Lightweight underwater target detection algorithm based on improved yolov8

Zeli Yang[1 [0009−0003−8174−0421]], Wenbo Zhang[1 [0009−0007−2549−2549]], Dongsheng Guo[1 [0000−0002−1648−632X] ⋆], Ziyang Zeng[1 [0009−0001−7728−5935]], Yuxing Li[1 [0009−0005−6213−7875]], and Yulong Wang[2 [0000−0002−6508−0051]]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570228, China gdongsh2022@hainanu.edu.cn

[2] School of Mechatronic Engineering and Automation Shanghai University Shanghai, China

Abstract. To address the issues of limited computing power of underwater equipment and low clarity of underwater images, an improved lightweight YOLOv8 algorithm is proposed. First, the Cross-scale Convolutional Feature-fusion Module (CCFM) is introduced to improve the model's performance in dealing with multi-scale underwater targets. The CCFM enhances the detection accuracy of small targets while reducing the number of parameters and computation. Then, the detection performance and efficiency is improved by introducing a dynamic head to unify the task-awareness, scale-awareness, and spatial-awareness. The dynamic head effectively enhances the clarity of images. Subsequently, the Mixed Local Channel Attention (MLCA) is introduced. MLCA enhances the network's ability to extract key features while ensuring the computational and detection efficiency of the model. The experimental results show that compared with the original model (yolov8n) on the publicly available underwater target detection dataset RUOD without using pretrained weights. The following is an analysis of the data. The map50 reaches 85% and improves by 0.8%, the map50-95 improves by 0.9%, the amount of parameters is reduced by 21.4%, the amount of computation is reduced to 7.4 GFLOPs, and the size of the model is reduced to 5.1M. In this paper, the original yolov8 is lightened as well as the accuracy is improved, and the improved algorithm is well suited for target detection in underwater robots.

Keywords: yolov8 · lightweight · cross-scale convolutional feature-fusion module · mixed local channel attention · dynamic head.

1 Introduction

Underwater target detection technology is a key marine technology whose background covers a wide range of fields, including marine target detection,environmental monitoring, military applications and underwater search and rescue [1].

⋆ Corresponding author

In recent years, target detection technology has developed rapidly, and traditional target detection algorithms were once all the rage [2]. Although traditional underwater target detection algorithms perform well, there are still many limitations. On the one hand, due to the need to use a sliding window to generate candidate regions, a large number of candidate regions may be generated for large images, resulting in increased computation. On the other hand, the manually designed feature extractor may not be able to adequately represent the target features, which subsequently affects the detection accuracy. The aforementioned reasons have led to the gradual replacement of traditional target detection algorithms by target detection algorithms based on deep learning. Currently popular deep learning target detection algorithms are categorized into two-stage and single-stage target detection algorithms [3]. The two-stage target detection algorithm generates candidate regions first, and then classifies and localizes according to the generated candidate regions, and the classic algorithms have the R-CNN series [4]. Although the two-stage target detection algorithm achieves higher accuracy, it requires a significant amount of computational resources. Therefore, the efficiency of this algorithm is relatively low, making it unsuitable for underwater robots that have strict demands on lightweight and real-time performance. The single-stage target detection algorithm can directly predict on the input image, without generating candidate regions [5]. Although it has lower accuracy compared to two-stage object detection algorithms, single-stage object detection algorithms are more faster and lightweight. Due to its faster speed, the single-stage algorithm is suitable for underwater robots with high demands on computation and real-time performance.

Yolo, as a classic single-stage algorithm, is applied to underwater object detection. Yang Fan et al. [6] introduced the SPD-D3 module in yolov5 [7] and added Effective Channel Attention mechanism (ECA), to the detection head. Applied to underwater target detection, it successfully improved detection accuracy, but the FPS was somewhat reduced. Tang Luting et al. [8] introduced a new Partial Convolution (PConv) into yolov7 [9] for underwater target detection. Although the detection accuracy has been greatly improved, the number of parameters, the amount of computation, and FPS have not been effectively improved. Although these algorithms ensure detection accuracy, they are not lightweight and real-time enough. Underwater robots not only require high detection accuracy, but also require lightness and real-time performance. Therefore, there is still much room for improvement in applying these algorithms to underwater robots and completing real-time tasks. Since yolov8 is able to balance precision, lightweight, and real-time performance very well, so this article uses it for improvement.

Considering the unique characteristics of underwater environments and the high demand for lightweight and real-time performance in underwater robots, we have chosen to improve the yolov8n algorithm. Our main focus is on reducing the number of parameters and computations while maintaining accuracy, enhancing real-time performance, and minimizing the model size. The key work presented in this paper is as follows:

1.Adding the CCFM into the neck of the network enables fusion of features from different scales. This fusion improves the model's performance when dealing with multi-scale underwater targets, enhances the ability of detecting small underwater targets, and reduces the number of parameters and computation.2.A novel dynamic head is introduced to replace the original detection head. It integrates three attention mechanisms of task perception, scale perception, and spatial perception into the detection head. Each attention module focuses on one dimension, making the feature maps clearer and improving the detection capability of underwater targets. 3.Employing the lightweight Multi-Level Context Attention (MLCA) module to ensure the computational and detection efficiency of the model. It enhances the ability of the network to capture key features of underwater targets, improving the network's feature selection capability.

2 Yolov8 Description

Yolov8, building upon the success of previous yolo series algorithms, introduces a series of further improvements [10]. The main structure of yolov8 is divided into two parts: the backbone and the head, without a separate neck part, as the content of the neck is integrated into the head part.

In the Backbone section, yolov8 replaces the C3 module used in yolov5 with the C2f module and references several bottleneck modules to improve the nonlinear representation of the model. And the split module splits the feature map into two parts to reduce the number of parameters and computation [11]. The Head part, replacing the coupling head in yolov5 with a decoupling head containing two branches, one predicting location and one predicting category. Enables more efficient handling of semantic information at different scales and finenesses by separating feature extraction from pixel-level prediction. Yolov8 discards the previous anchor-based approach used by yolov5 and changes to use anchor-free, which gets rid of the dependence on the a priori knowledge in the dataset. And significantly improves the expressiveness and generalization potential of 'object shapes', which enables it to detect moving objects, objects of varying sizes, and also shows better performance and better flexibility in dealing with occluded objects [12]. Finally, PAN-FPN removes the convolutional module from the upsampling, reducing the computational effort. For the loss function part, yolov8 uses VFL loss as the classification loss, and DFL loss and CIOU loss as the bounding box regression loss. Yolov8's sample matching uses an innovative task alignment assignment mechanism. It scores based on the alignment degree between the labeled bounding boxes and the predicted bounding boxes, assigning anchor points to specific labeled bounding boxes [13]. The task alignment assignment mechanism significantly reduces the error between labeling and prediction, thus improving the accuracy of the model. For the training part, yolov8 turned off Mosaic data augmentation in the last 10 epochs of training, a method that effectively improves model accuracy.

The aforementioned improvements make yolov8 more suitable for underwater target detection. However, there are still some areas that require further opti-

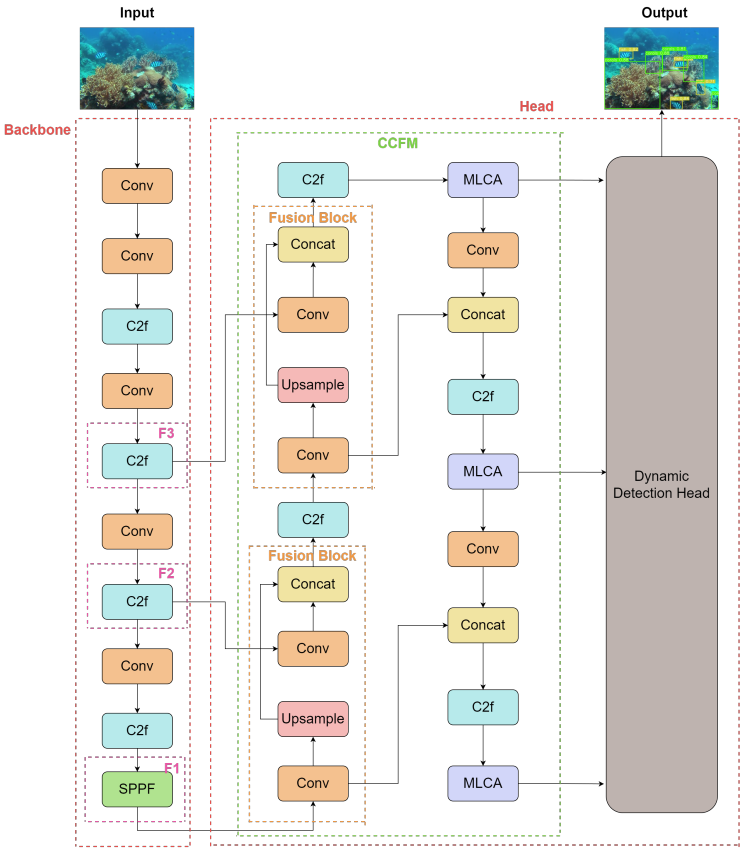


Fig. 1: Structure of improved yolov8 network

mization. For underwater robots with limited computing power, there are high demands on the lightweight, real-time performance, and accuracy of the target detection model. To better suit underwater robots, this paper will improve the yolov8 algorithm from these aspects.

3 Improved yolov8

Given the limited computing power of underwater robots, the target detection model applied to them needs to be more lightweight and have higher real-time performance. To address these issues, the following details the improvements made to yolov8 in this paper. The network structure diagram of the improved yolov8 is shown in Figure 1.

3.1 Lightweight Feature Fusion Module

The Cross-scale Convolutional Feature-fusion Module(CCFM) is a lightweight cross-scale feature fusion module that employs lightweight convolutions [14]. By reducing the number of channels, it decreases the number of parameters and computations, ensuring computational efficiency while improving the performance of target detection. Given the limited computational capacity of underwater robots, it is difficult for them to run large models in real-time. The lightweight characteristics of the feature fusion module CCFM enable it to reduce the number of parameters and computations while maintaining detection accuracy and enhancing detection efficiency. The CCFM module enables the algorithm to be easily deployed on underwater robots and complete underwater target detection tasks in real-time. It fully meets the requirements of underwater robots for detection tasks.

CCFM employs a series of convolutional layers designed with lightweight techniques. It extracts key information from the input feature maps, effectively reducing the computational complexity. After feature extraction, we utilize a lightweight convolutional operation to integrate the feature maps. By adopting 1x1 convolutional kernels, we successfully achieve the horizontal fusion of feature maps of different scales, further enhancing the representation ability of features. The fused feature maps are then passed through a series of 1x1 convolutional layers for further processing. These layers further enhance the representation ability of the feature maps, providing more accurate and rich feature information for subsequent object detection tasks. Finally, the processed feature maps are passed to the subsequent detection head for object prediction and recognition. This step is crucial in transforming the feature information into actual object detection results.The calculation process of CCFM is as follows:

$$output = CCFM(F_1, F_2, F_3). \tag{1}$$

Among them, the structures of F_1, F_2, F_3 and $CCFM$ are labeled in Figure 1.

CCFM utilizes multiple Fusion modules, which fuse multi-scale features. It gradually integrates multi-scale features from the bottom to the top in the backbone network, and ultimately generates fused features rich in strong semantic information. This provides strong support for subsequent underwater target detection tasks. The Fusion Block module is adopted in CCFM to complete multi-scale fusion operations. When applied in underwater target detection, it can balance the detection effect of targets of different scales very well. In addition, it improves the detection effect of different targets, especially for small targets.

3.2 New Dynamic Detection Head

The detection head part adopts a new type of dynamic detection head, named Dynamic Head(dyhead). In underwater feature maps, there are often a lot of redundancies and noises. In order to address such issues, this detection head

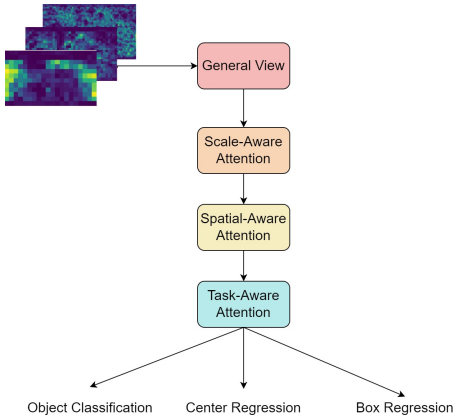


Fig. 2: Structure of Dynamic head

has been proposed. Its innovation lies in integrating the three key elements of scale perception, spatial perception, and task perception into a unified framework [15]. The attention mechanism is effectively applied in the target detection head, improving the performance and efficiency of target detection, especially in terms of accuracy. Its structural diagram is shown in Figure 2.

After being processed by the feature fusion module, the feature maps are then passed through the scale-aware, spatial-aware, and task-aware attention modules. Each attention mechanism focuses on only one dimension, making the feature maps clearer and more focused.

The calculation process of Dynamic Head is as follows:

$$Output(\mathcal{I}) = A_C\left(A_S\left(A_L(\mathcal{I})\cdot\mathcal{I}\right)\cdot\mathcal{I}\right)\cdot\mathcal{I} \tag{2}$$

Among them, the input feature tensor $\mathcal{I} \in \mathcal{R}^{L\times S\times C}$, and A_C, A_S and A_L represent task-aware attention, spatial-aware attention, and scale-aware attention respectively.

The spatial-aware attention module focuses attention on discriminative regions that consistently exist across spatial locations and feature levels. First, we utilize deformable convolution techniques to sparsify attention learning, enabling the model to focus more on key regions. Subsequently, we further integrate feature information from different levels at the same spatial positions, leveraging multi-scale features to improve the detection efficiency of objects of varying sizes.The calculation process of spatial-aware attention is as follows:

$$A_S(\mathcal{I})\cdot\mathcal{I} = \frac{1}{L}\sum_{l=1}^{L}\sum_{k=1}^{K} w_{l,k}\cdot\mathcal{I}\left(l; p_k + \Delta p_k; c\right)\cdot\Delta m_k \tag{3}$$

Among them, L stands for layer, K represents the number of sparse sampling locations, $p_k + \Delta p_k$ is a shifted location by the self-learned spatial offset Δp_k to focus on a discriminative region and Δm_k is a self-learned importance scalar at

location p_k. Both are acquired through a learning process that takes into account the input features extracted from the median layer of the feature hierarchy I.

The task-aware attention module utilizes a switch control mechanism. We can automatically select whether to learn the activation threshold based on task requirements. It enables the intelligent opening or closing of channels to adapt to different task scenarios. Through the Task-Aware Attention module, the feature maps are able to generate unique activation patterns for different downstream tasks, thus achieving targeted feature extraction and representation. The calculation process of task-aware attention is as follows:

$$A_C(\mathcal{I}) \cdot \mathcal{I} = \max\left(\alpha^1(\mathcal{I}) \cdot \mathcal{I}_c + \beta^1(\mathcal{I}), \alpha^2(\mathcal{I}) \cdot \mathcal{I}_c + \beta^2(\mathcal{I})\right) \tag{4}$$

Among them, I_c is the feature slice at the $c-th$ channel, $[\alpha^1, \alpha^2, \beta^1, \beta^2]^T = \theta()$ is a hyper function that learns to control the activation thresholds. $\theta()$ initially employs a global average pooling across the L × S dimensions, aiming to decrease the data's dimensionality. Following this, it incorporates two sequentially arranged fully connected layers, accompanied by a normalization layer. Finally, it applies a shifted sigmoid function to normalize the output to [1, 1].

The scale-aware attention module can dynamically fuse features of different scales based on the importance of semantic information at each scale, enabling more precise feature extraction and fusion operations. Under the effect of the scale-aware attention module, the feature maps exhibit higher sensitivity to changes in different scales. This successfully enhances the model's ability to recognize multi-scale targets. When applied in underwater target detection, it can achieve good detection results for both large and small targets.The calculation process of scale-aware attention is as follows:

$$A_L(\mathcal{I}) \cdot \mathcal{I} = \sigma\left(f\left(\frac{1}{SC}\sum_{S,C}\mathcal{I}\right)\right) \cdot \mathcal{I} \tag{5}$$

Among them, $f()$ is a linear function, and $\sigma()$ is a hard-sigmoid function.

3.3 Mixing Local Channel Attention

The Mixed Local Channel Attention (MLCA) is a lightweight attention mechanism that improves detection accuracy while maintaining almost the same number of parameters and computational complexity. This module integrates channel information with spatial information, while also combining local and global features. It effectively enhances the expressive ability of the network, thus improving its performance [16]. Its structural diagram is shown in Figure 3.

The input feature map first undergoes Local Average Pooling(LAP) to aggregate local regional features. Then, the output feature map is divided into two branches for further processing. One branch is passed to Global Average Pooling(GAP) to capture the spatial information of the entire feature map, generating a compact feature representation. Feature rearrangement is then performed,

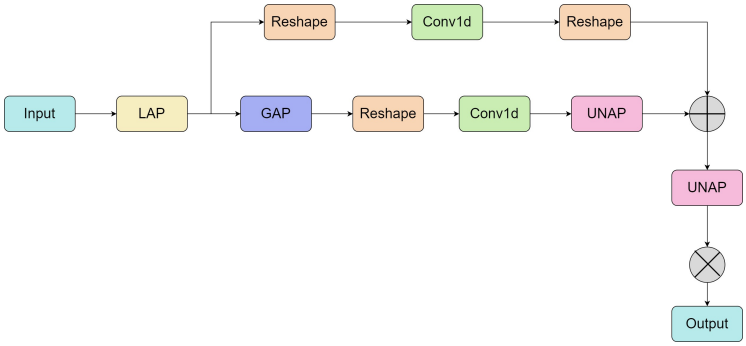


Fig. 3: Structure of MLCA

followed by a one-dimensional convolution to compress the feature channels, reducing the number of parameters and computational cost. The processing of this branch is completed through an inverse pooling operation [17]. The other branch first undergoes feature rearrangement, followed by a one-dimensional convolution to compress the feature channels, reducing the number of parameters and computational cost. Feature rearrangement is then performed to complete the processing of this other branch.

After both branches have been processed, the results from global pooling and local pooling are summed. This integrates global context information and is then restored to the original spatial dimension through an inverse pooling operation. Finally, the result from the previous step is multiplied with the original input feature. This process essentially serves as a feature screening mechanism that enhances the importance and attention to valuable features of underwater targets.

4 Experimental Design and Result Analysis

4.1 Experimental Details

Experimental Environment and Model Hyperparameter Settings Table 1 below describes the hyperparameter configuration of the environment and model for this experiment.

Experimental Data Set The experimental dataset used in this thesis is Rethinking general Underwater Object Detection (RUOD) [18]. The RUOD dataset consists of a total of 14,000 images, specifically divided into 9,800 training images and 4,200 test images, with a total of 74,903 labeled objects. The dataset contains 10 common aquatic categories, namely holothurian, echinus, scrollop , starfish, fish, corals, diver, cuttlefish, turtle, jellyfish . The dataset has rich biological species and complex environmental backgrounds, which can be used to comprehensively evaluate the performance of target detection algorithms.

Table 1: **Experimental environment and parameters**

Parameter	Configuration
CPU	12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU 2.50GHz
GPUs	RTX 3090 (24GB)
CUDA	11.1
Operating System	Ubuntu 18.04
Python	3.8.10
PyTorch	1.8.1
Momentum	0.937
Weight Decay	0.0005
Batch Size	64
Learning Rate	0.01
Image Size	640
Epochs	300

Model evaluation indicators Considering that the target detection algorithm applied to underwater robots needs to take into account both detection accuracy and model lightweighting, this experiment chooses to use Precision (P), Recall (R), mean Average Precision (mAP), Parameter quantity (Parameters), computational volume Giga FLoating-point Operations Per second (GFLOPs), and model size as the main evaluation indexes for the performance of the improved model. The calculation formula is as follows:

$$P = \frac{TP}{TP + FP}, \tag{6}$$

$$R = \frac{TP}{TP + FN}, \tag{7}$$

$$AP = \int_0^1 P(R)dR, \tag{8}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n}. \tag{9}$$

Among them, TP denotes a positive sample predicted to be a positive case, FP denotes a negative sample predicted to be a positive case, FN denotes a positive sample predicted to be a negative case, P denotes precision (check accuracy), R denotes recall (check completeness), P(R) denotes the value as a function of the curve with recall as the horizontal axis and precision as the vertical axis, P-R denotes the number of categories, AP denotes the average precision, and mAP denotes mean average precision [19].

4.2 Comparison Of ablation experiments

In order to verify the validity of this experimental improvement, ablation experiments were done here for validation [20]. It is worth mentioning that the

Table 2: **Comparison of Ablation experiment results**

Model	CCFM	dyhead	MLCA	Parameters	GFLOPs	FPS	Map50	Model Size
yolov8-n	×	×	×	3.01M	8.1	588	84.2%	6.15MB
model a	√	×	×	1.97M	6.6	625	83.9%	4.12MB
model b	×	√	×	4.76M	14.7	313	86.0%	9.60MB
model c	×	×	√	3.01M	8.1	625	84.6%	6.16MB
model d	√	√	×	2.36M	7.4	357	84.9%	4.93MB
model e	√	×	√	1.97M	6.7	625	84.0%	4.13MB
model f	×	√	√	2.51M	7.9	370	84.8%	5.20MB
model g	√	√	√	2.36M	7.4	385	85.0%	4.93MB

experimental environments used here are consistent to minimize the influence of irrelevant variables on the experiments. The test benchmark is YOLOv8n, with different improvement modules added separately. The results of the ablation experiments are shown in Table 2.

It can be seen from the experimental results in the table that the improvement has significant effects in terms of lightweight and accuracy. The introduction of the lightweight feature fusion module CCFM has made the lightweight improvement effect obvious. The number of parameters, computation, and model size have all been significantly reduced, with a slight improvement in inference speed. This is of great help to the deployment and real-time operation of underwater robots. However, CCFM will slightly reduce mAP. To improve detection accuracy, a new dynamic detection head, Dynamic Head, is introduced. In terms of accuracy, such as mAP50, mAP50-95, precision, and recall rate, there have been significant improvements, especially mAP50 has increased by 1.8%. However, the improvement in the number of parameters, computation, inference speed, and model size is not satisfactory, which is difficult to meet the task requirements of underwater robots. To solve this problem, we integrated CCFM with Dynamic Head. It has excellent detection results for multi-scale targets, especially small targets. After integration, both the number of parameters and computation have been reduced, further adapting to the low computing power characteristics of underwater robots. In addition, the model size has also been significantly reduced, providing convenience for deployment on underwater robots. To improve the robustness of the network, a lightweight hybrid local channel attention mechanism, MLCA, is introduced. It combines local and global features while paying attention to the overall and details of the image. There is a slight increase in inference speed and mAP, while the number of parameters, computation, and model size remain basically unchanged. After integrating these modules to obtain the algorithm in this paper, each improved module has played its own advantages. Both lightweight and detection accuracy have been improved, which can well adapt to the working requirements of underwater robots.

Table 3: **Comparison of ablation experiment results**

Model	Parameters	GFLOPs	FPS	Map50	Model Size
RT-DETR-l	28.46M	100.6	185	84.6%	56.3MB
yolov5-n	1.78M	4.3	118	83.2%	3.86MB
yolov6-n	4.63M	11.3	578	84.4%	40.2MB
yolov7-tiny	6.03M	13.1	313	85.3%	12.10MB
yolov8-n	3.01M	8.1	588	84.2%	6.15MB
yolov8-n-CCFM-dyhead-MLCA	2.36M	7.4	385	85.0%	4.93MB

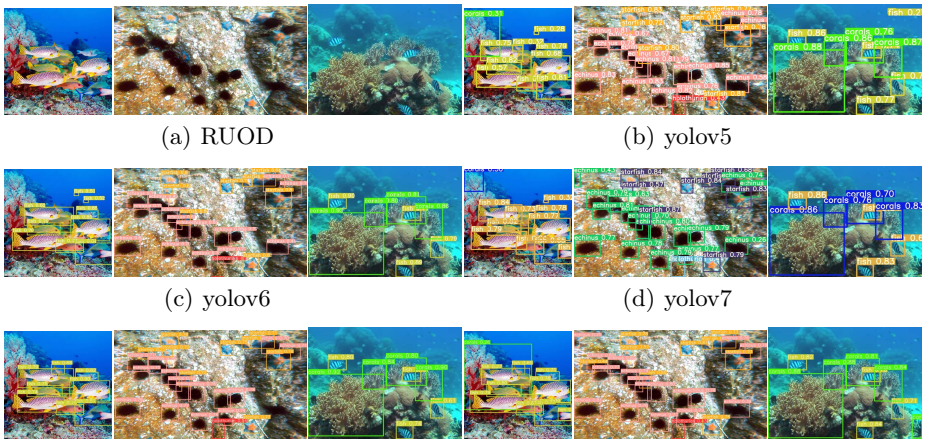
4.3 Comparison of Different models

To validate the effectiveness of the algorithm presented in this paper, it is compared with yolov5, yolov6 [21], yolov7, and yolov8. Here, it is ensured that the parameters such as experimental environment, dataset, number of training rounds, learning rate, etc. are consistent, and the possibility of irrelevant variables affecting the experiments is minimized as much as possible. The results of the comparison experiments are shown in Table 3.

As can be seen from Table 3, using RUOD dataset, The improved yolov8 algorithm in this paper is more suitable for underwater application scenarios. Considering parameters, computation, inference speed, and mAP comprehensively, the algorithm presented in this paper performs well. While RT-DETR excels in achieving remarkable accuracy, it also introduces an increased number of parameters and computational intricacies, ultimately leading to a substantial degree of redundancy. Yolov5 has fewer parameters and computations, seemingly more lightweight. However, its inference speed lags behind yolov6, yolov7 and yolov8, making it difficult to meet the high real-time requirements of underwater robots. Yolov6's mAP improvement is not significant, and it has significant disadvantages in terms of lightweight, so it is excluded first. Yolov7-tiny boasts the highest detection accuracy. However, due to its excessive number of parameters and computations, as well as its inferior inference speed, it is also excluded. Although yolov5's parameters and computations are slightly lower than yolov8, its mAP and real-time performance are inferior. It is difficult to meet the requirements of underwater robots for lightweight, real-time, and detection accuracy. This paper improves the yolov8 algorithm, further reducing the number of parameters, computations, and model size, while also increasing the mAP. Although the inference speed is slightly lower than the original model, it can still meet the real-time requirements underwater. In summary, while meeting the requirements of real-time detection, the algorithm in this paper has improved the detection accuracy by 0.8% compared with the baseline model, reduced the number of parameters by 21.6%, decreased the amount of computation by 8.6%, and shrunk the model size by 19.8%. This has improved the target detection capability of underwater robots.

Here is a selection of comparison charts showing the detection results of different algorithms, as illustrated in Figure 4. By analyzing the detection result

charts, it can be found that the improved algorithm has certain advantages in detecting small underwater targets and occluded objects in the water.



(a) RUOD (b) yolov5

(c) yolov6 (d) yolov7

(e) yolov8 (f) ours

Fig. 4: Comparison of detection results of different algorithms τ

5 Conclusion

There are numerous challenges facing the current target detection algorithms applied to underwater robots. The main issues include the difficulties in deploying overly large models, failure to meet real-time performance standards and low detection accuracy for small underwater targets. This paper studies a lightweight object detection algorithm suitable for underwater environments and improves yolov8. The improved algorithm better meets the detection accuracy and lightweight requirements of underwater robots. The introduction of a lightweight cross-scale feature fusion module effectively reduces the number of parameters and computations, while slightly improving the inference speed. A new dynamic detection head, Dynamic Head, is adopted, which applies multiple attention mechanisms to the detection head. Ensuring that the computational cost and number of parameters are within an acceptable range, while also improving image clarity and detection accuracy for multi-scale targets. A lightweight mixed local channel attention mechanism is introduced to further enhance the detection capability for small objects. From the results, there is also some enhancement in real-time performance. Integrating the above modules into yolov8 significantly reduces the number of parameters and computations. At the same time, it reduces the model size, facilitating the deployment of the algorithm to underwater robots. Additionally, the detection performance for small objects and occluded objects has been improved. The improved algorithm achieves both

lightweight and accuracy. Compared to the original yolov8, the improved algorithm in this paper is more suitable for application to underwater robots or underwater scenes.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (62463004) and Scientific Research Fund of Hainan University (KYQD(ZR)23025).

References

1. M. Zhang, S. Xu, W. Song, Qi. H, and Q. Wei. Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion. *Remote Sensing*, 13(22):4706, 2021.
2. B. Li, W. Jiang, and J. Gu. Research on target detection algorithm based on deep learning technology. In *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*, pages 137–142. IEEE, 2021.
3. L. Wei, X. Feng, K. Zha, S. Li, and H. Zhu. Summary of target detection algorithms. In *Journal of Physics: Conference Series*, volume 1757, page 012003. IOP Publishing, 2021.
4. F. Liang, Y. Zhou, X. Chen, F. Liu, C. Zhang, and X. Wu. Review of target detection technology based on deep learning. In *Proceedings of the 5th International Conference on Control Engineering and Artificial Intelligence*, pages 132–135, 2021.
5. T. Jiang, X. Mu, X. Wei, Z. Zeng, and R. Luo. Research progress of single-stage small target detection based on deep learning. In *2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM)*, pages 893–898. IEEE, 2022.
6. F. Yang, Z. Li, Y. Wang, and D. Guo. An underwater object detection algorithm based on improved yolov5s. In *2023 2nd International Conference on Advanced Sensing, Intelligent Manufacturing (ASIM)*, pages 83–87. IEEE, 2023.
7. F. Lei, F. Tang, and S. Li. Underwater target detection algorithm based on improved yolov5. *Journal of Marine Science and Engineering*, 10(3):310, 2022.
8. L. Tang and H. Huang. Lightweight underwater object detection algorithm based on yolov7. *Electronics Optics and Control*, pages 1–9, 2024.
9. K. Liu, Q. Sun, D. Sun, L. Peng, M. Yang, and N. Wang. Underwater target detection based on improved yolov7. *Journal of Marine Science and Engineering*, 11(3):677, 2023.
10. F. M Talaat and H. ZainEldin. An improved fire detection approach based on yolo-v8 for smart cities. *Neural Computing and Applications*, 517:243–256, 2023.
11. H. GENG, Z. Liu, J. JIANG, Z. FAN, and J. LI. Embedded road crack detection algorithm based on improved yolov8. *Journal of Computer Applications*, 44(5):1613, 2023.
12. Y. Zhang, H. Zhang, Q. Huang, Y. Han, and M. Zhao. Dsp-yolo: An anchor-free network with dspan for small object detection of multiscale defects. *Expert Systems with Applications*, 241:122669, 2024.
13. X. Wang, H. Gao, Z. Jia, and Z. Li. Bl-yolov8: An improved road defect detection model based on yolov8. *Sensors*, 23(20):8361, 2023.

14. Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, and J. Chen. Detrs beat yolos on real-time object detection. *arXiv preprint arXiv:2304.08069*, 2023.
15. X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang. Dynamic head: Unifying object detection heads with attentions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7373–7382, 2021.
16. D. Wan, R. Lu, S. Shen, T. Xu, X. Lang, and Z. Ren. Mixed local channel attention for object detection. *Engineering Applications of Artificial Intelligence*, 123:106442, 2023.
17. T. Y. Hsiao, Y. C. Chang, H. H. Chou, and C. T. Chiu. Filter-based deep-compression with global average pooling for convolutional networks. *Journal of Systems Architecture*, 95:9–18, 2019.
18. C. Fu, R. Liu, X. Fan, P. Chen, H. Fu, W. Yuan, M. Zhu, and Z. Luo. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing*, 2023.
19. W. He, Z. Huang, Z. Wei, C. Li, and B. Guo. Tf-yolo: An improved incremental network for real-time object detection. *Applied Sciences*, 9(16):3225, 2019.
20. Y. Lu, L. Zhang, and W. Xie. Yolo-compact: an efficient yolo network for single category real-time object detection. In *2020 Chinese control and decision conference (CCDC)*, pages 1931–1936. IEEE, 2020.
21. C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.