Detection of Small Underwater Organisms Based on Improved YOLOv8

Liheng Miao, Ying Tian

Abstract—**The underwater environment is complex and diverse, making it challenging to locate aquatic organisms accurately. The precise identification of underwater animals is crucial for ecological research and fisheries management. Addressing the issue of inaccurate localization of small underwater targets, this study introduces a novel model, YOLOv8-2PCC, based on the YOLOv8 algorithm with improvements. First, to improve the efficiency of the YOLOv8 network, the C2F module in the original YOLOv8 network model was replaced with convolution to reduce the computational load of the model. Secondly, the up-sampling operator CARAFE is employed, which excels in capturing features at various scales. Finally, a small target detection layer has been incorporated to extract additional shallow features, effectively enhancing the model's ability to detect small targets. Utilizing the URPC dataset for training and testing, the results indicate that our proposed algorithm achieves a mean Average Precision (mAP) of 85.9%. Compared to YOLOv8n, there is a 4.4% improvement, effectively enhancing the accuracy of underwater organism detection in complex underwater environments.**

Index Terms—**YOLOv8, C2F module, CARAFE, small target detection layer**

I. Introduction

Underwater biological target detection holds significant value in current research and application domains. Firstly, underwater organisms are vital components of marine ecosystems. By detecting and understanding the distribution, quantity, and behavior of underwater organisms, we can deepen our understanding of Marine ecosystems, providing a scientific basis for preserving marine biodiversity and ecological balance. Secondly, underwater organism target detection is crucial for fisheries management. Accurate identification of underwater targets enables effective monitoring of fishery resources, formulation of scientific fisheries management strategies, and prevention of overfishing, ensuring the sustainable development of fisheries. Additionally, underwater organisms' target detection finds extensive applications in marine environmental monitoring, ecological research, marine scientific exploration, and other fields, offering robust technological support for humanity's better understanding, protect, and utilize marine resources. Therefore, in-depth research on underwater organisms' target detection technology holds paramount significance in advancing marine science and sustainable utilization of the ocean.[1-3].

Early target detection methods primarily relied on manually designed features and machine learning-based classifiers. These methods include Haar features with cascade classifiers HOG [4] features with SVM [5] classifiers, and others. While these methods performed well in certain scenarios, their effectiveness was limited in complex backgrounds and situations involving multiple target classes. With the development of deep learning, especially the emergence of Convolutional Neural Networks (CNNs), significant progress has been made in target detection. The R-CNN [6] series methods (including Fast R-CNN, Faster R-CNN) introduced mechanisms such as Region Proposal Networks (RPN) [7] and Region of Interest (RoI) Pooling, greatly improving detection speed. Single-stage detectors such as YOLO and SSD [8-10] further simplified the target detection process by simultaneously handling target localization and classification within a single network, achieving real-time performance. In recent years, end-to-end detectors like RetinaNet and YOLOv4/v5 have become research hotspots. RetinaNet [11] addressed class imbalance issues by introducing Focal Loss, while YOLOv4/v5 achieved significant improvements in speed and accuracy through enhanced network structures and optimized training strategies.YOLOv8 typically exhibits improvements in target detection performance. Each generation's enhancements are often accompanied by more accurate detections and lower false positive rates, contributing to a more reliable application across diverse real-world scenarios. YOLOv8 is committed to maintaining real-time performance while enhancing detection accuracy. By refining network structures, optimizing algorithms, and leveraging more efficient hardware, YOLOv8 may achieve increased processing speed and operational efficiency to a certain extent. Each iteration of YOLO introduces new technologies and features. YOLOv8 may incorporate novel concepts, layers, or training strategies to enhance the algorithm's overall performance and adaptability. Therefore, YOLOv8 was chosen as the fundamental model for this study [12-13].

To better address the demands of detection in complex underwater environments, this paper proposes an algorithm based on the improved YOLOv8, named YOLOv8-2PCC. incorporating the concept of partial convolution into the network. To enhance the detection capability for small targets, a small target detection layer detection is added. The lightweight upsampling operator CARAFE is introduced to aggregate contextual information within a larger receptive field, improving both the detection speed and accuracy.

Manuscript received February 5, 2024; revised June 17, 2024.

Liheng Miao is a postgraduate student majoring in software engineering at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (e-mail: 1113596430@qq.com).

Ying Tian is a professor at the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China. (corresponding author to provide phone: +8613898015263; e-mail: astianying@126.com)

II. PRINCIPLES OF YOLOV8

Among various object detection algorithms, the YOLO series stands out for its outstanding balance between speed and accuracy. It accurately and swiftly identifies targets, making it suitable for deployment on various mobile devices. YOLO has been widely applied in various fields, including object detection, tracking, and segmentation. YOLOv8 is the most advanced object detection algorithm, offering exceptional performance, and is particularly well-suited for underwater organism detection. Its network structure is illustrated in Fig. 1.

To meet various requirements, YOLOv8 is divided into different versions based on the depth and width of the network, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. These versions demonstrate superior performance on the COCO dataset when compared to other YOLO versions. In this study, considering the issue of model size, YOLOv8n [14] was chosen as the research subject.

The YOLOv8 model consists of four main parts: Input, Backbone, Neck, and Head. For the input, the Mosaic data augmentation method is employed, and certain hyperparameters are modified for different-sized models. Notably, larger models enable MixUp and CopyPaste data augmentation to enrich the dataset, enhancing the model's generalization and robustness. The Backbone is responsible for extracting information from images and providing it to Neck and Head. It comprises multiple Conv, C2F modules, and the SPPF module at the end. The Conv module consists of a single Conv2d, BatchNorm2d, and an activation function to extract and organize features. YOLOv8 incorporates the C2F structure, inspired by the C3 module's residual structure and YOLOv7's [15] ELAN concept, ensuring lightweight while obtaining richer gradient flow information. The SPPF, which is the spatial pyramid pooling, is capable of fusing features from different scales. The Neck section primarily facilitates feature fusion by utilizing features extracted by the backbone network. It adopts an FPN [16] + PAN [17] structure, enhancing semantic expression and localization capability across multiple scales. The Head section outputs information about the categories and positions of detected targets, using the processed features from the previous sections. It employs a decoupled head structure, separating classification and detection heads to address the different focus points of classification and localization.

Moreover, it employs anchor-free object detection, which enhances detection speed. In terms of loss calculation, the dynamic allocation of positive and negative samples is adopted. It utilizes VFL Loss for classification and DFL Loss + CIOU Loss for regression. In summary, YOLOv8 incorporates advanced components and strategies aimed at enhancing detection performance, with a primary focus on improving speed and accuracy.

In different terms, the Conv module in YOLOv8 is a composite module composed of Conv2d (2D convolution), BN (batch normalization), and SiLU (Sigmoid-Linear Unit). The convolutional layer performs convolution operations on input data by applying a set of learnable filters (also known as convolution kernels or matrices) to extract feature information. These filters have different capabilities for extracting features, effectively capturing features such as edges and shapes in the input data. To enhance the network's expressive power, the output undergoes activation through the non-linear activation function SiLU. However, the Conv module has a large number of parameters, requiring substantial computational resources. Additionally, due to the local nature of convolution operations, the Conv module still has limitations in understanding global context information, leading to insufficient comprehension of global information.

The C2F module integrates design concepts from the C3 module and ELAN (Efficient Lightweight Attention Network), ensuring model lightweightness while obtaining rich gradient flow information. It comprises components such as Conv, Split, and BottleNeck.

The SPPF (Serial Parallel Pooling Fusion) is a method proposed based on SPP (Spatial Pyramid Pooling) to enlarge the receptive field. It achieves this by serially using multiple 5×5 max-pooling layers, thereby reducing the number of parameters and significantly lowering computational load. The SPPF module takes the feature map as input, processes it through the ConvBNSiLU module, and performs max-pooling downsampling operations. Finally, the

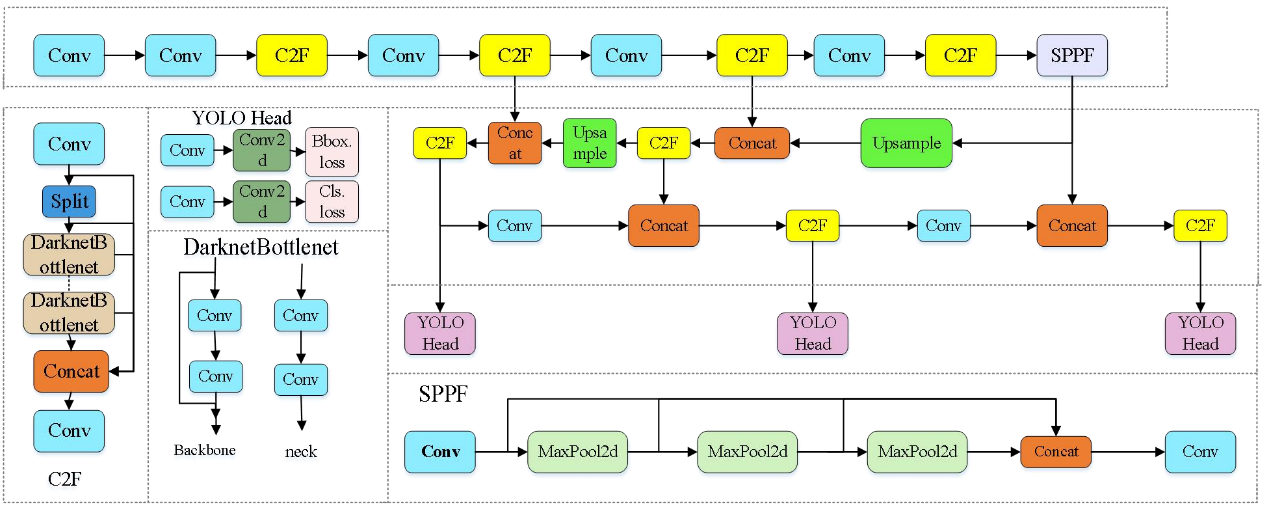


Fig. 1. YOLOv8 model

downsampling results from different layers are concatenated to form the output feature map. By utilizing SPPF, an effective enlargement of the receptive field is achieved, enabling the extraction of global contextual information with fewer parameters and reduced computational load.

III. IMPROVED MODEL

This section introduces improvements made upon YOLOv8 to propose a new network model suitable for underwater organism detection. Firstly, the introduction of the PConv convolutional module is employed to reduce model complexity. Secondly, the upsampling operator CARAFE is introduced to enhance the network's regression accuracy and convergence speed. Lastly, the addition of a small target detection layer is implemented to improve the detection performance for small targets. The network architecture of the proposed model is illustrated in Fig. 2.

A. The C2F module incorporates Partial Convolution (PConv)

The YOLOv8 backbone network primarily utilizes conventional convolutions and the C2F module, enabling high-quality feature extraction from images. Recognizing that the detected images encompass complex scenarios, a simpler convolutional approach is chosen to replace certain conventional convolutions, simplifying the model. NVIDIA introduced a novel convolutional method called PConv, aimed at efficiently extracting features by reducing computations and memory access. PConv [18] selectively applies conventional convolutions to specific input channels, allowing more flexible convolutional calculations when dealing with images containing missing or irregular regions. The characteristics of partial convolutions may contribute to a better capture and integration of contextual information. The following is the formula for PConv.As defined in (1).

$$x' = \begin{cases} W^T(X \odot M)\dfrac{\text{sum}(I)}{\text{sum}(M)} + b, & \text{if } \text{sum}(M) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

X is the input feature map, W is the convolution kernel, M is the input mask (0-1 distribution), b is the bias for convolution operation, and X and M represent pixels within the current operation area. The notation sum(1) refers to a matrix of the same size as the convolution kernel (e.g., 3×3), with all elements being 1. In the first layer of PConv, M contains 1 for undamaged areas and 0 for damaged areas. To enhance the model's capability to handle images with complex scenarios, PConv is introduced into the C2F module, replacing the two original convolutional modules in C2F with PConv. The newly introduced PC2F is capable of replacing the C2F component of the original model. The structure is illustrated in Fig. 3.

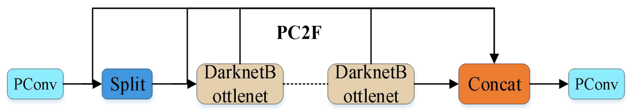


Fig. 3. The structure of PC2F

B. Integrate a small object detection layer

The deep-layer feature maps are suitable for coarse-grained image classification, while the shallow-layer feature maps are more focused on providing detailed positional information [19]. In this study, the detection layers of YOLOv8n have been expanded from the original P3, P4, and P5 layers to include P2, P3, P4, and P5 layers. P2, P3, P4, and P5 are obtained by extracting features at different stages of the Convolutional Neural Network (CNN). Generally, P2 is located in the earlier stages of the network with higher resolution, while P5 is positioned in deeper stages with lower resolution and a larger receptive field. Due to its higher resolution, P2 can more effectively capture details in the images. For small objects, the higher resolution makes it easier to differentiate between the target and the background, thus improving detection accuracy. Conversely, P5 and similar layers have a larger receptive field, which is suitable for detecting larger objects. A larger receptive field can cover a more extensive area, capturing more contextual information.

The term "parameter count" refers to the number of learnable parameters in a target detection model, which primarily consists of the model's weights and biases. These parameters are adjusted during the training process based on the training data, allowing the model to adapt its behavior to specific tasks. In target detection tasks, the model must learn

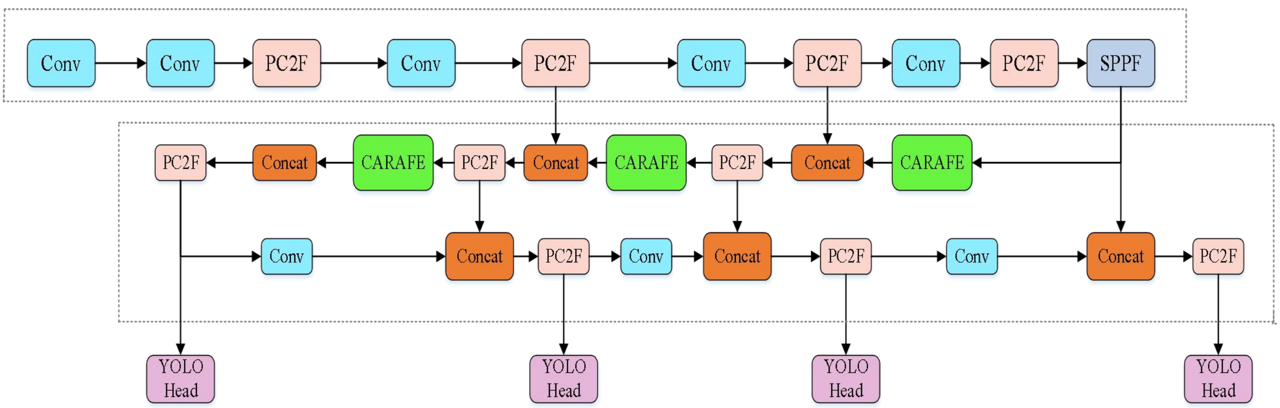


Fig. 2. Improved YOLOv8

to extract information about the location, shape, and class of objects from input images. This adaptation process enables the model to effectively identify and classify objects within the images. The quantity of parameters directly influences the complexity and capacity of the model. Having more parameters usually implies a more complex model with the ability to learn more intricate patterns and representations. However, an excessive number of parameters may lead to overfitting, where the model becomes too tailored to the training data and performs poorly on unseen data. In target detection, finding an appropriate parameter count involves striking a balance in the model design process. On one hand, the model needs sufficient capacity to learn complex representations for the task at hand. On the other hand, an excessive number of parameters may lead to overfitting, diminishing the model's generalization ability. Therefore, managing the parameter count is a crucial consideration in the design and training of deep learning models.

C. *Introducing the upsampling operator CARAFE*

Upsampling is a common operation in image processing and computer vision, primarily used to increase the size or resolution of images or feature maps. In deep learning and neural networks, upsampling is employed to enlarge low-resolution images or feature maps to higher resolutions, thereby capturing more detailed information. This operation is crucial for preserving details and spatial information in images, playing a significant role in tasks such as object detection and image segmentation. By using upsampling, models can more accurately restore downsampled features, thus enhancing overall processing accuracy and effectiveness.

Content-Aware ReAssembly of FEatures (CARAFE), is a universal, lightweight, and highly effective operator to fulfill this goal. CARAFE has several appealing properties, including a Large field of view. Unlike previous works that only exploit subpixel neighborhoods, CARAFE can aggregate contextual information within a large receptive field. Content-aware handling. Instead of using a fixed kernel for all samples, CARAFE [20] enables instance-specific content-aware handling, which generates adaptive kernels on the fly. Lightweight and fast to compute, CARAFE introduces little computational overhead and can be readily integrated into modern network architectures. We conducted comprehensive evaluations on standard benchmarks in object detection. The introduction of CARAFE brings new possibilities to upsampling operations in image processing and computer vision tasks, particularly suitable for scenarios with high semantic information requirements, providing models with a more comprehensive understanding of context. Therefore, adding the CARAFE upsampling operator in this paper enables the extraction of more abundant feature information.

IV. Experiment and analysis

A. *Datasets*

This study employs the underwater target detection dataset URPC. The dataset is divided into three parts: 1334 test images, 667 validation images, and 4469 training images. The images encompass four object categories: holothurian, echinus, scallop, and starfish. Some examples of this dataset are illustrated in Fig. 4.

Fig. 4. URPC dataset example

B. *Evaluation Indicators*

In this study, precision, recall and mean average precision (mAP) were employed as evaluation metrics to assess the model's performance.

Precision: Precision refers to the ratio of true positives to the sum of true positives and false positives, where TP is true positives (the number of correctly predicted positive instances) and FP is false positives (the number of instances incorrectly predicted as positive). As defined in (2).

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall: Recall is the ratio of true positives to the sum of true positives and false negatives, where FN is false negatives (the number of instances incorrectly predicted as negative). As defined in (3).

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

mAP (mean Average Precision) is a metric used for the comprehensive evaluation of model performance in object detection tasks. It represents the average precision (AP) across all categories. In object detection, each category has an associated AP, indicating the average precision for that category at different confidence thresholds. mAP is the average of AP values for all categories and serves as a holistic measure to assess the model's performance across different classes. The calculation involves summing the AP values for each category and then dividing by the total number of categories. This provides a comprehensive evaluation of the model's overall performance in the entire detection task, while individual class AP values offer insights into the model's performance on specific classes. As defined in (4).

$$\mathrm{mAP} = \frac{\mathrm{AP}_1 + \mathrm{AP}_2 + \ldots + \mathrm{AP}_n}{n} \tag{4}$$

In the PR curve, the horizontal axis represents recall, and the vertical axis represents precision. Each point on the curve corresponds to the precision and recall of the model at a specific probability threshold. By plotting these points at different thresholds, the performance of the model at various operating points can be observed. Therefore, to comprehensively compare performance, the PR curves before and after the modifications were compared, as shown in Fig. 5 and Fig. 6.

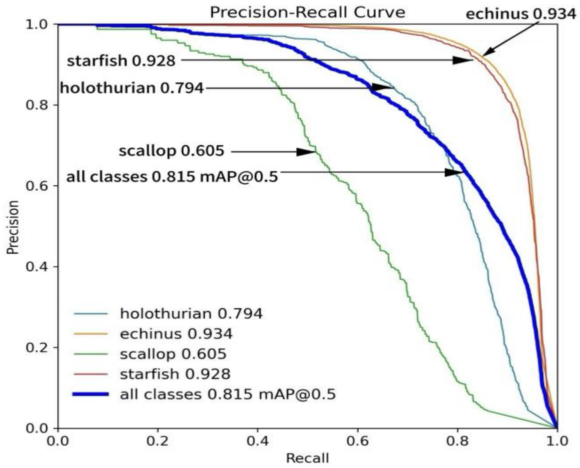


Fig. 5. YOLOv8 P-R curve

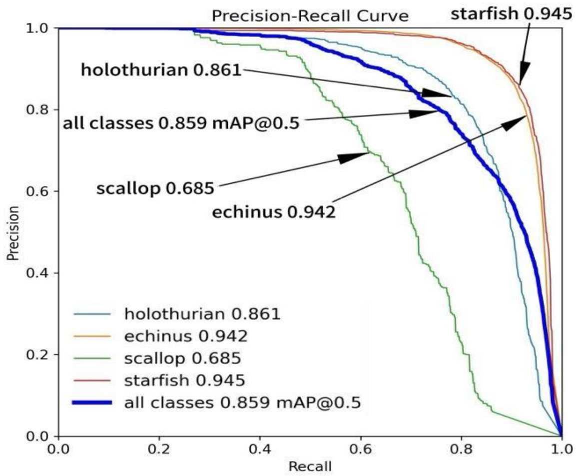


Fig. 6. YOLOv8-2PCC P-R curve

To more clearly show the data comparison, Table I presents the comparison between the YOLOv8 model and the YOLOv8-2PCC model, indicating an overall improvement of 4.4%. Holothurian and scallop show significant increases, with growth rates of 6.7% and 8%, respectively. Echinus and starfish also experience certain improvements, with increases of 0.8% and 1.7%, respectively. Overall, there is a substantial enhancement in detection performance.

TABLE I
THE MAP CHANGES FOR EACH CLASS

class	YOLOv8	YOLOv8-2PCC
all	0.815	0.859
holothurian	0.794	0.861
echinus	0.934	0.942
scallop	0.605	0.685
starfish	0.928	0.945

C. *Experimental environment and configuration*

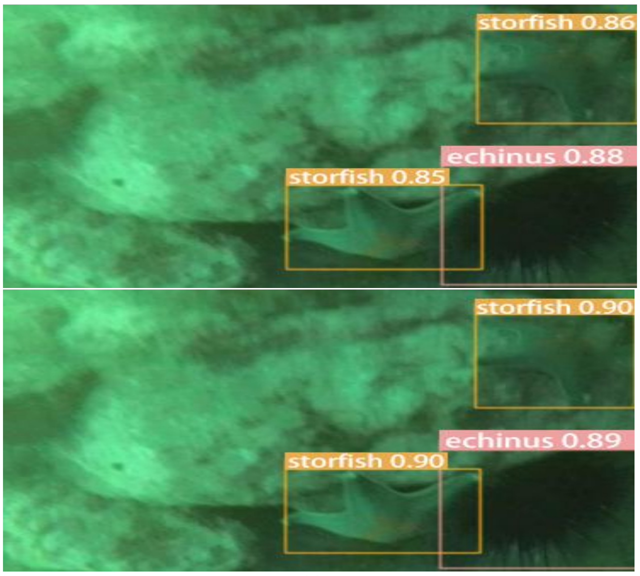
The specific experimental environment is shown in Table II. The development language of this model is Python. During training, the input images are set to 640 × 640, and the SGB function is used as the optimizer. The model is trained for 200 epochs with a batch size of 16. The momentum and decay parameters are set to 0.937 and 0.0005, respectively. The learning rate is set at 0.01, and the cosine annealing algorithm is utilized. Mosaic augmentation is applied during the last 10 epochs.

TABLE II
EXPERIMENTAL ENVIRONMENT

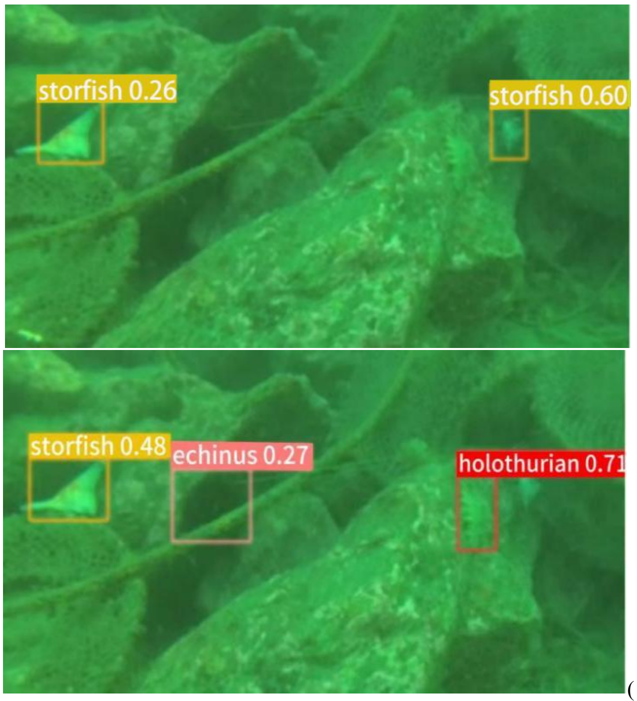
Environment	Configuration
Operating system	Ubuntu 22.04.3 LTS
CPU	Intel® Xeon(R) CPU E5-2650 V4
GPU	GeForce GTX 1080 Ti
Internal memory	96.0 GiB
Python	3.9
Pytorch	1.8.1

D. *Experimental results and analysis*

This study qualitatively evaluates the detection performance of YOLOv8 and YOLOv8-2PCC using images from two different scenarios. The experimental setup includes images sized at 640×640 pixels with a confidence threshold of 0.25. The results are depicted in Fig. 7. (The above is the result image of YOLOv8, and below is the result image of YOLOv8-2PCC.). In scenario a, where underwater creatures exhibit diversity, YOLOv8-2PCC, and YOLOv8 demonstrate improved detection accuracy, indicating their ability to extract richer features from the input images and enhance precision. In scenarios b and c, where the underwater environment is complex and organisms are often obscured, YOLOv8-2PCC outperforms YOLOv8 in detecting more small targets. Overall, YOLOv8-2PCC demonstrates a generally superior detection performance compared to YOLOv8, highlighting the network's capability to extract more comprehensive semantic information, leading to improved performance.



(a)



(b)

(c)

Fig. 7. The comparison between YOLOv8 and YOLOv8

E. Ablation Experiment

To comprehensively evaluate the detection performance of the proposed YOLOv8-2PCC algorithm, ablation experiments were meticulously designed based on the YOLOv8 framework. All experiments were conducted under the same experimental environment configuration and using identical hyperparameters, and the resulting data is presented in Table III. In the table, PC2F represents the improved C2F module proposed in this paper, CARAFE represents the upsampling operator, P2 represents the added small target detection layer, and '√' indicates the introduction of the module in that experiment group.

From Table III, it can be observed that compared to the original algorithm, in the second and fourth experiment groups, after introducing PC2F, respectively adding the CARAFE upsampling operator and the small target detection layer led to an increase in the number of parameters. mAP@0.5 and mAP@50-0.95 both increased, with a significant improvement in the fourth group, demonstrating that adding the small target detection layer is an effective means to enhance detection performance. In the third experiment group, introducing the CARAFE upsampling operator and P2 small target detection layer resulted in a slight increase in parameters, but both mAP@0.5 and mAP@0.5:0.95 showed varying degrees of improvement. This indicates that these modules can effectively enhance model detection accuracy without significantly changing the model's complexity. In the fifth experiment group, introducing PC2F, CARAFE upsampling operator, and P2 small target detection layer simultaneously led to a slight increase in parameters. mAP@0.5 increased by 4.4%, and mAP@50-0.95 increased by 6.7%, striking a balance between detection accuracy and parameter count. This demonstrates that the experimental approach in this paper can significantly improve algorithm detection performance with a modest change in parameters, making it more suitable for underwater organism detection tasks.

TABLE IV
EXPERIMENTAL COMPARISON

Model	runtime(h)	mAP@0.5	mAP@50-95
YOLOv5n	41.129	0.805	0.562
YOLOv6n	6.503	0.796	0.588
YOLOv7-tiny	61.597	0.827	0.597
YOLOv8	10.245	0.815	0.609
YOLOv8-2PCC	13.618	0.859	0.676

F. Ablation Experiment

As shown in Table IV, a comparison of the data between YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8, and YOLOv8-2PCC was conducted. The detection accuracy (mAP) of the YOLOv8-2PCC network is respectively 5.4%, 6.3%, and 3.2% higher than the faster YOLOv5n, YOLOv6n, and YOLOv7-tiny networks. In comparison to YOLOv8, the enhanced network demonstrates a 4.4% increase in detection accuracy (mAP) accompanied by a minimal increase in inference speed. Moreover, the mAP@50-95 has experienced respective increases of 11.4%, 8.8%, 7.9%, and 6.7%. It can be seen that YOLOv8-2PCC is effective.

TABLE III
EXPERIMENTAL ENVIRONMENT

group	PC2F	CARA FE	P2	Param eters(M)	mAP @0.5	mAP @50-9 5
1				3.01	0.815	0.609
2	√	√		7.56	0.826	0.625
3		√	√	3.18	0.829	0.632
4	√		√	29.06	0.849	0.662
5	√	√	√	7.73	0.859	0.676

V. Conclusion

In this experiment, we introduced a series of improvements, including incorporating PConv convolution into C2F, forming a new module called PC2F, using the CARAFE upsampling operator, and adding a dedicated small object detection layer. These enhancements aim to improve the performance of the object detection algorithm, particularly in handling small objects and upsampling semantic information. We observed that these improvements played a positive role in enhancing detection accuracy and effectiveness. The newly introduced small object detection layer excelled in detecting small objects, while the CARAFE upsampling operator contributed to better capturing semantic information. Overall, these enhancements make our object detection model more suitable for complex scenarios and small object detection tasks.

References

[1] Edgington D R, Kerkez I, Oliver D,et al. "Detecting, Tracking and Classifying Animals in Underwater Video." *OCEANS* 2006.DOI:10.1109/OCEANS.2006.306878.

[2] Chen Yuliang, Dong Shaojiang, Zhu Sunke, Sun Shizheng, & Hu Xiaolin. (2023). "Improved YOLOv3 Shallow Sea underwater organisms Target Detection." *Computer Engineering and Applications* 59.18(2023): pp.190-197.

[3] Yanchen, W. U., and Y. Wang. "A research on underwater target recognition neural network for small samples." *Journal of Northwestern Polytechnical University* 40.1(2022):40-46.

[4] "Underwater Image Enhancement using HOG and SIFT Method". *International Journal of Engineering and Advanced Technology*,2019,8(6): pp.3275-3279.

[5] Chen P C, Zou L, Wang W ."Underwater Image Transition Region Extraction and Segmentation Based on SVM." *Applied Mechanics and Materials*,2013,2658(397-400):2171-2176.

[6] Putra, M. Dalvin Marno. "Underwater Image Based Fish Detection and Recognition Using Deep Learning Algorithm." *Diss. South China University of Technology*, 2019.

[7] Li J, Chen W, Sun Y,et al.Object "Detection Based on DenseNet and RPN." *2019 Chinese Control Conference* .2019.DOI:10.23919/ChiCC. 2019.8866610.

[8] Morera N, ngel Sánchez, A. Belén Moreno, et al. "SSD vs. YOLO for Detection of Outdoor Urban Advertising Panels under Multiple Variabilities." *Sensors*, 20(16).DOI:10.3390/s20164587.

[9] Jiang Z, Wang R. "Underwater Object Detection Based on Improved Single Shot MultiBox Detector." 2020.DOI:10.1145/3446132.3446170.

[10] Qiang W, He Y, Guo Y, Li B, & He L, (2020)." Exploring underwater target detection algorithm based on improved SSD." *Xibei Gongye Daxue Xuebao/Journal of Northwestern Polytechnical University*(4).

[11] Shen, Zhou, and C. Nguyen. "Temporal 3D RetinaNet for fish detection." *2020 Digital Image Computing: Techniques and Applications (DICTA) 2020.*

[12] Liu, Kaiyue, et al. "Performance Validation of Yolo Variants for Object Detection." *BIC 2021: 2021 International Conference on Bioinformatics and Intelligent Computing 2021.*

[13] Terven J,Esparza C M D, González R A J."A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS." *Machine Learning and Knowledge Extraction*,2023,5(4): pp.1680-1716.

[14] Liu Q, Huang W, Duan X, et al. "DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n." *Electronics*, 2023,12(18).

[15] Wang, Chien Yao, A. Bochkovskiy, and H. Y. M. Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *arXiv*, 2022.

[16] Xing, Huijun, et al. "Dual Attention Based Feature Pyramid Network." *China Communications* 17.8 (2020): 11.

[17] Haifeng, Yu, et al. "Multiple attentional path aggregation network for marine object detection." *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies (2023).*

[18] Park S, Yeo Y J, Shin Y G. "PConv: simple yet effective convolutional layer for generative adversarial network." *Neural Computing and Applications*, 2022.DOI:10.1007/s00521-021-06846-2.

[19] Yan N, Liu D, Li H, & Wu, F. (2020). "Semantically Scalable Image Coding With Compression of Feature Maps." *2020 IEEE International Conference on Image Processing (ICIP). IEEE.*

[20] Wang J, Chen K, X, R, Liu Z, Loy C. C, & Lin D (2022). "Carafe++: unified content-aware reassembly of features." *IEEE Transactions on Pattern Analysis and Machine Intelligence(9),* 44.