electronics

MDPI

Article

An Improved YOLOv5 Underwater Detector Based on an Attention Mechanism and Multi-Branch Reparameterization Module

Jian Zhang [1,2,†], Hongda Chen [2,†], Xinyue Yan [2], Kexin Zhou [2], Jinshuai Zhang [2], Yonghui Zhang [1,*], Hong Jiang [1] and Bingqian Shao [2]

[1] School of Information and Communication Engineering, Hainan University, Haikou 570228, China
[2] School of Applied Science and Technology, Hainan University, Haikou 570228, China
* Correspondence: yhzhang@hainanu.edu.cn
† These authors contributed equally to this work.

Abstract: Underwater target detection is a critical task in various applications, including environmental monitoring, underwater exploration, and marine resource management. As the demand for underwater observation and exploitation continues to grow, there is a greater need for reliable and efficient methods of detecting underwater targets. However, the unique underwater environment often leads to significant degradation of the image quality, which results in reduced detection accuracy. This paper proposes an improved YOLOv5 underwater-target-detection network to enhance accuracy and reduce missed detection. First, we added the global attention mechanism (GAM) to the backbone network, which could retain the channel and spatial information to a greater extent and strengthen cross-dimensional interaction so as to improve the ability of the backbone network to extract features. Then, we introduced the fusion block based on DAMO-YOLO for the neck, which enhanced the system's ability to extract features at different scales. Finally, we used the SIoU loss to measure the degree of matching between the target box and the regression box, which accelerated the convergence and improved the accuracy. The results obtained from experiments on the URPC2019 dataset revealed that our model achieved an mAP@0.5 score of 80.2%, representing a 1.8% and 2.3% increase in performance compared to YOLOv7 and YOLOv8, respectively, which means our method achieved state-of-the-art (SOTA) performance. Moreover, additional evaluations on the MS COCO dataset indicated that our model's mAP@0.5:0.95 reached 51.0%, surpassing advanced methods such as ViDT and RF-Next, demonstrating the versatility of our enhanced model architecture.

Keywords: YOLOv5; deep learning; object detection

Citation: Zhang, J.; Chen, H.; Yan, X.; Zhou, K.; Zhang, J.; Zhang, Y.; Jiang, H.; Shao, B. An Improved YOLOv5 Underwater Detector Based on an Attention Mechanism and Multi-Branch Reparameterization Module. *Electronics* **2023**, *12*, 2597. https://doi.org/10.3390/electronics12122597

Academic Editor: Fernando De la Prieta Pintado

Received: 10 April 2023
Revised: 6 June 2023
Accepted: 7 June 2023
Published: 8 June 2023

1. Introduction

With the vast majority of the world's oceans still unexplored, the ability to accurately detect and locate underwater targets such as minerals, oil and gas deposits, and marine life is essential for sustainable and effective resource management. Underwater target detection technology includes a variety of methods, such as sonar and acoustic imaging [1–3], magnetic and electromagnetic sensing [4,5], and visual inspection using remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs). These technologies allow researchers and industry professionals to map the seafloor, identify potential resource deposits, and locate marine life for conservation or fishing purposes. However, one major disadvantage of acoustic imaging is that sound waves may be absorbed or scattered by various obstacles, leading to decreased resolution and difficulty in detecting small or distant objects. In addition, it may be affected by environmental factors such as water temperature and salinity, which can further complicate the imaging process. Similarly, electromagnetic sensing imaging may also be hindered by physical obstacles and the properties of the materials being imaged. For example, electromagnetic waves may be blocked or distorted

by conductive materials, leading to incomplete or inaccurate imaging results. Both of these require specialist and expensive equipment. In contrast, the images obtained through the visible light band can be of higher resolution and lower cost and are more visual, which makes it a widely used solution.

Nevertheless, in the challenging underwater environment, the obtained image still suffers from different degrees of degradation, such as color distortion [6–10], low light and contrast [11,12], and haze-like effects [11–19]. The original YOLOv5 model is easily affected by the above problems. It does not have an effective mechanism to protect the network from these less important or even harmful features. An attention mechanism could greatly alleviate this problem. The idea behind learning the attention weight is to let the network narrow and lock the focus area, before finally forming the focus of attention, which is very important for target detection. This mechanism helps to refine the perceptual information while preserving its context. In the past few years, many efforts have been made to effectively integrate various attention mechanisms into the deep convolution neural network architecture to improve the performance of detection tasks, and these have proven effective.

The deployment of underwater target detectors is constrained by limited hardware resources, which requires the detector to achieve high detection accuracy with inadequate hardware support. YOLOv5 uses the same structure for training and inferencing, which not only limits its accuracy, but also increases the requirements for the hardware standards of the model on the inferencing side. Inspired by DAMO-YOLO [20], we believe that the introduction of reparameterization and a fusible structure is a suitable way to solve this problem.

Underwater objects present unique challenges to the detector due to their small size, being hidden from view, or being situated against a complex background. These factors increase the demands placed on the detector's regression branch. Traditional IoU loss functions (i.e., DIoU/CIoU [21], GIoU [22], EIoU [23], and ICIoU [24]) predict the distance, overlapping area, and aspect ratio of the ground truth box. However, these loss functions are not effective when the directions of the ground truth box and the predicted box are inconsistent. This defect causes the position of the prediction box to fluctuate continuously during the training process, resulting in slower model convergence and lower accuracy.

In this paper, we propose a new underwater target detector based on YOLOv5 to address the above problems. Our contributions are detailed as follows:

- We added a global attention mechanism (GAM) to YOLOv5 to help the backbone focus on the key area, avoiding confusion due to the challenging underwater backgrounds.
- Inspired by DAMO-YOLO, we introduced a multi-branch reparameterized structure to improve the aggregation of multi-scale features, which made our model more accurate and robust under complex conditions.
- We introduced the SIoU loss function to improve the accuracy and accelerate the convergence.
- The proposed underwater target detector takes into account the smaller computational overhead and higher computational accuracy. The experimental results on the URPC2019 dataset showed that the mAP@0.5 of our model was 1.8% and 2.3% higher than that of YOLOv7 and YOLOv8, respectively. In addition, supplementary experiments on the COCO dataset proved that our improvement can also be applied to land target detection.

2. Related Works

2.1. Object Detection

In terms of structure, target detectors based on deep neural networks can be divided into three parts, namely the backbone, neck, and head. The backbone is used to extract image features. Common backbones include VGG [25], ResNet [26], CSPNet [27], and Swin Transformer [28], and lightweight backbone networks include ShuffleNet [29,30], MobileNet [31–33], and RepVGG [34].

The neck is designed to make more effective use of the features extracted from the backbone network and give play to the advantages of multi-scale features. The current designs tend to use several top-down and bottom-up paths for connection, so as to facilitate the aggregation of backbone network features at different stages. In the field of underwater object detection, the SA-FPN [35] fully utilizes the pyramid structure to perceive features at different scales. Other popular neck networks include the FPN [36], PAN [37], BiFPN [38], ASFF [39], and RepGFPN [20].

The head is usually divided into one-stage target detectors and two-stage target detectors. Their main difference lies in whether they predict the object category and boundary box at the same time. Most early target detection models used two-stage target detectors, such as the famous RCNN model [40] and the subsequent variants Fast RCNN [41], Faster RCNN [42], and Mask RCNN [43]. Boosting RCNN [44] is the latest underwater target detector based on the RCNN, with detection accuracy surpassing many previous two-stage detectors. These two-stage target detectors have the advantage of high accuracy, but it is difficult for the referencing speed to meet the needs of real-time detection. In contrast, single-stage target detectors have faster inference speed, and after years of development, these detectors have also achieved a relatively high accuracy. Popular one-stage target detectors include SSD [45], RetinaNet [46], and the YOLO series [47–54].

2.2. Attention Mechanism

In the field of computer vision, researchers use attention mechanisms to improve the performance of networks. Common attention mechanisms can be divided into three categories: channel attention, spatial attention, and channel and spatial attention.

With the proposal of squeeze-and-excite networks (SENets) [55], efficient channel attention calculation became an important way to improve the performance of networks. SENets have a simple structure and remarkable effect. They can adjust the feature response between channels through feature recalibration. The important components of channel attention include the global second-order pooling block (GSoP) [56], the style-based recalibration module (SRM) [57], the effective channel attention block (ECA) [58], and the bilinear attention block (Bi-attention) [59].

As for spatial attention mechanisms, the recurrent attention model (RAM) [60] was the first to incorporate RNNs in its visual attention mechanism, which have since been adopted by a range of other RNN-based methods. The Glimpse network [61], similar to the RAM, was based on how humans perform visual recognition, and it proposed that the network take Glimpse as the input in order to update its hidden state, demonstrating its effectiveness.

The global attention mechanism (GAM) [62] used in this paper is a channel and spatial attention mechanism. By utilizing the channel and spatial attention mechanism, the model can dynamically weigh the importance of different channels and spatial locations of the input features. This allows the model to selectively focus on the most-significant features and areas, enhancing its ability to capture relevant information and suppress noise. As a result, the channel and spatial attention mechanism provides the best of both worlds, effectively combining the benefits of channel and spatial attention mechanisms to achieve superior performance on the target detection task.

2.3. IoU Loss

In object detection, the IoU loss is used to measure the overlap between the prediction box and the ground truth box. It effectively prevents the interference of the boundary box size in the form of proportion.

The earliest IoU loss [63] had two main disadvantages. First, when the prediction box and the ground truth box did not intersect, whether the distance between them was near or far, the calculated IoU was always zero, so the distance could not be measured. Second, when the intersection ratio of the prediction box and the ground truth box was the same, it was again impossible to determine their relative locations.

In order to solve the first problem of the IoU loss, i.e., that when there was no intersection between the prediction box and the ground truth box, the distance could not be measured, the GIoU [22] introduced the concept of the minimum closure area. This refers to the smallest rectangular box that can surround the prediction box and the ground truth box.

For the second problem, the DIoU [21] was proposed. On the basis of the GIoU, it introduces the distance loss between the center points of the prediction box and the ground truth box. Based on the DIoU loss, the difference between the aspect ratios of the prediction box and the ground truth box was introduced as the CIoU loss. This improved the situation in the following three ways [64]:

- Increasing the overlapping area of the ground truth box and the predicted box;
- Minimizing the distance between their center points;
- Maintaining the consistency of the boxes' aspect ratios.

However, none of these losses considers the direction of the mismatch between the prediction box and the ground truth box. This shortfall leads to slow convergence and low accuracy.

3. Methodology

We retained the overall architecture of YOLOv5 [51] and improved or replaced some of its modules. In this section, we will introduce YOLOv5 and our improvements to the backbone, neck, and IoU loss. The overall architecture of our improved YOLOv5 network is shown in Figure 1.

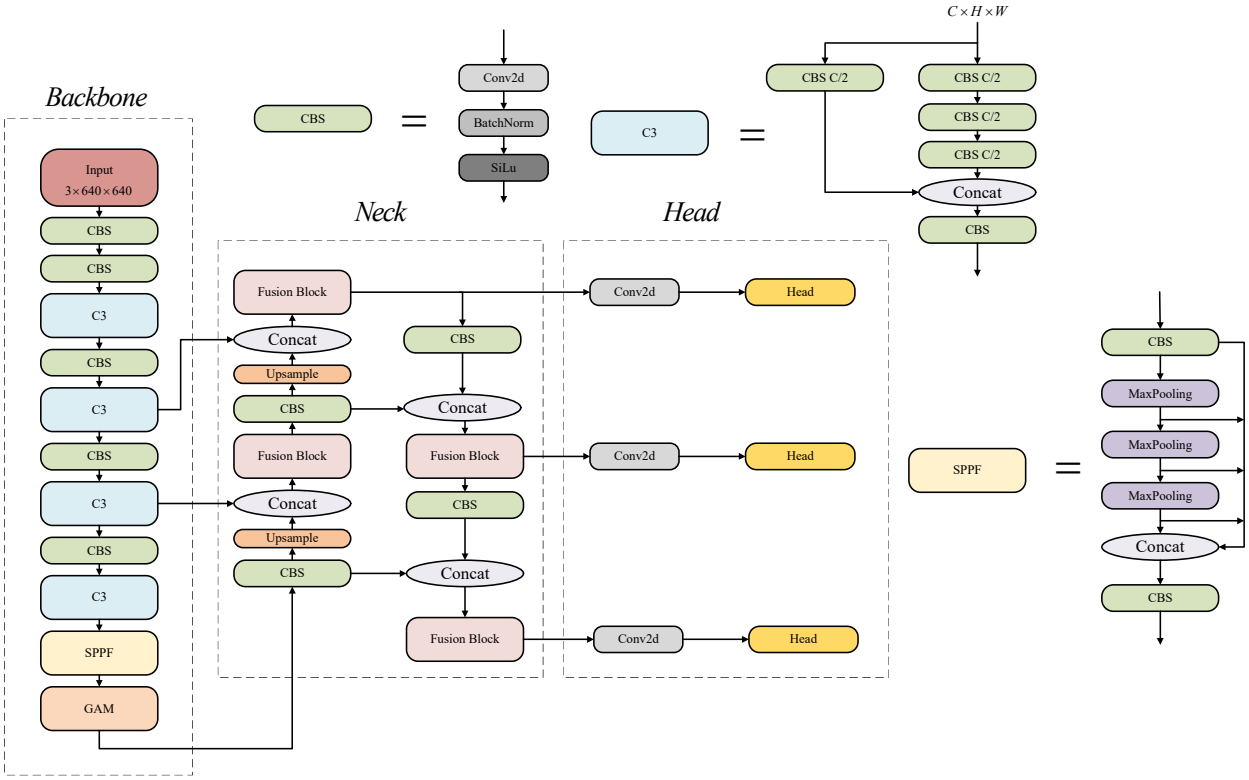

Figure 1. Overall architecture of our method. We introduce the network below by describing the backbone, neck, and head. See Figures 2 and 5 for schematics of the GAM and fusion block.

3.1. Backbone

The basic component of the backbone network of YOLOv5 is the CBS module, which is a convolution–BatchNorm–SiLu activation function sequence superposition module. These basic components are stacked together, and C3 modules are inserted in the middle. A C3

module is a combination of CBS blocks under the guidance of a CSPNet design. Through the stacking of C3 blocks and CBS blocks, the backbone continuously learns the features of higher dimensions. At the deepest level of the network, YOLOv5 features an SPPF module, which can be regarded as an optimized and faster spatial pyramid pooling (SPP) operation. It converts a feature image of any size into a feature vector of a fixed size, so that the input size of the network no longer needs to be fixed, thus realizing the fusion of local and global features.

Our network also included the global attention mechanism (GAM) [62] module after the SPPF module. This kind of attention calculation deep in the network helped it understand the high-dimensional features and focus on the key objects and key features. The network could retain the channel and space information to enhance cross-dimensional interaction by adding the GAM at an appropriate location in the backbone network. An overview is shown in Figure 2.

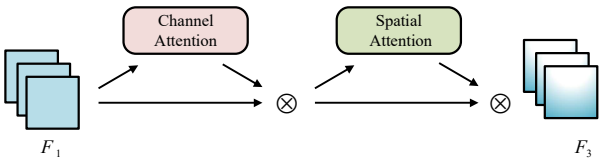

Figure 2. Overview of GAM; ⊗ stands for elementwise multiplication.

The goal of the GAM was to suppress information reduction and amplify global dimension interaction features. The combination and arrangement of the three channels and spatial attention submodules greatly affected the impact of the attention mechanism. CBAM [65] compared three different combinations and arrangements (sequential channel–space, sequential space–channel, and the parallel use of two attention modules) and, finally, proved that the sequential channel–spatial attention mechanism was the best choice. The GAM followed the arrangement of mechanisms in CBAM and redesigned its submodules. The GAM is expressed by Equations (1) and (2). For the input feature map $F_1 \in \mathbb{R}^{C\times H\times W}$, we could define the intermediate state F_2 and the output F_3 as:

$$F_2 = M_c(F_1) \otimes F_1 \tag{1}$$

$$F_3 = M_s(F_2) \otimes F_2 \tag{2}$$

The channel attention submodule applied 3D permutation to maintain the data in three dimensions. First, the GAM transformed the dimensions of the input feature from $C \times W \times H$ to $W \times H \times C$. It then magnified the cross-dimension channel–spatial dependencies with a two-layer multi-layer perceptron (MLP). An MLP is a type of encoder–decoder architecture with a compression rate r. It is simply composed of a linear layer used to reduce channels, a ReLU activation function, and another linear layer restored to the original number of channels. Rate r was set to 4 in our experiment. Finally, these processed features were subject to reverse permutation, and F_2 was generated using the sigmoid function. Figure 3 shows the channel attention submodule.

The GAM employed two convolutional layers to combine spatial information in the spatial attention submodule. The first 7×7 convolution layer reduced the number of input feature channels to $1/r$. In our experiment, r was set to 4. The second layer expanded the number of feature channels to the original value. Max pooling had an adverse effect, as it decreased the amount of information. The GAM eliminated pooling to maintain the feature maps more effectively. Consequently, the spatial attention module led to an expansion in the parameters in certain cases. Thus, we strongly recommend the use of group convolution instead of traditional convolution. This could reduce the number of parameters while hardly affecting the performance. The spatial attention submodule is illustrated in Figure 4.

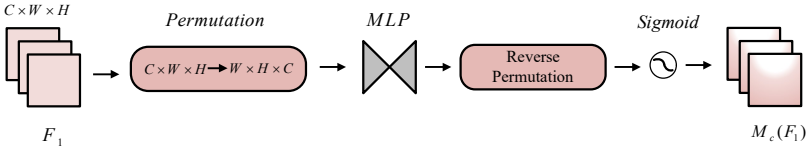

Figure 3. Schematic of channel attention mechanism.

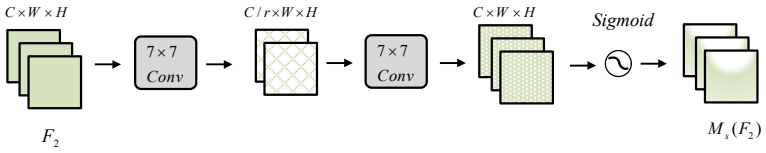

Figure 4. Schematic of spatial attention mechanism.

3.2. Neck

The basic component of the YOLOv5 neck is also the C3 module. Following the design of the PAFPN, the feature maps of different scales in the backbone were deeply fused. In YOLOv5, the neck eventually outputs feature maps of $80 \times 80 \times 256$, $40 \times 40 \times 512$, and $20 \times 20 \times 1024$ to correspond to target objects of a small, medium, and large scale.

We replaced the original C3 module with a fusion block, which enhanced the fusion of multi-scale features. Based on the design of efficient layer aggregation networks (ELANs) [66], fusion blocks can effectively implement rich gradient flow information at different levels. At the same time, they further improve performance by introducing reparameterized convolution modules.

The fusion block was based on DAMO-YOLO [20]. An overview of the fusion block is shown in Figure 5. Its main design goal was to upgrade CSPNet by incorporating a reparameterization mechanism and ELAN connections. CSPNet and the ELAN both further improved the performance of the model from the perspective of gradient optimization. Their design focuses on capturing as much rich gradient information as possible, which is crucial for the training of deep neural networks.

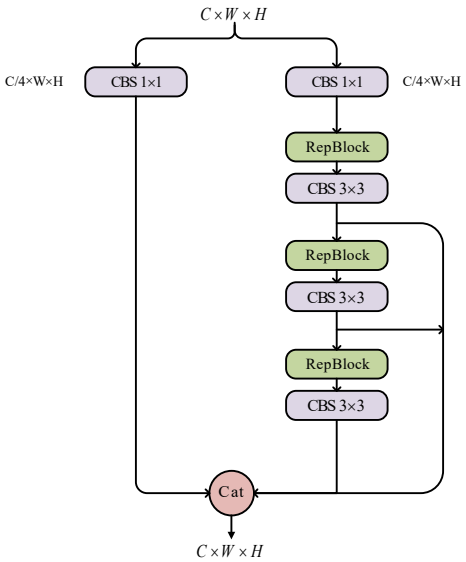

Figure 5. Schematic of fusion block; a schematic of RepBlock is shown in Figure 6.

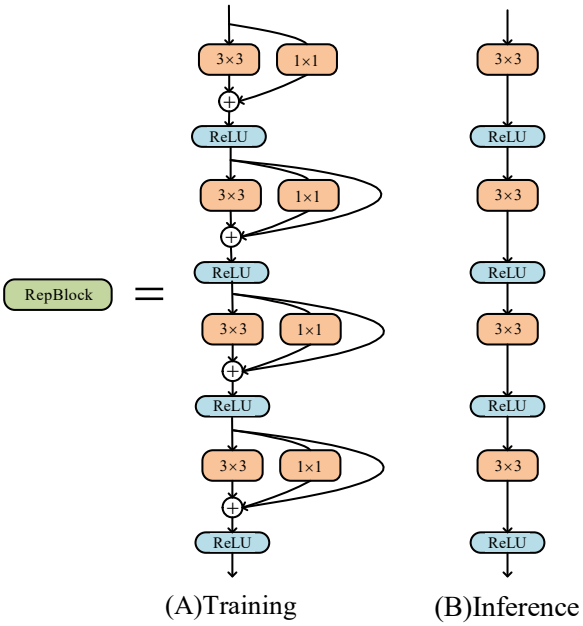

Figure 6. Schematic of RepConv block. This block had different structures for training and inferencing.

Connecting the output features of all previous layers as the input of the next layer and maximizing the number of branch paths obviously increased the amount of gradient information, which had considerable advantages. However, a simple connection can easily lead to the reuse of the same gradient information and high computing costs. CSPNet divides the basic input feature graph into two parts and then merges them through a cross-stage hierarchy structure. This operation divides the gradient flow and spreads it across different network paths. The gradient information spread presented substantial correlation differences, thus achieving a richer gradient combination and considerably reducing the computational load. ELANs utilize a combination of the shortest and longest gradient paths to improve the learning of neural networks.

Fusion blocks also pay attention to the segmentation and aggregation of gradient flow. In our model, the four-gradient path fusion block was used. A $C \times W \times H$ input feature was first divided into two branches, whose dimensions were reduced to $C/4 \times W \times H$ by a 1×1 convolution. One of the branches was directly connected to the final output. The other passed through the CBS and reparameterized the convolution block in sequence. Additionally, another three gradient paths split off from this branch to connect to the final output. Thus, the output feature still had the dimensions $C \times W \times H$.

The reparameterized convolution block (RepBlock) is shown in Figure 6. This was another key reason for the increased effectiveness of the fusion block. We switched out the original structure for a different structure by transforming the parameters into another set of parameters and coupling them with the new structure, thus altering the overall network architecture. RepVGG proposed restructuring the parameters to separate the multiple branches used for training and the single branch used for inference. During training, the RepConv block used a multi-branch convolution module, including 3×3 and 1×1 kernels and identity mapping. During inferencing, it adopted a plain architecture with only 3×3 convolution, which greatly reduced the number of parameters and improved the inference speed.

3.3. IoU Loss

The IoU loss was unable to correctly guide the network training without completely overlapping the prediction box and the ground truth box. In this study, we used the SCYLLA IoU (SIoU) loss [67] to replace the CIoU loss in order to accelerate the convergence and improve the accuracy.

The SIoU function is composed of four loss functions (angle cost, distance cost, shape cost, and IoU cost), which accurately measure the deviation between the target box and the true value. The SIoU loss is defined as:

$$L_{box} = 1 - IoU + \frac{\Delta + \Omega}{2} \tag{3}$$

where

$$IoU = \frac{|B \cap B^{GT}|}{|B \cup B^{GT}|} \tag{4}$$

See Figure 7 for an intuitive understanding of the IoU. We will introduce the remaining three loss functions and the definitions of Δ and Ω in detail in the following three sections.

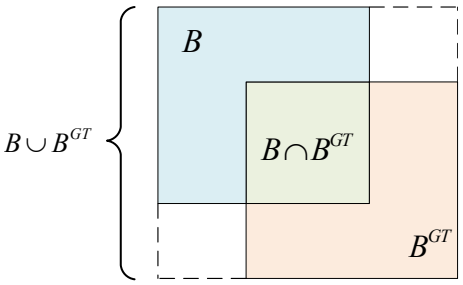

Figure 7. Schematic of IoU component definition.

3.3.1. Angle Cost

The loss function considers angles in order to reduce the number of variables in problems concerning distances. The model directs the prediction towards either the X or Y axis, whichever is closest, and then progresses along the applicable axis. In detail, it first tries to minimize α if $\alpha \leqslant \frac{\pi}{4}$; otherwise, it tries to minimize $\beta = \frac{\pi}{2} - \alpha$. The loss function is outlined and explained below (Figure 8):

$$\wedge = 1 - 2 * \sin^2\left(\arcsin(x) - \frac{\pi}{4}\right) \tag{5}$$

where

$$x = \frac{c_h}{\sigma} = \sin(\alpha) \tag{6}$$

$$\sigma = \sqrt{\left(b_{c_x}^{gt} - b_{c_x}\right)^2 + \left(b_{c_y}^{gt} - b_{c_y}\right)^2} \tag{7}$$

$$c_h = \max\left(b_{c_y}^{gt}, b_{c_y}\right) - \min\left(b_{c_y}^{gt}, b_{c_y}\right) \tag{8}$$

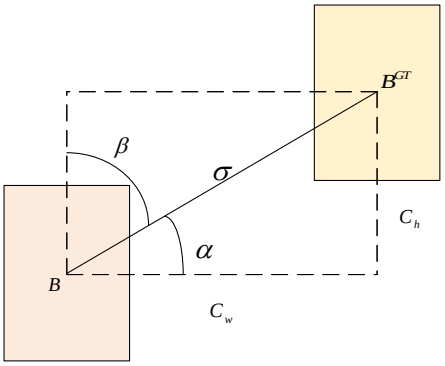

Figure 8. The scheme for angle cost.

3.3.2. Distance Cost

The distance cost is defined as follows (Figure 9):

$$\Delta = \sum_{t=x,y} \left(1 - e^{-\gamma \rho_t}\right) \tag{9}$$

where

$$\rho_x = \left(\frac{b_{c_x}^{gt} - b_{c_x}}{c_w}\right)^2, \rho_y = \left(\frac{b_{c_y}^{gt} - b_{c_y}}{c_h}\right)^2, \gamma = 2 - \Lambda \tag{10}$$

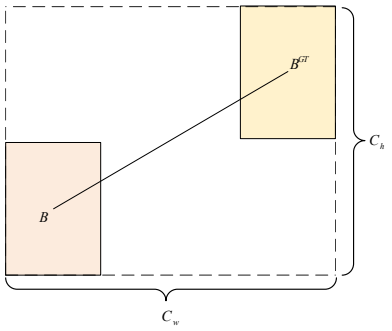

Figure 9. Scheme for the distance between the ground truth bounding box and the prediction box.

As α approaches zero, the impact of the distance cost is significantly diminished. As α approaches $\frac{\pi}{4}$, the magnitude of Δ's contribution increases. γ is given time priority over the distance value as the angle increases.

3.3.3. Shape Cost

The shape cost is defined as:

$$\Omega = \sum_{t=w,h} \left(1 - e^{-\omega_t}\right)^{\theta} \tag{11}$$

where

$$\omega_w = \frac{\left|w - w^{gt}\right|}{\max(w, w^{gt})}, \omega_h = \frac{\left|h - h^{gt}\right|}{\max(h, h^{gt})} \tag{12}$$

The magnitude of θ determines the cost of the shape, and each dataset has its own value. The cost of the shape is heavily dependent on this term, so it should be given due consideration.

4. Experiments

In this section, we first present the details of the implementation experiments and the evaluation metrics. Then, we describe the datasets used for evaluation. Finally, We carried out ablation experiments and demonstrate the superiority of our method by comparison with other methods.

4.1. Implementation Details

We built our network based on the widely used open-source project YOLOv5 [51] developed by Ultralytics. We implemented our network on Ubuntu 18.04, CUDA 10.2.89, pyTorch 1.10.0, and Python 3.7.13. The hardware environment and hyperparameters we used for training were different in the two datasets, and we present them separately in Sections 4.5 and 4.6.

4.2. Evaluation Metrics

In the field of target detection, precision, recall, and mean average precision (mAP) are the most-widely used indicators to measure the performance of target detection algorithms.

Precision and recall are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

The definitions of positive and negative cases are shown in Table 1. The AP is the measure of the area of the curve enclosed by the precision and recall. The mAP is obtained by comprehensively weighting the average of the AP detected by all categories when the IoU is set to a certain value.

Table 1. Positive and negative case judgment.

Reference \ Prediction	Positive	Negative
Positive	True positive (TP)	False negative (FN)
Negative	False positive (FP)	True negative (TN)

4.3. Datasets

4.3.1. URPC Dataset

The China Underwater Robot Professional Competition (URPC) is an annual competition that brings together experts and enthusiasts from various fields such as robotics, engineering, and marine science to showcase their innovations and advancements in underwater technology. The experimental dataset was obtained from the Target Recognition Group of the 2019 competition. The URPC2019 dataset [68] comprises 3765 training images and 942 validation images, encompassing five water target categories: echinus, starfish, holothurian, scallop, and waterweeds in Figure 10. In our experiment, we resized the images to 416 × 416 pixels.

(*a*) (*b*) (*c*) (*d*) (*e*)

Figure 10. URPC2019 dataset samples, namely (**a**) echinus, (**b**) starfish, (**c**) holothurian, (**d**) scallop, and (**e**) waterweeds

4.3.2. COCO Dataset

The MS COCO 2017 dataset [69] is a widely used benchmark dataset for object detection, segmentation, and captioning tasks. The images in the MS COCO 2017 dataset were collected from a wide range of sources and depict everyday scenes in natural contexts. It has 80 different categories including people, animals, vehicles, household objects, and other common items. For the object detection task, the dataset were split into training, validation, and test sets, with roughly 118 K, 5 K, and 40 K images, respectively. As the labels for this testing set were not public, we evaluated the metrics using the validation set.

4.4. Ablation Experiments

In this section, we conducted ablation experiments to verify the effectiveness and reliability of the improvements. In the YOLOv5 project, the network was partitioned into five sizes—N, S, M, L, and X—based on varying widths and depths. Our method adopted a similar design approach. Since underwater target detectors are frequently

deployed on mobile platforms with restricted computing power and storage, the S-size model, which exhibits exceptional performance while maintaining low system overhead, is highly recommended and practical. Therefore, for all the experiments on the URPC2019 dataset, we used the S-size model. The results can be seen in Table 2. The experimental results showed that every modification we made was successful. With the improvements made in the GAM, fusion block, and SIoU, the model achieved respective improvements of 0.9%, 0.8%, and 1.0%. Overall, the three improvements achieved a 4.1% higher mAP@0.5 score compared to the original YOLOv5_s model.

Table 2. Ablation experiments on URPC2019 with S-size model; "✓" indicates that we used this module.

Module			mAP@0.5 (%)
GAM	Fusion Block	SIoU	
			76.1
✓			77.0
	✓		76.9
		✓	77.1
✓		✓	78.1
✓	✓		77.8
	✓	✓	77.5
✓	✓	✓	80.2(+4.1)

4.5. Experiments on URPC2019

In this section, we employed our highly recommended underwater target detector, the S-size model, to conduct experiments on the URPC2019 dataset. Through comparative analysis with other advanced target detectors, we verified the effectiveness and superiority of our proposed method. The Experimental configuration is shown in Table 3.

Table 3. Experimental configuration when training on the URPC2019 dataset.

Parameter	Configuration
CPU	Intel(R) Xeon(R) Gold 5122@3.6 GHz
GPU	GeForce RTX 2080
Momentum	0.900
Weight decay	0.0005
Batch size	8
Learning rate	0.01
Epochs	100

We also present the PR curves in Figure 11. This demonstrated our model's ability to balance precision and recall, which are two critical performance metrics in target detection tasks. Our model had a higher area under the PR curve, indicating that it had higher precision and recall across all decision thresholds, which suggested that it was better at identifying positive cases while minimizing false positives.

In Table 4, we compare our algorithm with some advanced object detection algorithms with similar parameters. This included the latest one-stage detector YOLOv8 and two-stage detector Boosting RCNN [44]. Compared with YOLOv7 and YOLOv8, our method improved the mAP@0.5 by 1.8% and 2.3% respectively. It is worth noting that, when the reparameterized structure was fused, both the parameters and FLOPs of the model decreased, and the frames per second (FPS) increased by 25%. These performance improvements did not affect the accuracy. This demonstrated the unique advantage of the reparameterized structure in building lightweight networks. These experimental results demonstrated that our detector achieved a satisfactory level of accuracy with reasonable parameters and computational resources.

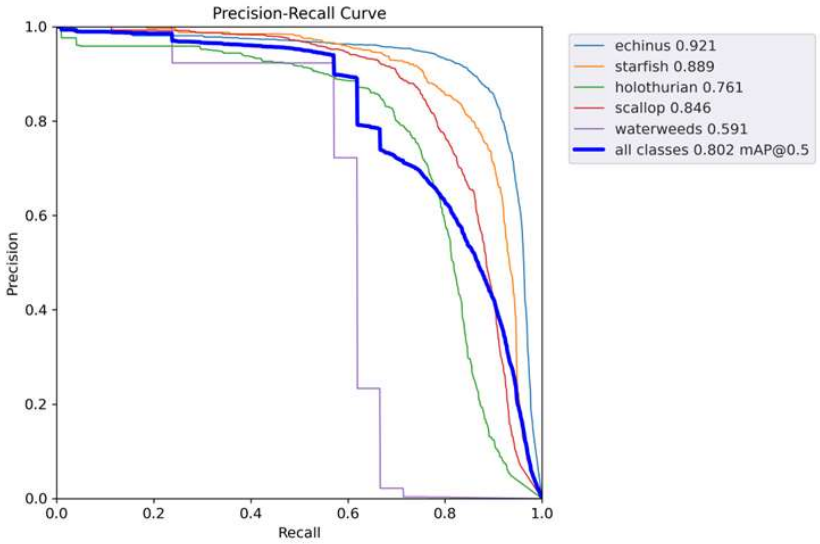

Figure 11. PR curves for URPC2019 dataset.

Table 4. Scores on URPC2019 compared with other methods. All algorithms had an input resolution of 640 × 640, and the image resolution of the dataset was 416 × 416; * indicates that the reparameterization structure of the model was fused. Bold font represents our best result.

Method	AP (%)					mAP@0.5(%)	Param.	FLOPs	Batch 1 FPS
	Echinus	Starfish	Holothurian	Scallop	Waterweeds				
Boosting RCNN [44]	89.2	86.7	72.2	76.4	26.6	70.2	45.9 M	77.6 G	22
YOLOv5_S	91.7	88.3	76.0	84.9	39.8	76.1	7.0 M	15.8 G	161
YOLOv7	91.9	89.6	78.3	86.5	45.7	78.4	36.5 M	103.2 G	75
YOLOv8_S	91.0	88.8	76.3	85.2	48.1	77.9	11.2 M	28.6 G	121
Our_S	92.1	88.9	76.2	84.7	59.1	80.2	14.0 M	28.0 G	83
Our_S *	**92.1**	**88.9**	**76.2**	**84.7**	**59.1**	**80.2**	**13.7 M**	**27.3 G**	**100**

Some intuitive detection diagrams are shown in Figure 12. We divided these pictures into two groups. The first group comprised images with small and blurred targets, captured in unfavorable shooting conditions. Despite the challenging environment, our model successfully completed the detection task without any missed or incorrect detections. The second group of images depicted scenes where targets appeared densely, and our model accurately located and classified all types of objects. In the aforementioned application scenarios, the original YOLOv5 model experienced disturbances from the environment, resulting in more missed detections. Conversely, our model demonstrated superior accuracy and robustness.

4.6. Experiments on MS COCO

We further tested our five size models on the MS COCO dataset to demonstrate that the proposed structure had good applicability. The hardware environment we used for training and inferencing was an Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz and Tesla V100 SXM2 32 GB. In order to ensure the stability of the training and facilitate the comparison, almost all hyperparameters were based on YOLOv5, except for the number of epochs. Under our experimental conditions, training the YOLOv5_s model for 300 epochs took over 72 h, which was highly impractical for us. Based on time-saving considerations, all the experiments in the MS COCO dataset were carried out under the same conditions with 100 epochs.

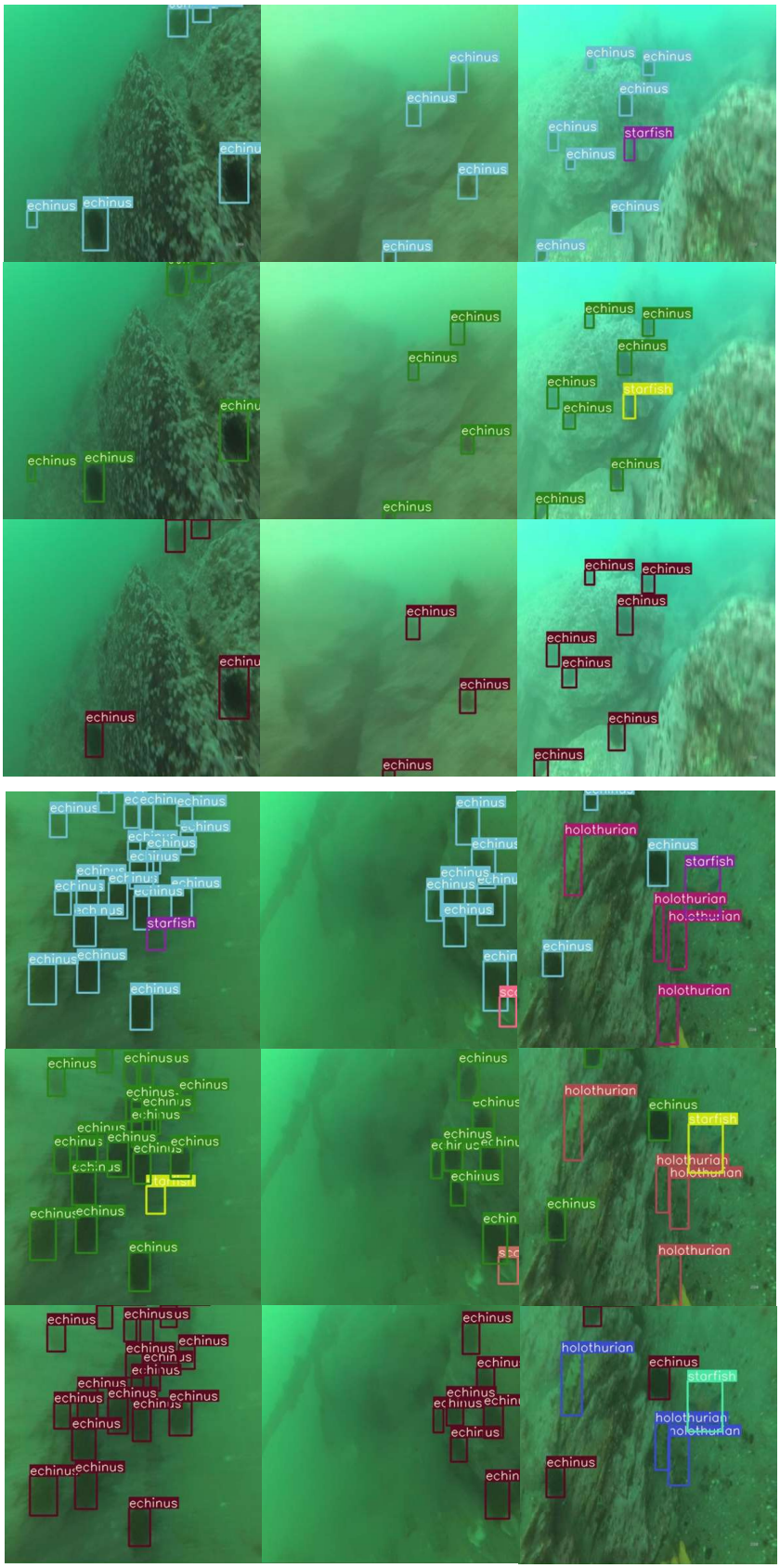

Figure 12. Detected images from URPC2019. The first row shows the ground truth; the second shows the results of our improved S-size model, and the third shows the results of YOLOv5_s.

The results in Table 5 show that our methods improved the mAP@0.5 by 7.1%, 2.9%, 4.0%, 3.0%, and 2.3% and the mAP@0.5:0.95 by 6.1%, 4.6%, 4.6%, 3.3%, and 2.7%, respectively, representing substantial improvements. The highest mAP@0.5:0.95 score obtained by our models was 51.0%. We used this score for the comparisons with other methods to show the precision level of our model.

With the COCO API, we were able to test the performance of the model for three types of targets: large, medium, and small. The improvements in small object detection were noteworthy. For all five model sizes, our models achieved improvements ranging from 1.9% to 3.2% compared with the originals.

Table 5. Experiments on MS COCO; training set was train2017, and test set was val2017. Bold font represents our model.

Method	mAP@0.5 (%)	mAP@0.75 (%)	mAP@0.5:0.95 (%)	mAP@0.5:0.95 (%)	mAP@0.5:0.95 (%)	mAP@0.5:0.95 (%)
				AP_S	AP_M	AP_L
YOLOv5_N	43.5	27.2	26.3	13.4	30	33.9
Our_N	**50.6 (+7.1)**	**35.2**	**33.3 (+6.1)**	**16.6 (+3.2)**	**36.6**	**44.7**
YOLOv5_S	56.9	39.5	37.0	21.3	41.9	47.8
Our_S	**59.8 (+2.9)**	**44.7**	**41.6 (+4.6)**	**23.6 (+2.3)**	**45.5**	**55.7**
YOLOv5_M	61.2	45.5	42.2	26.5	47.2	53.9
Our_M	**65.2 (+4.0)**	**50.6**	**46.8 (+4.6)**	**29.0 (+2.5)**	**51.4**	**60.9**
YOLOv5_L	65.1	50.2	46.2	30.6	51.3	58.9
Our_L	**68.1 (+3.0)**	**53.6**	**49.5 (+3.3)**	**32.5 (+1.9)**	**54.3**	**63.2**
YOLOv5_X	67.0	52.3	48.3	32.5	53.3	61.0
Our_X	**69.3 (+2.3)**	**55.1**	**51.0 (+2.7)**	**34.6 (+2.1)**	**55.8**	**64.5**

We compared some target detection algorithms, considering both CNN-based and Transformer-based models. Representative CNN-based models included RF-Next and YOLOR. The object detectors based on Transformers included the DETR series and ViDT Swin. The test results are shown in Table 6. The results showed that our model achieved the highest mAP@0.5:0.95. This proved that our improvement had good generalization performance. It not only had good accuracy in underwater target detection, but also was well applied to land general target detection.

Table 6. Scores on MS COCO compared with other methods. Bold font represents our model.

Method	Test Data	mAP@0.5:0.95 (%)
Sparse-DETR [70]	COCO val2017	49.3
DETR-DC5 [71]	COCO val2017	43.3
YOLOR-CSP [72]	COCO val2017	50.8
ViDT Swin-base [73]	COCO val2017	49.2
SQR-Adamixer-R101 [74]	COCO val2017	49.8
RF-ConvNeXt-T Cascade RCNN [75]	COCO val2017	50.9
Our_X	COCO val2017	**51.0**

5. Conclusions

In this paper, we proposed an improved YOLOv5 underwater object detection method. By introducing an attention mechanism, a multi-branch reparameterized structure, and a different loss function, the proposed method achieved higher accuracy in experiments on the URPC2019 dataset compared with the most-advanced algorithm of the YOLO series with a smaller number of parameters and calculation, proving its superior performance. For land target detection under better hardware conditions, we conducted further testing on the MS COCO dataset using our five models of varying depths and widths. Our experimental results demonstrated that our enhancement continued to yield positive outcomes.

However, we also faced some problems. With the incorporation of attention modules and reparameterization modules, in particular the introduction of additional convolutional and skip connection structures within the reparameterization module, the training time of the model was extended. In our training setup, the training durations per epoch for YOLOv5, YOLOv7, YOLOv8, and our model were 56 s, 51 s, 181 s, and 140 s, respectively. We hope that, in the future, we can further reduce the training time to accelerate the deployment process.

We should be cautious about the changes to a lightweight underwater detector. Fast inferencing, low overhead, and high accuracy are always contradictory. Balancing the relationship between them requires careful adjustment. In the design of future underwater target detectors, we should first choose technologies that integrate low computing and memory costs. The attention mechanism is an effective means to improve the detection ability of underwater targets, but it will cause additional burden on the training and inference ends. By contrast, using the SIoU is a less-expensive operation and has also been proven to be effective. It is ideal to use the reparameterized module to reconstruct the network. The module almost only increases the training time. The single-channel structure similar to VGG after fusion also makes it more hardware-friendly. If its training structure can be redesigned to use gradient information more effectively, this will greatly improve the performance.

6. Discussion

Our model was based on YOLOv5. Considering that YOLOv8 has a similar architecture to YOLOv5, porting the proposed method to YOLOv8 would be quite feasible. The improved YOLOv8 may have higher accuracy than our current model, but it will also face an increase in parameter and computational complexity. Meanwhile, as YOLOv8 does not have an advantage in inference speed compared to YOLOv5, if our method is directly extended to YOLOv8, it is likely to lead to a further decrease in the FPS. This may pose a challenge under the growing demand for real-time performance. All of this needs to be verified by our future experiments.

Author Contributions: Conceptualization, Y.Z., J.Z. (Jian Zhang) and H.J.; methodology, Y.Z. and J.Z. (Jian Zhang); software, H.C., J.Z. (Jian Zhang), K.Z. and X.Y.; validation, H.C., X.Y., J.Z. (Jinshuai Zhang), and K.Z.; formal analysis, J.Z. (Jian Zhang) and H.C.; investigation, J.Z. (Jian Zhang), H.C., and X.Y.; resources, Y.Z.; data curation, J.Z. (Jinshuai Zhang), B.S. and K.Z.; writing—original draft preparation, J.Z. (Jian Zhang) and H.C.; writing—review and editing, Y.Z.; visualization, X.Y., B.S. and K.Z.; supervision, Y.Z. and H.J.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Research and Development Project of Hainan Province, Grant No. ZDYF2019024, and the Hainan Provincial Natural Science Foundation of China, Grant Number 620QN236.

Data Availability Statement: Our code, model and dataset can be obtained from https://github.com/jojo-spirit/Improved-YOLOv5-Underwater-Detector.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Park, J.; Kim, J. Robust Underwater Localization Using Acoustic Image Alignment for Autonomous Intervention Systems. *IEEE Access* **2022**, *10*, 58447–58457. [CrossRef]
2. Almanza-Medina, J.E.; Henson, B.; Shen, L.; Zakharov, Y. Motion Estimation of Underwater Platforms Using Impulse Responses From the Seafloor. *IEEE Access* **2022**, *10*, 127047–127060. [CrossRef]
3. Baweja, P.S.; Maurya, P. Acoustics Based Docking for a Coral Reef Monitoring Robot (C-Bot). In Proceedings of the OCEANS 2022, OCEANS-IEEE, OCEANS Conference, Chennai, India, 21–24 February 2022. [CrossRef]
4. Zhao, Y.; Zhang, F.; Li, D.; Jin, B.; Lin, R.; Zhang, Z. Research on AUV terminal electromagnetic positioning system based on two coils. In Proceedings of the 2022 OCEANS Hampton Roads, 2022, OCEANS-IEEE, OCEANS Hampton Roads Conference, Hampton Roads, VA, USA, 17–20 October 2022. [CrossRef]

5. Lin, R.; Zhao, Y.; Li, D.; Lin, M.; Yang, C. Underwater Electromagnetic Guidance Based on the Magnetic Dipole Model Applied in AUV Terminal Docking. *J. Mar. Sci. Eng.* **2022**, *10*, 995. [CrossRef]
6. Huang, M.; Ye, J.; Zhu, S.; Chen, Y.; Wu, Y.; Wu, D.; Feng, S.; Shu, F. An Underwater Image Color Correction Algorithm Based on Underwater Scene Prior and Residual Network. In Proceedings of the Artificial Intelligence and Security: 8th International Conference, ICAIS 2022, Qinghai, China, 15–20 July 2022; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2022; pp. 129–139.
7. Yin, M.; Du, X.; Liu, W.; Yu, L.; Xing, Y. Multi-scale Fusion Algorithm for Underwater Image Enhancement based on Color Preservation. *IEEE Sens. J.* **2023**, *23*, 7728–7740. [CrossRef]
8. Tao, Y.; Dong, L.; Xu, L.; Chen, G.; Xu, W. An effective and robust underwater image enhancement method based on color correction and artificial multi-exposure fusion. *Multimed. Tools Appl.* **2023**, 1–21. . [CrossRef]
9. Yin, S.; Hu, S.; Wang, Y.; Wang, W.; Li, C.; Yang, Y.H. Degradation-aware and color-corrected network for underwater image enhancement. *Knowl.-Based Syst.* **2022**, *258*, 109997. [CrossRef]
10. Pipara, A.; Oza, U.; Mandal, S. Underwater Image Color Correction Using Ensemble Colorization Network. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW 2021), Montreal, BC, Canada, 11–17 October 2021; pp. 2011–2020. [CrossRef]
11. Xie, Y.; Yu, Z.; Yu, X.; Zheng, B. Lighting the darkness in the sea: A deep learning model for underwater image enhancement. *Front. Mar. Sci.* **2022**, *9*, 1470. [CrossRef]
12. Xu, S.; Zhang, J.; Bo, L.; Li, H.; Zhang, H.; Zhong, Z.; Yuan, D. Retinex based Underwater Image Enhancement using Attenuation Compensated Color Balance and Gamma Correction. In Proceedings of the International Symposium on Artificial Intelligence and Robotics 2021, Fukuoka, Japan, 21–27 August 2021; Volume 11884. [CrossRef]
13. Luchman, S.; Viriri, S. Underwater Image Enhancement Using Adaptive Algorithms. In Proceedings of the Progress in Artificial Intelligence and Pattern Recognition: 7th International Workshop on Artificial Intelligence and Pattern Recognition (IWAIPR), Havana, Cuba, 5–7 October 2021; Volume 13055, pp. 316–326. [CrossRef]
14. Fu, X.; Ding, X.; Liang, Z.; Wang, Y. Jointly adversarial networks for wavelength compensation and dehazing of underwater images. *Multimed. Tools Appl.* **2023**, 1–25. [CrossRef]
15. Yu, H.; Li, X.; Feng, Y.; Han, S. Underwater vision enhancement based on GAN with dehazing evaluation. *Appl. Intell.* **2023**, *53*, 5664–5680. [CrossRef]
16. Yang, G.; Lee, J.; Kim, A.; Cho, Y. Sparse Depth-Guided Image Enhancement Using Incremental GP with Informative Point Selection. *Sensors* **2023**, *23*, 1212. [CrossRef]
17. Xiang, Y.; Ren, Q.; Chen, R.P. A neural network for underwater polarization dehazing imaging. In Proceedings of the Optoelectronic Imaging and Multimedia Technology VIII, Nantong, China, 10–12 October 2021; Volume 11897. [CrossRef]
18. Ren, Q.; Xiang, Y.; Wang, G.; Gao, J.; Wu, Y.; Chen, R.P. The underwater polarization dehazing imaging with a lightweight convolutional neural network. *Optik* **2022**, *251*, 168381. [CrossRef]
19. Ding, X.; Liang, Z.; Wang, Y.; Fu, X. Depth-aware total variation regularization for underwater image dehazing. *Signal Process.-Image Commun.* **2021**, *98*, 116408. [CrossRef]
20. Xu, X.; Jiang, Y.; Chen, W.; Huang, Y.; Zhang, Y.; Sun, X. DAMO-YOLO: A Report on Real-Time Object Detection Design. *arXiv* **2022**, arXiv:2211.15444.
21. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
22. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 658–666.
23. Zhang, Y.F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L.; Tan, T. Focal and efficient IOU loss for accurate bounding box regression. *Neurocomputing* **2022**, *506*, 146–157. [CrossRef]
24. Wang, X.; Song, J. ICIoU: Improved loss based on complete intersection over union for bounding box regression. *IEEE Access* **2021**, *9*, 105686–105695. [CrossRef]
25. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
27. Wang, C.Y.; Mark Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580.
28. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 9992–10002.
29. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.

30. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing:Berlin/Heidelberg, Germany, 2018; Volume 11218, pp. 122–138.
31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
33. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.C.; Tan, M.; Chu, G.; Vasudevan, V.; Zhu, Y.; Pang, R.; et al. Searching for MobileNetV3. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1314–1324.
34. Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; Sun, J. RepVGG: Making VGG-style ConvNets Great Again. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 13728–13737.
35. Xu, F.; Wang, H.; Peng, J.; Fu, X. Scale-aware feature pyramid architecture for marine object detection. *Neural Comput. Appl.* **2021**, *33*, 3637–3653. [CrossRef]
36. Lin, T.Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
37. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
38. Zhu, L.; Deng, Z.; Hu, X.; Fu, C.W.; Xu, X.; Qin, J.; Heng, P.A. Bidirectional Feature Pyramid Network with Recurrent Attention Residual Modules for Shadow Detection. In Proceedings of the European Conference on Computer Vision (ECCV) 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; Volume 11210, pp. 122–137.
39. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *arXiv* **2019**, arXiv:1911.09516.
40. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [CrossRef]
41. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
42. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*; Cortes, C.; Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Montreal, QC, Canada, 2015; Volume 28.
43. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
44. Song, P.; Li, P.; Dai, L.; Wang, T.; Chen, Z. Boosting RCNN: Reweighting RCNN samples by RPN's error for underwater object detection. *Neurocomputing* **2023**, *530*, 150–164. [CrossRef]
45. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
46. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
47. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
48. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
49. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
50. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
51. Jocher, G. YOLOv5 by Ultralytics. 2022. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 September 2022).
52. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
53. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
54. ultralytics. Ultralytics YOLOv8. Available online: https://github.com//ultralytics//ultralytics// (accessed on 25 May 2023).
55. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
56. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global Second-Order Pooling Convolutional Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.

57. Lee, H.; Kim, H.E.; Nam, H. SRM: A Style-Based Recalibration Module for Convolutional Neural Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1854–1862.
58. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.
59. Fang, P.; Zhou, J.; Roy, S.; Petersson, L.; Harandi, M. Bilinear Attention Networks for Person Retrieval. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8029–8038.
60. Mnih, V.; Heess, N.; Graves, A.; kavukcuoglu, k. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Montreal, QC, Canada, 2014; Volume 27.
61. Ba, J.; Mnih, V.; Kavukcuoglu, K. Multiple object recognition with visual attention. *arXiv* **2014**, arXiv:1412.7755.
62. Liu, Y.; Shao, Z.; Hoffmann, N. Global Attention Mechanism: Retain Information to Enhance Channel-Spatial Interactions. *arXiv* **2021**, arXiv:2112.05561.
63. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. Unitbox: An advanced object detection network. In Proceedings of the 24th ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520.
64. Singh, S. YOLO-v4 Object Detector. 2013–2016. Available online: https://reckoning.dev/blog/yolo-v4// (accessed on 1 December 2022).
65. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; Volume 11211, pp. 3–19.
66. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800.
67. Gevorgyan, Z. SIoU loss: More powerful learning for bounding box regression. *arXiv* **2022**, arXiv:2205.12740.
68. Liu, H.; Song, P.; Ding, R. Towards domain generalization in underwater object detection. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), online, 25–29 October 2020; pp. 1971–1975.
69. Lin, T.; Maire, M.; Belongie, S.J.; Bourdev, L.D.; Girshick, R.B.; Hays, J.; Perona, P.; Ramanan, D.; Doll'ar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. *arXiv* **2014**, arXiv:1405.0312.
70. Roh, B.; Shin, J.; Shin, W.; Kim, S. Sparse detr: Efficient end-to-end object detection with learnable sparsity. *arXiv* **2021**, arXiv:2111.14330.
71. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
72. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206.
73. Song, H.; Sun, D.; Chun, S.; Jampani, V.; Han, D.; Heo, B.; Kim, W.; Yang, M.H. Vidt: An efficient and effective fully transformer-based object detector. *arXiv* **2021**, arXiv:2110.03921.
74. Chen, F.; Zhang, H.; Hu, K.; Huang, Y.k.; Zhu, C.; Savvides, M. Enhanced Training of Query-Based Object Detection via Selective Query Recollection. *arXiv* **2022**, arXiv:2212.07593.
75. Gao, S.; Li, Z.Y.; Han, Q.; Cheng, M.M.; Wang, L. RF-Next: Efficient receptive field search for convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2984–3002. [CrossRef]