propagation direction, causing water to appear turbid; the dynamics of ocean currents, wave action, and robotic movements further exacerbate this turbidity, resulting in underwater images with low contrast and blurred details. Second, the absorption of light by the water and its suspended materials shows significant wavelength dependence across different spectral regions, leading to color distortion

Received 27 June 2024; Revised 10 March 2025; Accepted 11 March 2025; Published 18 June 2025. This paper was recommended for publication in its revised form by editorial board member, Biao Wang.
Email Addresses: [§]chengna@dlpu.edu.cn, [¶]2905450254@mail.dlut.edu.cn, [‖]sunxuehao@neusoft.com, [**]whyu@dlut.edu.cn
[**]Corresponding author.

in the acquired underwater images. Third, marine organisms often cluster together, obscuring each other and increasing the challenges associated with detection. In recent years, deep learning-based marine organism detection has made significant strides in both accuracy and efficiency, with several excellent networks proposed. Li *et al.* [13] proposed an underwater object detection network based on angled multi-scale aggregated feature pyramid to solve the challenges of underwater object detection caused by color distortion and loss of small object details. Song *et al.* [14] proposed a two-stage underwater detector called Enhanced R-CNN, which solved the object detection challenges caused by uneven lighting, low contrast and occlusion in complex underwater environments through uncertainty modeling and difficult sample mining. These two-stage models [15–17] can typically achieve high accuracy but are challenging to implement on the AUVs platforms with limited computing resources. Recently, many one-stage underwater object detection models have been proposed. Guo *et al.* [18] proposed an underwater object detection method based on an improved MSRCP and Yolov3 to address the challenges of detecting aquatic products in conditions of image blur and low contrast. Shen *et al.* [19] used a cross-global interaction strategy to reduce the interference of underwater background on the detected target. However, these marine organism detection methods [13, 18, 20–24] require high-performance computing and a large runtime memory footprint to maintain good detection results, leading to high computing overhead and power consumption on the onboard embedded devices of the AUVs platform. Therefore, the urgent challenge in deploying marine organism detection algorithms for AUVs is to reduce the size of trainable parameters without compromising accuracy significantly. This paper focuses on optimizing the detector architecture to achieve real-time inference performance suitable for AUV-edge GPU devices.

We propose a lightweight detector for AUVs that exhibits better adaptability and performance on edge GPU devices by refining the backbone, neck, and head structures of Yolov7-tiny. Specifically, the backbone network is substituted with a lightweight feature extraction network to facilitate effective feature extraction with reduced computational overhead. The neck structure introduces large and selective kernel attention blocks to enhance feature representation. Moreover, a lightweight task-specific context decoupling header is introduced to improve detection accuracy and speed. Compared to state-of-the-art lightweight marine object detection methods, our approach exhibits superior accuracy and portability. Importantly, even with additional prediction heads, our method maintains its advantages in terms of accuracy, parameter efficiency, GFLOPs cost, and inference time. To validate our method's effectiveness, we designed a portable, low-cost, and small-sized AUV equipped with the NVIDIA Jetson TX2 platform and successfully implemented the proposed method in the Yellow Sea area. Experimental results show that our method can significantly enhance the marine life detection performance of AUVs in resource-limited environments.

This work's contributions can be summed up as follows:

(i) We propose a novel lightweight marine organism detection network that can run in real-time on AUV edge devices for efficient marine organism detection and has the ability to detect small organisms.
(ii) We make the first attempt to introduce a large and selective kernel attention block to the marine organism detection task, which can dynamically adjust its large spatial receptive field to more effectively handle the wide variations of organisms detected under different environments.
(iii) We develop a portable, low-cost AUV equipped with an NVIDIA Jetson TX2 platform, and carry out extensive validation experiments to show the proposed system's competitive accuracy, robustness, and efficiency on resource-restricted platforms in complex underwater environments.

This paper is arranged as follows. Section 2 provides a brief review of related works in marine object detection. The proposed method is presented in Sec. 3. Section 4 describes the experiments conducted on the underwater object dataset. Section 5 concludes.

2. Related Work

The field of underwater object detection has witnessed remarkable advancements, drawing inspiration from general object detection techniques. Deep learning-based underwater object detection algorithms can be broadly classified into two categories: two-stage and one-stage approaches. Researchers have dedicated significant efforts in recent years to enhance the precision and speed of underwater object detection, leading to notable outcomes. Xu *et al.* [16] proposed an object detector based on a scale-aware feature pyramid for detecting objects in underwater environments. Qi *et al.* [17] introduced a two-stage underwater small object detection network with a deformable convolutional pyramid structure, aiming to address challenges such as object deformation, occlusion, and scale changes in the underwater environment. Although two-stage methods often achieve exceptional detection accuracy, they tend to be less efficient and challenging to adapt to AUV platforms with limited computational resources. On the other hand, one-stage underwater object detection methods significantly reduce the computational cost of models and cater to the demand for rapid underwater object detection in comparison to their two-stage counterparts. Several researchers have recently

turned their attention to one-stage underwater object detection algorithms. Hu *et al.* [20] proposed a sea urchin detection network based on the Single Shot MultiBox Detector (SSD) algorithm [25]. Chen *et al.* [26] proposed SWIPENet, a neural network architecture for detecting small underwater targets based on DSSD method [27]. However, despite the advancements in recognition accuracy, these methods often lead to a substantial increase in computational costs, which in turn adversely impacts the speed of inference.

Lightweighting has always been a key objective in optimizing neural network architectures during the development of deep learning models [28]. Many researchers have focused on designing efficient detector architectures for lightweight models, including representative approaches such as Nanodet [29], Yolox-nano [30], Yolox-tiny [30], and Yolox-s [30]. The lightweight models typically consist of three components. First, the feature extraction component, based on convolutional neural network (CNN) backbones, incorporates lightweight designs such as Mobilenet [31], GhostNet [32], and ShuffleNet [33]. Second, the feature fusion component, often implemented through the neck layer, has witnessed extensive experimentation with lightweight neck designs. Various methods have been proposed, such as the dilation encoder and unified matching introduced by YOLOF [34] to replace the feature pyramid network. NanoDet [29] adopts an interpolation-based approach to complete upsampling and downsampling, fully eliminating convolutions in PANet [35]. Additionally, both PPYolovE [36] and Yolov7 [37] leverage re-parameterized convolution [38] to enhance detection performance within the neck layer. Lastly, the detection head component predicts categories and bounding boxes employing feature allocation based on the neck layer. Notably, recent advancements have predominantly focused on improving the regression and classification tasks within the head layer [39–41].

Many researchers work on lightweight underwater object detection to assist AUVs in understanding objects with limited computational resources. Fan *et al.* [42] used neural structure search and knowledge distillation in the Yolov8 model to solve the problem of insufficient computing resources for underwater target detection. Wu *et al.* [43] improved both the accuracy and training efficiency of underwater object detection by compressing the backbone of the ATSS model [44], introducing a plug-and-play receptive field module (F-ASPP) to enhance receptive fields and spatial information, and optimizing the learning rate strategy alongside the classification loss function. Additionally, Ouyang *et al.* [45] designed a lightweight Mobile-bone network for feature extraction by combining the advantages of CNN and ViT, and used the deformable upsampling (DU) method based on semantic alignment to optimize the feature fusion process. This approach effectively improves the model's learning capabilities and generalization while reducing parameters. Consequently, this paper aims to optimize the architecture of underwater object detectors, enabling real-time inference performance on edge GPU devices suitable for AUVs.

3. Proposed Approach

Given that Yolov7-tiny can effectively reduce the number of parameters and computational complexity while maintaining high accuracy, we propose a lightweight marine organism detection network based on Yolov7-tiny to enhance both the accuracy and speed of marine organism detection. Figure 1 consists of backbone, neck, and head. Notably, the backbone network is replaced with the lightweight GhostNet, enabling efficient feature extraction while minimizing computational overhead. In addition, the neck structure incorporates the large and selective kernel attention block (LKSABlock) within the ELAN module, enhancing the model's ability to capture intricate spatial and semantic information. GSConv is specifically designed to reduce model parameters and computational workload while retaining the features of dense convolution to enhance the expressive power and accuracy of the model. This approach maximizes the potential of GSConv to effectively capture and encode task-specific contextual information while optimizing the task-specific decoupled contextual head, thereby providing more comprehensive and accurate feature representation for subsequent detection tasks. The proposed method can achieve efficient and real-time marine organism detection on the Jetson TX2 GPU embedded platform. Next, we will introduce this design in detail.

3.1. *Backbone network*

The backbone architecture of Yolov7-tiny comprises CBS, E-ELAN, and MP modules. While the E-ELAN module exhibits consistent input and output channels, fewer group convolutions, fewer network branches, and reasonable tailoring of element-level operations, it falls short of the criteria for lightweight network models. Han *et al.* [32] proposed the GhostNet network, in which the Ghost module was introduced and used a depthwise convolution method with lower computational cost instead of conventional convolution, thereby reducing network calculation quantity. The Ghost module effectively mitigates the negative impact of redundant feature maps on network time consumption and parameter count, thereby making it very suitable for deploying constrained devices with limited resources while ensuring performance. Therefore, we replaced the original network with the lightweight GhostNet as the backbone architecture.

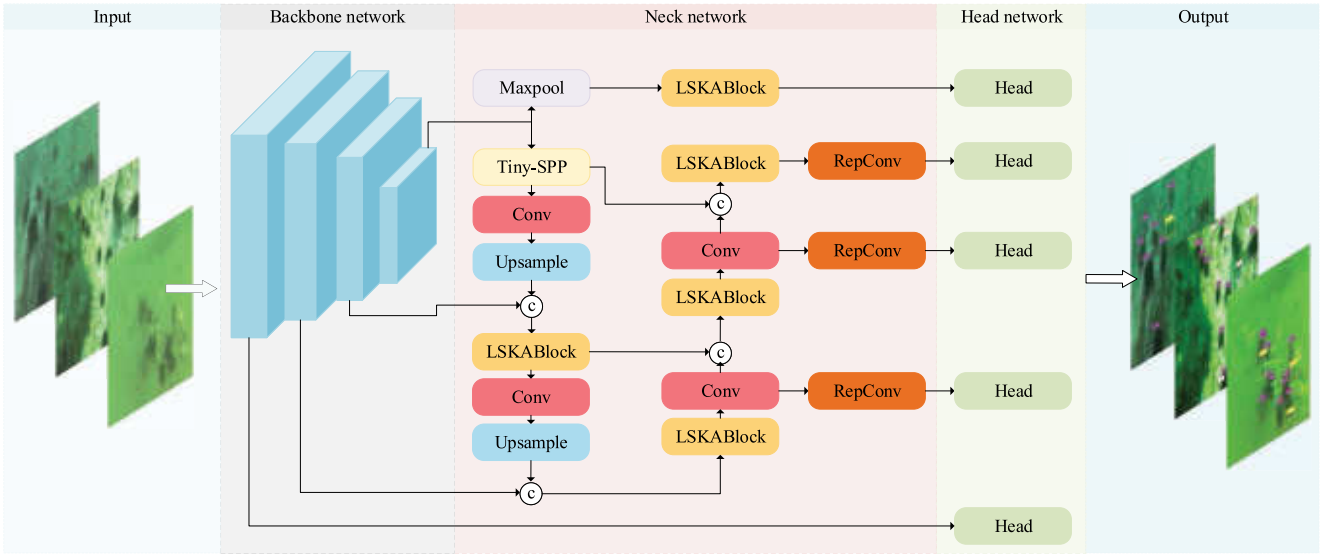

Fig. 1. The architecture of the proposed method.

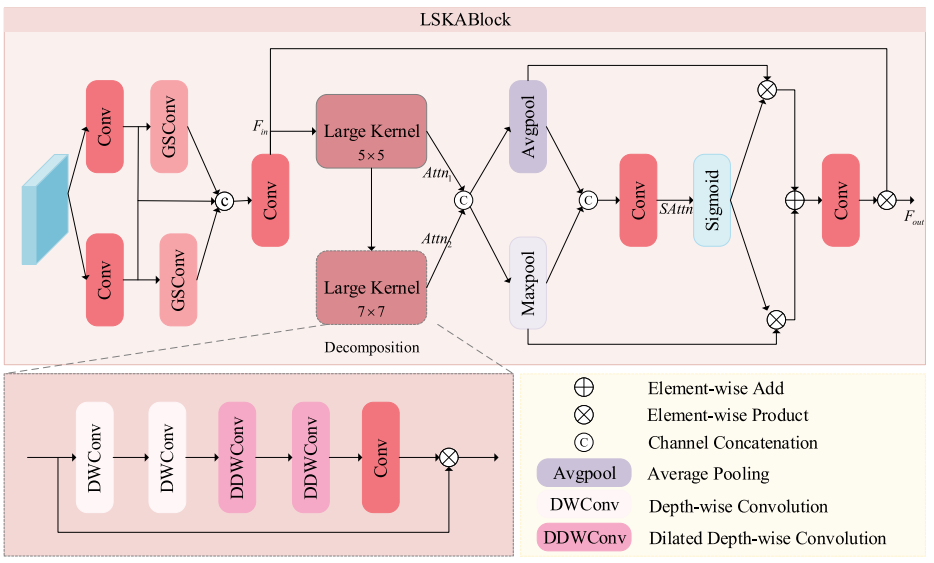

Fig. 2. The architecture of the LSKABlock.

3.2. *Neck network*

In response to the significant computational demands and parameter intricacies inherent in the large-kernel convolution technique, we decompose the large-kernel convolution technique to capture long-term relationships.

Specifically, the large kernel convolution $K \times K$ is deconstructed into a sequence of operations, including two depth-wise convolutions, two dilated depth-wise convolutions utilizing dilation factor d, and a 1×1 convolution. This process can be expressed as

$$\mathrm{Attn}_1(F_{\mathrm{in}}) = \mathrm{Conv}(\mathrm{DWConv}_2(\mathrm{DWConv}_1(F_{\mathrm{in}}))), \quad (1)$$

$$F = \mathrm{DWConv}_2(\mathrm{DWConv}_1(\mathrm{Attn}_1(F_{\mathrm{in}}))), \quad (2)$$

$$\mathrm{Attn}_2(F) = \mathrm{Conv}(\mathrm{DDWConv}_2(\mathrm{DDWConv}_1(F))), \quad (3)$$

$$\mathrm{Attn} = [\mathrm{Attn}_1(F_{\mathrm{in}}); \mathrm{Attn}_2(\mathrm{F})], \quad (4)$$

where DWConv_1 and DWConv_2 represent a $1 \times (2d-1)$ depth-wise convolution and a $(2d-1) \times 1$ depth-wise convolution; $\mathrm{DDWConv}_1$ and $\mathrm{DDWConv}_2$ denote a $1 \times \frac{K}{d}$ dilated depth-wise convolution and a $\frac{K}{d} \times 1$ dilated depth-wise convolution; $F_{\mathrm{in}} \in \mathbb{R}^{C \times H \times W}$ is the input feature. $\mathrm{Attn} \in \mathbb{R}^{C \times H \times W}$ denotes attention map. C, H, and W represent the number of channels, height, and width of the feature map, respectively. [;] is a concatenate operation. The current decomposition approach adeptly captures long-term interactions with minimal computational overhead and parameters, while also achieving adaptability in both spatial and channel dimensions through the facilitation of channel fusion for each spatial feature vector. In order to enhance the network's ability to focus on pertinent spatial contextual regions for object detection, we propose the incorporation of a spatial selection mechanism designed to adaptively aggregate information across large kernels in the spatial dimension. Specifically, our methodology integrates channel-based average pooling and max pooling to facilitate the successful extraction of features with spatial associations, accompanied by the utilization of a convolutional layer with two channels. The resulting aggregated features are subsequently transformed into N spatial attention maps. This process is represented as

$$\mathrm{SAttn} = \mathrm{Conv}([\mathrm{Poolavg}(\mathrm{Attn}); \mathrm{Poolmax}(\mathrm{Attn})]), \quad (5)$$

where Poolavg and Poolmax represent average pooling and max pooling operations. Following this, the features derived from the decomposed large kernel sequence undergo a weighting procedure involving the corresponding spatial selection mask of each decomposed large kernel, obtained through the application of sigmoid activation function. Subsequently, these weighted features are fused with the input features via the convolutional layer, thereby yielding the generation of a feature map. This process is represented as

$$F_{\mathrm{sout}} = \sigma(\mathrm{SAttn}) \otimes \mathrm{Attn}_1 \oplus \sigma(\mathrm{SAttn}) \otimes \mathrm{Attn}_2, \quad (6)$$

$$F_{\mathrm{out}} = \mathrm{Conv}(F_{\mathrm{sout}}) \otimes F_{\mathrm{in}}, \quad (7)$$

where σ is sigmoid activation function; "$\oplus$" is an element-wise add operation; "$\otimes$" is an element-wise product operation.

3.3. *Head network*

In a conventional object detection framework, the classification and localization tasks are commonly executed through two parallel branches featuring identical structures. This approach assumes that the corresponding receptive fields in the feature map for these tasks are identical. However, it is crucial to acknowledge the distinct requirements of these tasks. While classification heavily depends on regions abundant in semantic contextual information, localization places greater emphasis on contour regions containing boundary-aware features. In response to this consideration, we present a lightweight task-specific context decoupling head aimed at further decoupling feature encoding for classification and regression tasks. This strategy can better focus on the most relevant features, thereby enhancing the accuracy of both classification and localization tasks by addressing their specific demands. Generally speaking, low-level features contain detailed information but lack semantic context, while high-level features exhibit the opposite. For classification tasks, our objective is to generate feature encodings that are spatially coarse yet rich in semantic information. High-level features offer the advantage of providing more contextual semantic information. Consequently, a natural idea is to fuse features from deep layers and integrate rich semantic information into the current feature map. It can be expressed as

$$L_c(F_l^{\mathrm{cls}}) = L_{\mathrm{cls}}(P_{\mathrm{cls}}(F_l^{\mathrm{cls}}), C(F_l^{\mathrm{cls}}), c), \quad (8)$$

where P_{cls} is the feature projection function for classification. c is the class label. C is the branch's last layer, which decodes features into classification scores. F_l^{cls} receives two different levels feature maps (F_l and F_{l+1}) for our head to generate semantically rich feature maps for classification. As illustrated in Fig. 3(b), we first downsample F_l by a factor of two before concatenating it with F_{l+1} to produce the final F_l^{cls}. F_l^{cls} is expressed as

$$F_l^{\mathrm{cls}} = [\mathrm{DGConv}(F_l; F_{l+1}], \quad (9)$$

where DGConv is a shared downsampled ghost convolution layer. In this way, we can not only exploit the sparsity of salient features F_l from the l layer, but also benefit from the rich contextual semantic information F_{l+1} from $l+1$ deeper layers. This aids in the efficient inference of object types, particularly for those lacking texture or with severe occlusions.

For localization tasks, our goal is to generate high-resolution feature maps with richer edge information to better regress object boundaries. Low-level features include more detailed information but lack semantic information.

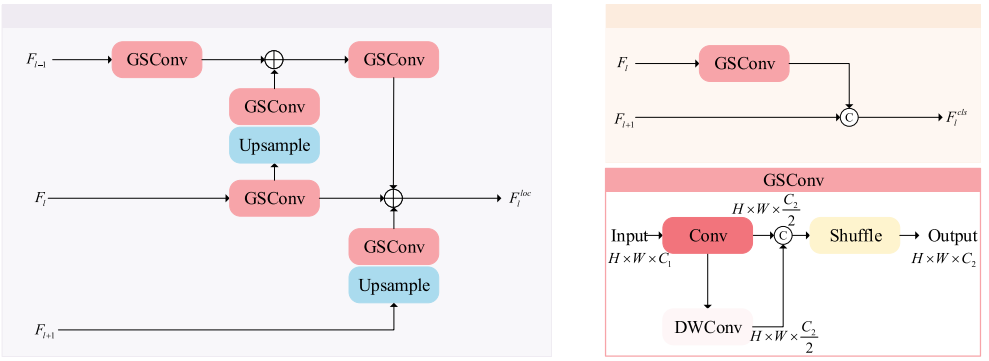

Fig. 3. The architecture of the head. (a) Structure of semantic context feature encoding for localization tasks. (b) Structure of detail-preserving feature encoding for classification tasks.

Therefore, the information from the shallow layer is directed back to the next layer feature map, and the features of adjacent levels are fused. It can be expressed as

$$L_l(F_l^{\text{loc}}) = L_{\text{loc}}(P_{\text{loc}}(F_l^{\text{loc}}), R(F_l^{\text{loc}}), b), \tag{10}$$

where P_{loc} is the feature projection function used for localization, while R is used for the last layer in this branch that decodes features into bounding box locations. b is the bounding box. Our head receives feature maps at three different levels ($l-1$, l, and $l+1$) to generate feature maps (F_{l-1}, F_l, and F_{l+1}) rich in texture details and boundary information for localization. F_{l-1} provides more details and edge features, whereas F_{l+1} provides a more complete perspective of the object. As seen in Fig. 3(a), we fuse F_{l-1} and F_{l+1} using a U-Net structure and GSConv. It can be written as follows:

$$F_l^{\text{loc}} = F_l + \mu F_{l+1} + \text{DConv}(\mu(F_l + F_{l+1})), \tag{11}$$

where μ represents upsampled operation and DConv is another shared downsampled convolutional layer. Detectors with decoupled heads minimize classification and localization losses based on different feature maps. It can be written as follows:

$$L = L_c(F_l^{\text{cls}}) + \alpha L_l(F_l^{\text{loc}}), \tag{12}$$

where α is the balancing coefficient for the loss function. L_c and L_l represent the classification loss function and the localization loss function, respectively.

4. Experiments and Results

4.1. *Implementation details*

The availability of datasets in the field of marine object detection is currently limited. To comprehensively evaluate our proposed method, we utilized the MOD dataset [23], which was collected and constructed in a real ocean environment using an AUV. This dataset encompasses four distinct categories: holothurian, echinus, starfish, and scallop. The MOD dataset plays a pivotal role in providing fundamental data and indispensable support for crucial applications such as marine object monitoring and fishing industries. During the training process, we employed the SGD optimizer with a batch size of 8 and a learning rate of 0.01. To ensure optimal performance, the training procedure spanned a total of 300 epochs, with the learning rate dynamically adjusted via a Cosine Annealing learning rate schedule. In addition to the conventional AP and mAP detection evaluation metrics, we also considered supplementary indicators such as parameters (Params), model size, and FLOPs to holistically assess the algorithm's detection speed.

4.2. *Qualitative and quantitative results*

To further evaluate the performance of our proposed method, we compared it with classic lightweight networks

Table 1. Comparison with the state-of-the-art lightweight object detection methods on the MOD dataset.

Method	Backbone	Input size	Echinus	Holothurian	Scallop	Starfish	mAP_{50}(%)	FLOPs(G)	Params(M)
Yolov4-tiny	CSPDarknet53-tiny	416 × 416	85.60	44.53	21.10	71.63	55.71	6.84	5.88
Yolov5-n	CSPDarknet53	512 × 512	90.01	54.31	31.15	77.70	63.29	4.24	1.77
Yolov5-s	CSPDarknet53	512 × 512	92.50	66.95	34.63	84.18	69.57	15.98	7.03
Yolox-tiny	Darknet53	640 × 640	91.96	72.74	36.02	85.46	71.55	12.90	5.06
Yolox-nano	Darknet53	640 × 640	88.24	52.21	29.54	79.61	62.40	**2.55**	**0.91**
Yolov7-tiny	Tiny-ELAN-Net	640 × 640	92.30	72.00	40.10	84.60	72.25	13.10	6.02
Ours	GhostNet	640 × 640	**93.90**	**78.30**	**43.10**	**86.60**	**75.48**	11.50	8.70

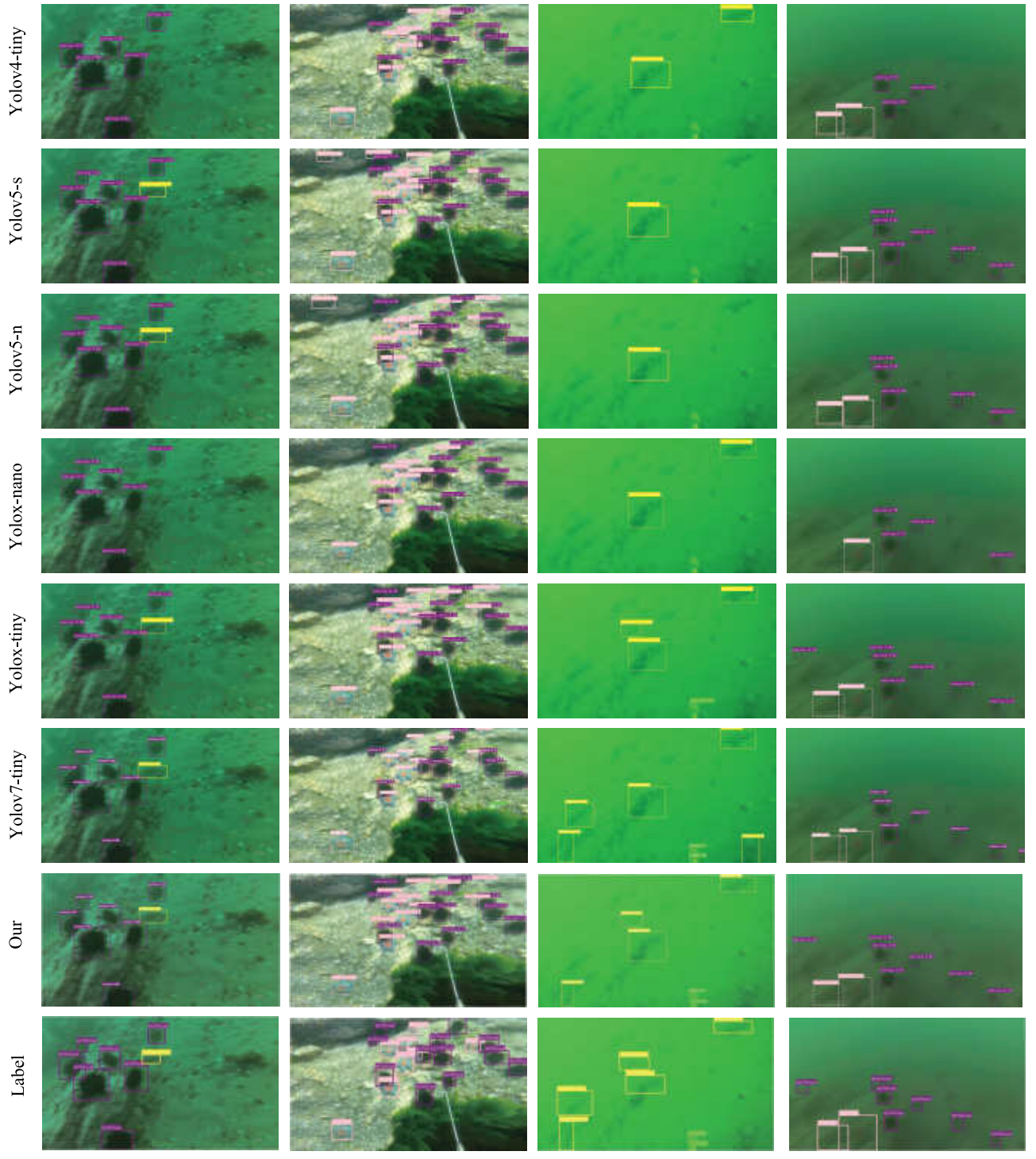

Fig. 4. The architecture of the proposed method.

such as Yolov4-tiny, Yolov5-n, Yolov5-s, Yolox-tiny, Yolox-nano, and Yolov7-tiny. The results are shown in Table 1. When compared to Yolov4-tiny, Yolov5-n, and Yolox-nano, our proposed method shows significant improvements in accuracy rates by 19.77%, 12.19%, and 13.08%, respectively, despite an increase in the number of FLOPs and parameters to varying degrees. On the other hand, when compared to Yolov5-s, Yolox-tiny, and Yolov7-tiny, our method reduces FLOPs to various extents while slightly increasing the number of parameters. In these cases, the accuracy improves by 5.91%, 3.93%, and 3.23%, respectively. It is worth noting that our method outperforms other mainstream lightweight networks in detecting underwater organisms. Figure 4 illustrates the detection effects of several classic lightweight object detection algorithms. Despite the challenges posed by incomplete or occluded targets, our proposed method exhibits remarkable detection robustness. Furthermore, our method demonstrates a favorable influence on the detection performance of small-scale targets.

4.3. *Ablation study*

Effect of backbone network. In order to compare with other lightweight networks, we have selected Mobilenetv3 [46], EfficientVit [47], FasterNet [48], and GhostNet as feature extraction networks, while keeping the detection head unchanged. The experimental findings, presented in Table 2, indicate a significant reduction in the number of parameters and FLOPs when replacing the original YOLOv7-Tiny backbone network with a lightweight alternative. Furthermore, a thorough analysis was conducted focusing on two key indicators: detection accuracy and detection speed. GhostNet demonstrates exceptional performance in the realm of lightweight networks, as it exhibits a compact model size, achieves precise target detection, and ensures rapid detection speed. Therefore, the decision to utilize GhostNet as the backbone network is crucial in achieving optimal results for our method.

Effect of module component. Extensive ablation studies, as detailed in Table 3, are instrumental in confirming the influence of individual components within our method on the overall performance. Positioned as the reference point, the complete model is represented in the seventh row. The impact of targeting each component is shown from the first to the third row, showcasing the individual effects on the detection performance of each ocean object when only the LSKABlock module, only the RepConv module, and only the head module are present, respectively. We systematically remove three specific modules from the full model, as depicted in the fourth to seventh rows. As can be seen from the analysis of the fifth and sixth rows in the table, the head module and the LSKABlock module produced the most significant quantitative improvements.

Table 2. Comparison with other methods of lightweight networks as feature extraction networks.

Backbone network	Publication	mAP_{50}(%)	Params(M)	FLOPs(G)
Original	CVPR'23	72.251	6.02	13.1
GhostNet	CVPR'20	**73.475**	4.31	7.5
Mobilenetv3	ICCV'19	70.275	4.48	**6.7**
EfficientVit	CVPR'23	71.412	5.53	10.2
FasterNet	CVPR'23	72.375	**4.10**	8.5

Table 3. Ablation study of each component in our proposed method on the MOD dataset.

LSKABlock	RepConv	Head	mAP_{50}(%)	Params(M)	FLOPs(G)
✓			73.950	4.45	7.9
	✓		73.525	**3.51**	**6.0**
		✓	**76.525**	23.67	33.3
✓	✓		73.975	3.71	6.50
✓		✓	75.825	19.83	28.9
	✓	✓	75.275	9.84	12.9
✓	✓	✓	75.475	8.70	11.5

Table 3 reveals that the introduction of the LSKABlock module into the neck network structure led to a marginal increase in FLOPs and model parameters, accompanied by a noteworthy improvement of 0.475% in mAP. Additionally, by replacing the traditional convolution operation with RepConv, comparable accuracy was achieved, with FLOPs and model parameters accounting for only 81% and 80% of the original approach, respectively. However, the utilization of the head model resulted in a substantial increase of 19.37% and 25.8% in FLOPs and model parameters, respectively, although it enhanced accuracy by 3.05%. Through the substitution of the neck network structure with LSKABlock and RepConv modules, we obtained an enhanced model with slightly reduced FLOPs and model parameters, while achieving a slight increase in accuracy. Considering the cumulative enhancements in both the neck and head components, our model attained an accuracy of 75.475%. It is worth noting that although there was an increase in FLOPs and parameter count, the magnitude of the increase was not substantial.

4.4. *The necessity of model lightweight*

The continuous advancement of CNN models has yielded remarkable improvements in model accuracy, resulting in an increase in FLOPs and parameters. However, the growing prevalence of heavy models imposes significant computational demands. In the domain of underwater robotics, the adoption of edge computing devices as the computational core is crucial due to their cost-effectiveness and portability. Table 4 presents a comprehensive inventory of hardware combinations, encompassing GPU, RAM, Giga floating-point operations per second (GFLOPS), thermal design power, and

Table 4. Comparison with the state-of-the-art lightweight object detection methods on the MOD dataset.

Computer platform	GPU	Memory	GFLOPS	Thermal design power	Manufacturing process
GeForce RTX3090	NVIDIA Ampere 10496 CUDA	24GB 384bit GDDR6X	285TFLOPs	350 W	8 nm
Jetson TX2	NVIDIA Pascal^{TM} GPU 256 CUDA	8GB 128bit LPDDR4	1.3TFLOPs	7.5 W	16 nm

manufacturing method, highlighting the variance in computing capabilities between edge computing devices and professional computing cards. Furthermore, a detailed account of the hardware environment within our deep learning workstation reveals an Intel Core^{TM} i7-11700 CPU clocked at 2.50 GHz, an RTX3090 GPU, and 32 GB of RAM.

In order to demonstrate the effectiveness of our proposed method, we designed a portable, low-cost, and compact AUV and successfully implemented the method in the Yellow Sea area in Dalian, China. Our self-designed AUV, as shown in the Fig. 5, consists of the following components: (1) LED light source; (2) Thrusters; (3) An NVIDIA Jetson TX2; (4) Lithium ion rechargeable battery; (5) Camera. The NVIDIA Jetson TX2 is a microcomputer with a GPU processing module, enabling robots equipped with the NVIDIA Jetson TX2 to utilize our proposed lightweight underwater object detection method for real-time detection processing of real scene data. The visual images collected by the AUV and the detection of organisms in the visual images are played in real time through the AUV console platform. The coordinated operation of thrusters facilitates the robot's ascent, descent, left and right movement, and steering control, allowing for various postures such as point positioning, straight lines, arcs, and other motion controls with stronger stability and balance capabilities. Due to the low underwater visibility, we

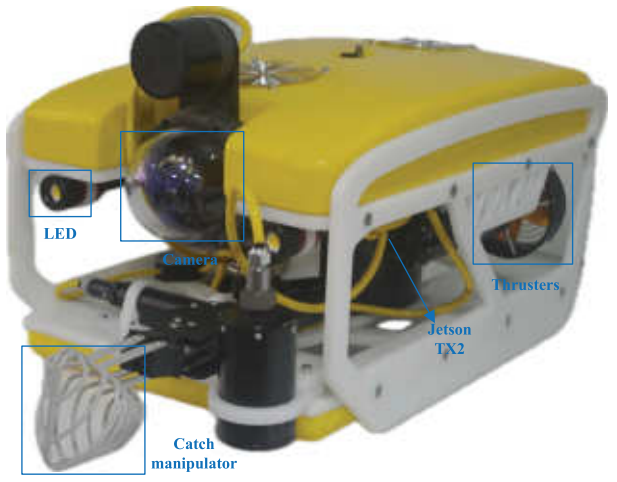

Fig. 5. A portable, low-cost AUV equipped with an NVIDIA Jetson TX2 platform.

Table 5. Comparison with the state-of-the-art object detection methods on both real-world scenes and the MOD test dataset.

Method	Input size	FPS^{Avg}_{TX2}	$Time^{Avg}_{TX2}$(s)	FPS^{Avg}_{3090}	$Time^{Avg}_{3090}$(s)
Yolov4-tiny	416 × 416	**22**	**0.0443**	**241**	**0.0042**
Yolov5-s	512 × 512	12	0.0835	114	0.0087
Yolov5-n	512 × 512	14	0.0720	125	0.0079
Yolox-tiny	640 × 640	10	0.0921	114	0.0087
Yolox-nano	640 × 640	10	0.0919	107	0.0093
Yolov7-tiny	640 × 640	12	0.0786	140	0.0071
Our	640 × 640	11	0.0913	123	0.0082

have chosen to use a 4K high-definition underwater camera and equipped it with adjustable LED light as a supplementary measure to improve the quality of image acquisition. The software environment of AUV is Python 3.6, CUDAToolkit 10.2, cuDNN7.6, and TensorRT 7.1.3. TensorRT is a runtime engine for executing models with higher throughput and lower latency. In real ocean scene tests, the model file is first converted to an ONNX intermediate format file, following which the compiler within the TensorRT suite is used to convert the intermediate file into an engine file. Subsequently, the C++/Python API interfaces within the intermediate file can be invoked to achieve end-to-end model applications. Edge computing devices offer only 0.5% of the computing power compared to workstations, while consuming a mere 2% of the power. As shown in Table 5, although the inference speed of our algorithm is not as fast as that of the Yolov4-tiny method, it is comparable to the other five models. Even on the weaker Jetson TX2, the inference speed of our algorithm can reach 11 FPS. In contrast, our method can achieve a perfect balance between accuracy and speed.

5. Discussion

We have presented a novel lightweight marine organism detection network that can run in real-time on edge devices for efficient detection and has the ability to detect small objects. Since the approach achieves a convincing balance between accuracy and speed by leveraging the lightweight GhostNet backbone integrated with the LKSAmodule and task-specific context decoupled headers empowered by GSConv. Experimental verification on the Jetson TX2 GPU platform confirmed the effectiveness of the proposed method, paving the way for its practical deployment in marine organism detection applications. In future work, we

will further study how to combine model design and compression algorithm to develop a friendly marine organism detection framework that is more suitable for deployment on AUV edge devices.

References

[1] H. B. Amundsen, W. Caharija and K. Y. Pettersen, Autonomous ROV inspections of aquaculture net pens using DVL, *IEEE J. Ocean. Eng.* **47**(1) (2021) 1–19.

[2] Y. Wu, Y. Duan, Y. Wei, D. An and J. Liu, Application of intelligent and unmanned equipment in aquaculture: A review, *Comput. Electron. Agric.* **199** (2022) 107201.

[3] M. A. Mercado, Y. Sekimori and T. Maki, Low-cost AUV visual system for marine ecosystem monitoring and analysis, in *OCEANS 2023-MTS/IEEE US Gulf Coast* (IEEE, 2023), pp. 1–6.

[4] M. Salhaoui, J. C. Molina-Molina, A. Guerrero-Gonz'alez, M. Arioua and F. J. Ortiz, Autonomous underwater monitoring system for detecting life on the seabed by means of computer vision cloud services, *Remote Sens.* **12**(12) (2020) 1981.

[5] W. Akram, T. Hassan, H. Toubar, M. Ahmed, N. Miškovic, L. Seneviratne and I. Hussain, Aquaculture defects recognition via multi-scale semantic segmentation, *Expert Syst. Appl.* **237** (2024) 121197.

[6] Z. Zhang, Y. Liu, X. Zhu, F. Li and B. Song, DSE-FCOS: Dilated and SE block-reinforced FCOS for detection of marine benthos, *Vis. Comput.* **40**(4) (2024) 2679–2693.

[7] T. Deng, H. Xie, J. Wang and W. Chen, Long-term visual simultaneous localization and mapping: Using a bayesian persistence filter-based global map prediction, *IEEE Robot. Autom. Mag.* **30**(1) (2023) 36–49.

[8] T. Deng, G. Shen, T. Qin, J. Wang, W. Zhao, J. Wang, D. Wang and W. Chen, Plgslam: Progressive neural scene represenation with local to global bundle adjustment, in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 2024, pp. 19657–19666.

[9] Z. Wu, W. Wang, J. Zhang, Q. Lyu, H. Zhang and D. Wang, Global localization in repetitive and ambiguous environments, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Yokohama, Japan, 2023), pp. 12374–12380.

[10] Z. Wu, Y. Yue, M. Wen, J. Zhang, G. Peng and D. Wang, MSTSL: Multisensor based two-step localization in geometrically symmetric environments, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Xi'an, China, 2021), pp. 5245–5251.

[11] Z. Wu, W. Wang, J. Zhang, Y. Wang, G. Peng and D. Wang, MGLT: Magnetic-lead global localization and tracking in degenerated repetitive environments, *IEEE/ASME Trans. Mechatron.* **29** (2024) 4687–4698.

[12] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang and D. Wang, Collaborative semantic understanding and mapping framework for autonomous systems, *IEEE/ASME Trans. Mechatron.* **26**(2) (2020) 978–989.

[13] X. Li, H. Yu and H. Chen, Multi-scale aggregation feature pyramid with cornerness for underwater object detection, *Vis. Comput.* **40**(2) (2024) 1299–1310.

[14] P. Song, P. Li, L. Dai, T. Wang and Z. Chen, Boosting r-cnn: Reweighting r-cnn samples by RPN's error for underwater object detection, *Neurocomputing* **530** (2023) 150–164.

[15] W.-H. Lin, J.-X. Zhong, S. Liu, T. Li and G. Li, Roimix: Proposal-fusion among multiple images for underwater object detection, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Barcelona, Spain, 2020), pp. 2588–2592.

[16] F. Xu, H. Wang, J. Peng and X. Fu, Scale-aware feature pyramid architecture for marine object detection, *Neural Comput. Appl.* **33** (2021) 3637–3653.

[17] S. Qi, J. Du, M. Wu, H. Yi, L. Tang, T. Qian and X. Wang, Underwater small target detection based on deformable convolutional pyramid, in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, Singapore, 2022), pp. 2784–2788.

[18] T. Guo, Y. Wei, H. Shao and B. Ma, Research on underwater target detection method based on improved msrcp and yolov3, in *IEEE International Conference on Mechatronics and Automation (ICMA)* (IEEE, Tianjin, China, 2021), pp. 1158–1163.

[19] X. Shen, H. Wang, Y. Li, T. Gao and X. Fu, Criss-cross global interaction-based selective attention in yolo for underwater object detection, *Multimed. Tools Appl.* **83**(7) (2024) 20003–20032.

[20] K. Hu, F. Lu, M. Lu, Z. Deng and Y. Liu, A marine object detection algorithm based on SSD and feature enhancement, *Complexity* **2020**(1) (2020) 5476142.

[21] L. Chen, Z. Liu, L. Tong, Z. Jiang, S. Wang, J. Dong and H. Zhou, Underwater object detection using invert multi-class adaboost with deep learning, in *International Joint Conference on Neural Networks (IJCNN)* (IEEE, Glasgow, England, 2020), pp. 1–8.

[22] S. Cai, G. Li and Y. Shan, Underwater object detection using collaborative weakly supervision, *Comput. Electr. Eng.* **102** (2022) 108159.

[23] N. Cheng, H. Xie, X. Zhu and H. Wang, Joint image enhancement learning for marine object detection in natural scene, *Eng. Appl. Artif. Intell.* **120** (2023) 105905.

[24] X. Shen, H. Wang, T. Cui, Z. Guo and X. Fu, Multiple information perception-based attention in yolo for underwater object detection, *The Vis. Comput.* **40**(3) (2024) 1415–1438.

[25] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, SSD: Single shot multibox detector, in *European Conference on Computer Vision* (Springer, Amsterdam, Netherlands, 2016), pp. 21–37.

[26] L. Chen, F. Zhou, S. Wang, J. Dong, N. Li, H. Ma, X. Wang and H. Zhou, Swipenet: Object detection in noisy underwater scenes, *Pattern Recognit.* **132** (2022) 108926.

[27] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi and A. C. Berg, DSSD: Deconvolutional single shot detector, preprint (2017), arXiv:1701.06659.

[28] G. Peng, Y. Huang, H. Li, Z. Wu and D. Wang, LSDNet: A lightweight self-attentional distillation network for visual place recognition, in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Kyoto, Japan, 2022), pp. 6608–6613.

[29] R. Lyu, Nanodet-plus: Super fast and high accuracy lightweight anchor-free object detection model, 2021, https://github.com/RangiLyu/nanodet.

[30] Z. Ge, Yolox: Exceeding yolo series in 2021, preprint (2021), arXiv:2107.08430.

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

[32] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, Ghostnet: More features from cheap operations, in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 1580–1589.

[33] X. Zhang, X. Zhou, M. Lin and J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in *IEEE Conference on Computer Vision and Pattern Recognition*, Utah, USA, 2018, pp. 6848–6856.

[34] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng and J. Sun, You only look one-level feature, in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 13039–13048.

[35] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, Path aggregation network for instance segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition*, Utah, USA, 2018, pp. 8759–8768.

[36] S. Xu, X. Wang, W. Lv, Q. Chang, C. Cui, K. Deng, G. Wang, Q. Dang, S. Wei, Y. Du *et al.*, Pp-yoloe: An evolved version of yolo, preprint (2022), arXiv:2203.16250.

[37] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in

IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, Canada, 2023, pp. 7464–7475.

[38] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding and J. Sun, RepVGG: Making VGG-style convnets great again, in *IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2021), pp. 13733–13742.

[39] Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li and Y. Fu, Rethinking classification and localization for object detection, in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 10186–10195.

[40] G. Song, Y. Liu and X. Wang, Revisiting the sibling head in object detector, in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 11563–11572.

[41] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang and L. Shao, D2det: Towards high quality object detection and instance segmentation, in *IEEE Conference on Computer Vision and Pattern Recognition*, Seattle, USA, 2020, pp. 11485–11494.

[42] Y. Fan, L. Zhang and P. Li, A lightweight model of underwater object detection based on yolov8n for an edge computing platform, *J. Mar. Sci. Eng.* **12**(5) (2024) 697.

[43] J. Wu, J. Chen, Q. Lu, J. Li, N. Qin, K. Chen and X. Liu, U-atss: A lightweight and accurate one-stage underwater object detection network, *Signal Process. Image Commun.* **126** (2024) 117137.

[44] S. Zhang, C. Chi, Y. Yao, Z. Lei and S. Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (IEEE, 2020), pp. 9759–9768.

[45] W. Ouyang, Y. Wei and G. Liu, A lightweight object detector with deformable upsampling for marine organism detection, *IEEE Trans. Instrum. Meas.* **73** (2024) 1–9.

[46] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, Searching for mobilenetv3, in *IEEE International Conference on Computer Vision*, Los Angeles, USA, 2019, pp. 1314–1324.

[47] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu and Y. Yuan, Efficientvit: Memory efficient vision transformer with cascaded group attention, in *IEEE Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, 2023, pp. 14420–14430.

[48] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee and S.-H. G. Chan, Run, don't walk: Chasing higher flops for faster neural networks, in *IEEE Conference on Computer Vision and Pattern Recognition*, Vancouver, Canada, 2023, pp. 12021–12031.