


Article

Lightweight Underwater Target Detection Algorithm Based on YOLOv8n

Dengke Song and Hua Huo * 

Information Engineering College, Henan University of Science and Technology, Luoyang 471000, China; 220320040389@stu.haust.edu.cn

* Correspondence: pacific_huo@126.com

Abstract: To address the challenges in underwater target detection, such as complex environments, image blurring, and high model parameter counts and computational complexity, an improved lightweight detection algorithm, RDL-YOLO, is proposed. This algorithm incorporates multiple optimizations based on the YOLOv8n model. The introduction of the RFACnv module optimizes the backbone network, enhancing feature extraction capabilities under complex backgrounds. The DySample dynamic upsampling module is used to effectively improve the model's ability to capture edge information. A lightweight detection head based on shared convolutions is designed to achieve model lightweighting. The combination of the normalized wasserstein distance (NWD) loss function and CIoU loss improves the detection accuracy for small targets. Experimental results on the UPRC (Underwater Robot Prototype Competition) and RUOD (Real-World Underwater Object Detection) datasets show that the improved algorithm achieves an average precision (mAP) increase of 1.4% and 1.0%, respectively, while reducing parameter count and computational complexity by 19.3% and 14.8%. Compared to other state-of-the-art underwater target detection algorithms, the proposed RDL-YOLO not only improves detection accuracy but also achieves model lightweighting, demonstrating superior applicability in resource-constrained underwater environments.

Keywords: underwater target detection; YOLOv8; RFACnv; DySample



Received: 17 April 2025

Revised: 23 April 2025

Accepted: 24 April 2025

Published: 28 April 2025

Citation: Song, D.; Huo, H. Lightweight Underwater Target Detection Algorithm Based on YOLOv8n. *Electronics* **2025**, *14*, 1810. <https://doi.org/10.3390/electronics14091810>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Underwater target detection is a critical task in computer vision and a key research topic in image processing, primarily involving object recognition and localization in complex underwater environments [1]. However, the underwater environment is highly challenging, with underwater images often suffering from severe noise, poor visibility, blurred edges, and low contrast, which significantly increase detection difficulty [2–4].

In recent years, with the progress in deep learning, many breakthroughs in target detection in underwater environments have been made. The most mainstream target detection algorithms are now primarily classified into two types as follows [5]: two-stage target detection algorithms and single-stage target detection algorithms. Two-stage target detection algorithms, exemplified by the Fast R-CNN [6] and the Faster R-CNN [7], first handle the input image with a Region Proposal Network (RPN) to produce a sequence of candidate regions that likely contain targets and then utilize a Convolutional Neural Network (CNN) to further process the candidate regions. Since they require two stages of computation, their speed of processing the target is low. Single-stage target detection algorithms predict the target category and location directly on the map of features, bypassing the step of candidate region generation, thus greatly accelerating the process of detection. The most

mainstream single-stage target detection algorithms are the YOLO [8] (You Only Look Once) series and the SSD [9] (Single Shot MultiBox Detector). Of the two, the YOLO series of algorithms has been widely employed in underwater target detection since they can maintain high speed in the process of detection while guaranteeing accuracy. Cao et al. [10] proposed the BG-YOLO method, which enhances detection performance by constructing a parallel structure with an enhanced branch and detection branch and introducing a feature guidance module between them. Liu et al. [11] introduced the BiFormer module based on YOLOv7, utilizing a dual-layer routing attention (BRA) mechanism to focus on target edge and texture features, effectively solving the target blurring problem. Zhou et al. [12] designed YOLOv9s-SD based on YOLOv9s, integrating a simple enhancement attention module (SME) to strengthen attention to target features while employing the WIoU v3 loss function to increase the accuracy of small target localization. By implementing a lightweight C2f module, constructing a fast feature pyramid structure, and introducing a lightweight FasterNet backbone, Guo et al. [13] successfully achieved the enhancement of underwater target recognition in real time with good accuracy. Liu et al. [14] embedded a Transformer self-attention module into the backbone network of YOLOv5s and added a Coordinate Attention (CA) module in the neck network, which was designed to improve feature extraction capabilities, achieving better detection performance. Zhang et al. [15] presented an enhanced network based on YOLOv8, improving underwater target detection accuracy by introducing the FasterNet-T0 backbone, adding small target prediction heads, adjusting the number of feature map channels, and integrating deformable convolution and CA mechanisms in the Neck section. Wu et al. [16] proposed the SVGS-DSGAT model, which integrates GraphSAGE and the DSGAT [17] (Dynamic Structure Graph Attention) module to enhance underwater object detection performance. The DSGAT module employs a graph attention mechanism to dynamically adjust edge weights within the graph structure, thereby improving the model's ability to recognize objects in complex underwater environments.

Although the aforementioned methods have achieved certain success in improving underwater target detection accuracy, their model complexity and computational resource requirements are relatively high, limiting their practical application in resource-constrained underwater environments. In underwater scenarios, devices often face limited power supply and low computational performance, making the demands for model lightweighting and efficiency particularly strict.

As a result, the focus of this work is in how to further improve the accuracy in detection while still keeping the model lightweight to satisfy the demands of underwater devices for both the lightweight model and high-quality detection performance. According to these challenges, a lightweight target detection algorithm, RDL-YOLO, is proposed, an upgraded version of the YOLOv8n with the following improvements:

1. The introduction of the Introduce RFACnv module optimizes the backbone network, effectively suppressing interference from complex underwater backgrounds and enhancing the capacity to extract features for underwater targets.
2. The DySample dynamic upsampler is adopted in the neck network to effectively reduce the loss of edge detail information during upsampling.
3. A lightweight shared convolution detection head is designed, which preserves detection efficiency while significant reducing the number of parameters and computational complexity.
4. By combining the NWD and CIoU loss functions, the NWD-CIoU loss function is constructed, improving the accuracy of bounding box localization for small underwater targets.

2. Materials and Methods

2.1. Object Detection Algorithm

The YOLO (You Only Look Once) family of algorithms are real-time object detectors based on Convolutional Neural Networks (CNNs) that were first proposed by Joseph Redmon et al. in 2015. As an important member of the YOLO family, the YOLOv8 maintains the typical feature of real-time detection and can achieve high-quality detection even under low hardware specifications. Hence, the model is very suitable for low-resource environments, such as in underwater environments. The C2f module is used in the backbone network in place of the old C3 module, and the Bottleneck Block and Spatial Pyramid Pooling-Fast (SPPF) module are used to improve the function of extracting features while decreasing the complexity of the computation.

The neck network adopts a PAN-FAN structure, utilizing the Path Aggregation Network (PAN) [18] and Feature Aggregation Network (FAN) [19] to perform multi-scale feature fusion, improving the model's detection ability for targets of different scales.

In the head network, YOLOv8 has a decoupled head structure, partitioning the classification and regression tasks, resulting in increased accuracy in detection. YOLOv8 provides a range of model sizes such as N/S/M/L/X to align with different computational resources and application demands.

The model in this paper is based on YOLOv8n with several improvements. Compared to YOLOv8n, the model is optimized in the following four main areas: first, the backbone network is improved with receptive field attention convolutions; second, DySample replaces the network's upsampling technique; third, the detection head is redesigned based on shared convolution; and finally, the original loss function is enhanced by incorporating NWD. Figure 1 depicts the improved model's structure.

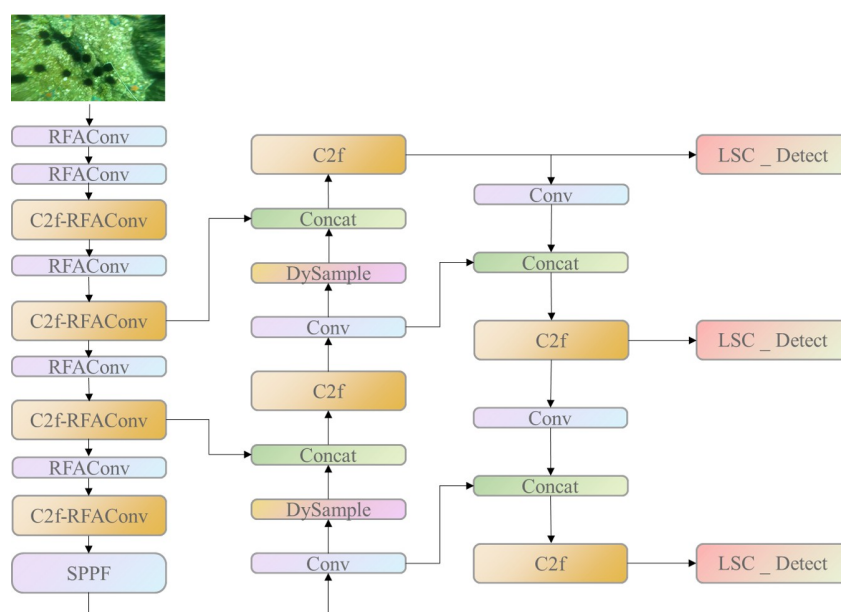


Figure 1. Network structure of RDL-YOLO.

2.2. Receptive-Field Attention Convolution

In reference [20], the C2f module was improved with a multi-scale convolutional EMC module, enhancing its ability to extract multi-scale feature information. However, it lacks attention to the correlation between preceding and succeeding modules. Due to the complex background and blurred details of underwater targets, the fixed convolution kernel size in the CBS module of YOLOv8 causes information loss of target features, particularly when processing small-scale targets where too much irrelevant information is

introduced. Additionally, when inputting feature information into the BottleNeck, further information loss occurs. To overcome these challenges, the RFAConv [21] module is proposed in this paper to improve the optimization of the backbone network's CBS and C2f modules. The novelty of RFAConv is in the integration of attention and receptive fields. The attention mechanism can inhibit interference from complicated background knowledge in underwater environments and enhance the highlighting of target features to make the model pay more attention to the most important parts and features of the target to precisely detect the target's boundary and location. Figure 2 depicts its structure.

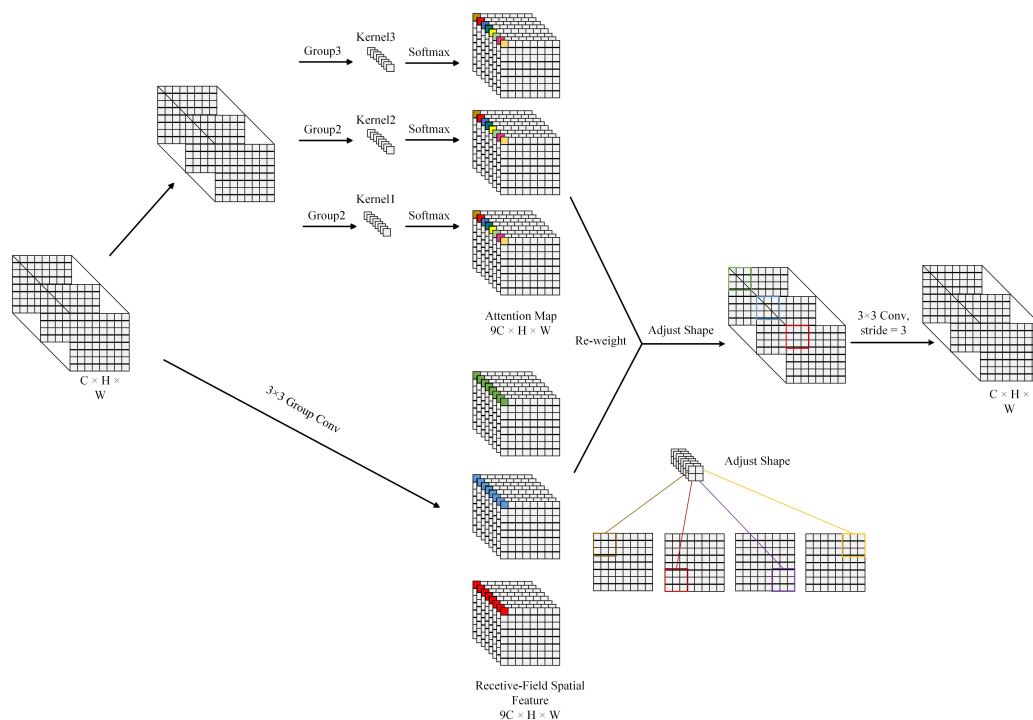


Figure 2. Process of spatial feature transformation of the receptive field.

The working process of RFAConv is as follows. To begin with, a “3 × 3 Group Conv” grouped convolution is applied to the input data with dimensions “C × H × W”. Grouped convolution reduces computation and parameter count, making the model more efficient. It also increases the diversity of convolution kernels, better capturing texture features of underwater targets with different shapes. Then, the grouped output is operated with the corresponding kernel and passed through the Softmax function to generate an attention map of dimension “9C × H × W”. The complex underwater environment, which includes suspended particles and uneven lighting, causes interference. The attention map can highlight the target area and suppress background noise and disturbances, focus on the key target features, and improve detection accuracy. Meanwhile, the input data are processed through grouped convolution to extract receptive field spatial features of the same dimension, and then, a re-weighting operation is performed. According to the attention map, target features are enhanced, irrelevant information is weakened, and target information in blurry images is reinforced. Finally, the re-weighted features are adjusted in shape, with part of the output returned to the “C × H × W” dimension through a “3 × 3 Conv, stride = 3” operation, while the other part continues to adjust and is split into sub-feature maps, allowing the model to better learn the target feature patterns and enhance underwater target detection accuracy. The calculation of RFAConv can be calculated as follows:

$$F = \text{Softmax}(g^{x1}(\text{AvgPool}(x))) \times \text{ReLU}(\text{Norm}(g^{xk}(x))) \quad (1)$$

To better extract features and suppress interference, replace the original convolution in C2f's BottleNeck with the RFACnv convolution, resulting in a more robust, accurate, and interference-resistant feature representation. The C2f-RFACnv structure is shown in Figure 3.

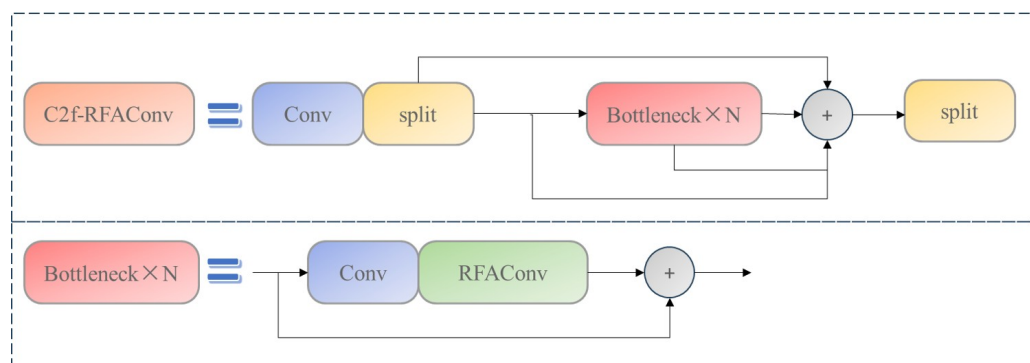


Figure 3. C2f-RFACnv structure diagram.

The improved BottleNeck first performs a 1×1 convolution operation to weight and combine features from different channels, enhancing both computational efficiency and model performance. Following this, the RFACnv convolution operation is applied to further suppress interference from the complex background information in underwater environments, improving the model's ability to extract features from underwater targets. In summary, the CBS and C2f modules improved by RFACnv can better extract features of underwater targets in complex backgrounds, strengthening the model's capacity to recognize underwater targets.

2.3. DySample Upsampling Module

As compared to aerial images, underwater images are prone to more noise, blurring, and color distortion. While upsampling, the nearest neighbor interpolation technique employed in YOLOv8 causes blurring or the loss of edge and detail information of the targets. To counteract the drawback, Xie et al. [22] replaced the default upsampling operation with Carafe upsampling, the model's receptive field, and improved feature extraction capacity. Carafe upsampling, however, is more susceptible to interference from noise and increased computational resource utilization. However, DySample [23] upsampling dynamically computes point sampling locations, allowing more precise recovery of the most important information while saving computational resources. Figure 4 depicts the working process.

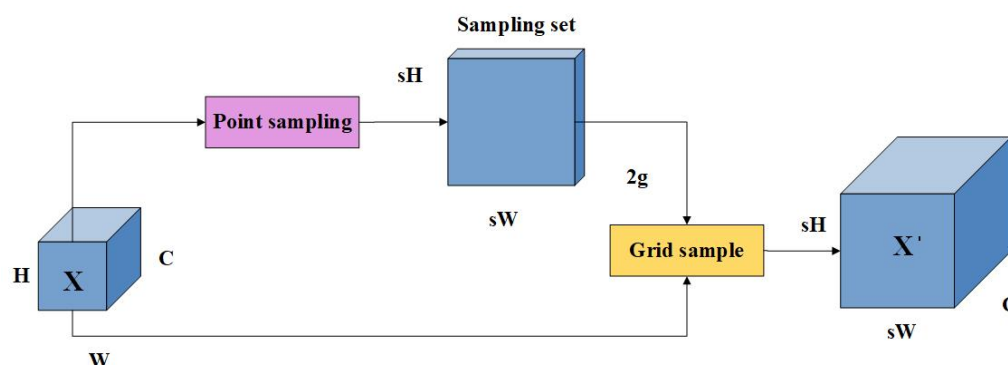


Figure 4. DySample module flowchart.

The feature map X is first input into a point sampling generator to generate a sampling set, which contains the sampling point information for subsequent upsampling operations. The grid

sampling module then takes as input the feature map X and the sampling set and grid samples X to obtain the upsampled feature map. This method can be represented as follows:

$$X' = \text{grid_sample}(X, S) \quad (2)$$

Figure 5 depicts the generation process for the sampling point set. First, the input feature X is processed through a linear layer to generate weights with specific spatial characteristics. Then, a 0.25-fold pixel rearrangement operation is performed, followed by the addition of the rearranged values to the original features, thus generating the sampling set. The pixel rearrangement ensures that the underwater feature map information is highly preserved, effectively avoiding the blurring and distortion issues often encountered with conventional upsampling methods.

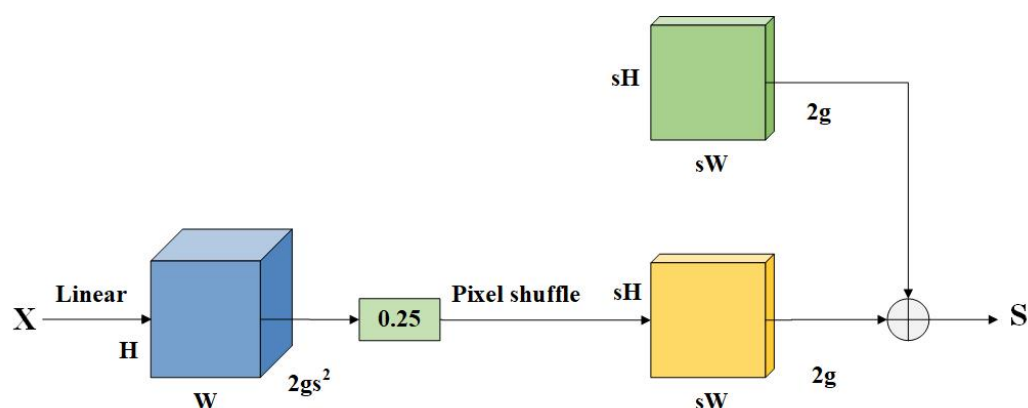


Figure 5. Dynamic point sampling set generation process.

DySample dynamically adjusts the sampling strategy, effectively reducing the loss of edge detail information during upsampling. This strategy ensures that the features in the edge areas are better preserved during image size restoration, thereby improving the network's ability to catch tiny details.

2.4. Shared Convolution Detection Head

YOLOv8's detection head adopts a decoupled head structure, consisting of three branches and multiple convolution layers, with each branch performing independent computations. This makes the model structure relatively complex, with high parameter count and computational requirements. In underwater target detection scenarios, detection devices typically have limited computational resources, so the optimization of the detection head is necessary. Based on the lightweight design principle, this paper proposes using shared convolutions to process inputs across layers. The multiple independent convolutions of the three detection heads are replaced by two DEConv [24] convolutions with shared weights. Since inconsistent input scales can affect subsequent shared convolutions, prior to entering the shared convolution, each layer's input is normalized using a 1×1 Conv_GN module [25]. This ensures that inputs of different scales are better adapted to the shared convolution processing, after which they are passed through the shared convolution. Finally, the feature maps are processed by two different convolution layers to calculate the bounding box regression and classification probabilities. The structure of LSC_Detect is shown in Figure 6.

Unlike Batch Normalization (BN), Conv_GN's normalization computation does not depend on batch size, making it more effective in cases with small batch sizes or varying batch sizes. This allows GN to maintain good normalization performance, contributing to improved model stability and generalization ability.

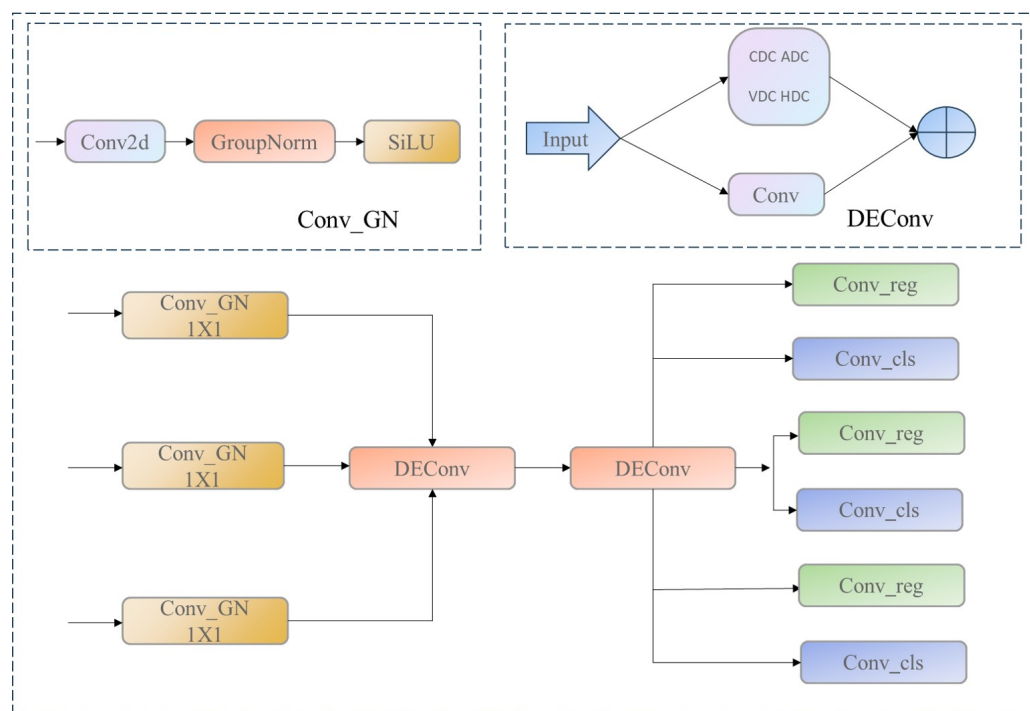


Figure 6. Structure of LSC_Detect.

DEConv (Detail-Enhanced Convolution) uses five types of parallel convolution layers to perform feature extraction. Among these, standard convolution (Conv) primarily collects image intensity data, i.e., capturing the actual brightness and color values of pixels. The central differential convolution (CDC) extracts pixel differences from the center point, the angular differential convolution (ADC) handles pixel variation in the corners of images, and horizontal differential convolution (HDC) and vertical differential convolution (VDC) deal with pixel variation in the horizontal and vertical orientations, respectively, to extract texture and edge details from the image in a holistic way. The differential convolution layers effectively extract the texture and the details of edges from an image by paying attention to pixel variation in various regions.

By combining the outputs of these parallel convolution layers, DEConv generates a feature map with more detailed and comprehensive information. The workflow of DEConv can be expressed in Equation (3) as follows:

$$F_{\text{out}} = \text{DEConv}(F_{\text{in}}) = \sum_{i=1}^5 F_{\text{in}} * K_i \quad (3)$$

where $K_{i=1:5}$ corresponds to CDC, ADC, HDC, VDC, and standard convolution; $*$ represents convolution; and \sum represents the summation operation.

2.5. Loss Function Improvement

In YOLOv8, the regression loss in the loss function is a combination of DFL (Distribution Focal Loss) and CloU (Complete Intersection over Union Loss) [26]. CloU mainly measures the overlap between the predicted and actual bounding boxes by considering the distance between their center points and the consistency of their aspect ratios. However, the intricacy of the underwater milieu, encompassing elements such as light refraction, scattering, and water turbidity, leads to a degradation of image quality, causing the target's boundaries to become blurry. This complicates the precise calculation of the position and aspect ratio of the expected and ground truth boxes. This is especially problematic for small underwater targets, which occupy very few pixels in the image and are prone to significant

boundary box localization errors. Since CIoU is sensitive to localization errors, this further exacerbates the issue, leading to a decrease in model performance. To address this, this paper introduces NWD (Normalized Wasserstein Distance) [27] to improve the CIoU loss function, proposing the NWD-CIoU loss. NWD uses the Wasserstein distance from optimal transport theory [28] to compute the distance between the predicted and actual bounding boxes, subsequently normalizing it to derive the normalized Wasserstein distance, which serves as a metric for the bounding box. NWD does not require considering whether the bounding boxes overlap and is impervious to variations in target scale, rendering it more appropriate for measuring the bounding box loss in small underwater target images. The specific calculation of NWD is shown in Equation (4) as follows:

$$L_{NWD} = 1 - \exp\left(-\frac{\sqrt{W^2(N_a, N_b)}}{C}\right) \quad (4)$$

where C is the number of categories in the dataset, $W^2(N_a, N_b)$ is the second-order Wasserstein distance; and N_a and N_b represent the Gaussian modeled distributions of bounding boxes A and B, respectively.

Considering that using NWD alone as the loss function would reduce the model's compatibility in handling different target scales, making it difficult to effectively deal with detection requirements for targets of varying sizes, this paper combines NWD and CIoU to construct the NWD-CIoU loss function for the model's localization loss. The formula for this combined loss is shown in Equation (5) as follows:

$$L_{NWD-CIoU} = \alpha L_{CIoU} + (1 - \alpha) L_{NWD} \quad (5)$$

where α is the weight assigned to the CIoU loss function. Through experimentation, weight α was determined to be 0.4.

3. Experimental Results and Analysis

3.1. Dataset Introduction

The URPC (Underwater Robot Professional Contest) series dataset is a high-quality underwater image dataset released by institutions such as the Ocean University of China since 2017. It is extensively utilized in underwater target detection, classification, image enhancement, and other research fields. The dataset utilized in the studies was created by integrating the URPC2020 and URPC2021 underwater optical image datasets, followed by the removal of duplicate images and similar images. Specifically, the Structural Similarity Index (SSIM) was used to assess the similarity between images, with a threshold of 0.8 set to identify and remove highly similar images, which typically result from continuous shooting. The processed dataset contains 7479 images, covering the following four categories: sea urchins, sea cucumbers, scallops, and starfish. These images showcase typical underwater environmental features, including color distortion, low contrast, uneven illumination, blurriness, and elevated noise levels. The dataset was randomly partitioned into training, validation, and test sets in a 7:1:2 ratio. The training set comprises 5235 photos, the validation set includes 747 images, and the test set contains 1497 images. Figure 7 displays many representative photos from the collection.

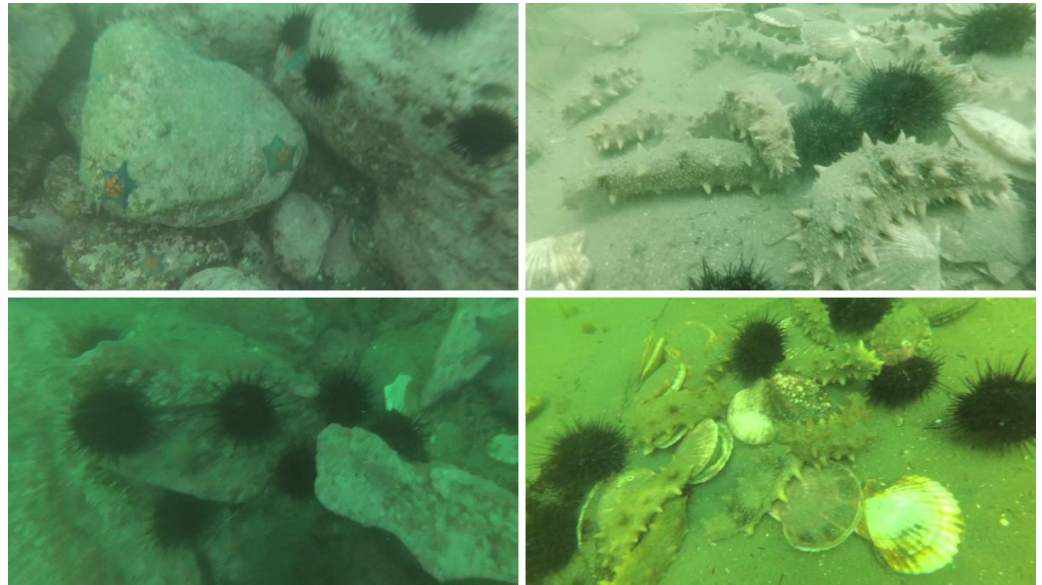


Figure 7. Sample images from the URPC dataset.

3.2. Experimental Environment

The studies were conducted utilizing the PyTorch 2.1.0 deep learning framework, version 2.1.0, with Python 3.10. The operating system employed was Ubuntu 22.04, and the hardware consisted of an Intel Core i5-12400F processor, an NVIDIA GeForce RTX 4090D graphics card with 24 GB of video RAM, and CUDA 12.1.

In the trials performed on the URPC dataset, the model training parameters were consistently established. The precise parameters are as follows: The image dimensions were configured to 640×640 , the batch size was established at 32, and the optimizer employed was the Stochastic Gradient Descent (SGD) technique, with a momentum of 0.937, a weight decay coefficient of 0.005, and an initial learning rate of 0.01. The model underwent training for 200 epochs.

3.3. Evaluation Metrics

To comprehensively assess the efficacy of the improved model, we selected Precision (P), Recall (R), and mean Average Precision (mAP) as the metrics to reflect the model's accuracy in the object detection task. Additionally, the Parameter count and computational complexity were employed as metrics for model lightweighting. The specific evaluation metrics are given in Equations (6)–(10).

$$P = \frac{TP}{TP + FP} \quad (6)$$

$$R = \frac{TP}{TP + FN} \quad (7)$$

$$AP = \int_0^1 P(R) dR \quad (8)$$

$$mAP = \frac{\sum_{i=1}^k AP_i}{k} \quad (9)$$

$$FLOPs = K^2 \times W \times H \times C_0 \times C_i \quad (10)$$

3.4. Ablation Experiment

Ablation tests were conducted using the UPRC dataset to evaluate the efficacy of the enhanced modules on YOLOv8n for underwater target detection. The experimental findings are presented in Figure 8 and Table 1.

The results in Table 1 show that in Experiment 2, after introducing RFACnv into the C2f module of the backbone network, mAP50 increased by 0.9%. In Experiment 3, when RFACnv was introduced into the CBS module of the backbone network, mAP50 increased by 0.7%. In Experiment 7, when both the CBS and C2f modules in the backbone network were enhanced with RFACnv, mAP50 improved by 1.1%, with a marginal increase in parameter count and computational complexity. In Experiment 4, substituting the original detection head with LSC_Detect resulted in a reduction in the model's parameter count by 0.65 M and its computational complexity by 1.6 G. In Experiment 6, after improving the loss function, mAP50 increased by 0.3%. The results of these experiments demonstrate the effectiveness of each of the improvement modules for underwater target detection. Experiments 7–10 implemented all enhancements to the YOLOv8n model, yielding a final mAP50 increase of 1.4%, alongside a decrease in parameter count and computational complexity by 19.3% and 14.8%, respectively. Figures 8 and 9 provide an intuitive depiction of the variations in different indicators throughout the ablation experiment. This signifies that the enhanced model not only augmented average precision but also reduced parameter count and computational complexity.

Table 1. Ablation study results. (✓ indicates that the improvement has taken effect).

Experiments	C2f-RFA	RFACnv	LSC_Detect	DySample	NWD-CIoU	mAP50/%	Params/M	FLOPs/G
1						82.5	3.01	8.1
2	✓					83.4	3.04	8.4
3		✓				83.2	3.03	8.4
4			✓			82.3	2.36	6.5
5				✓		82.9	3.01	8.1
6					✓	82.8	3.01	8.1
7	✓	✓				83.6	3.07	8.7
8	✓	✓	✓			83.3	2.43	6.9
9	✓	✓	✓	✓		83.7	2.43	6.9
10	✓	✓	✓	✓	✓	83.9	2.43	6.9

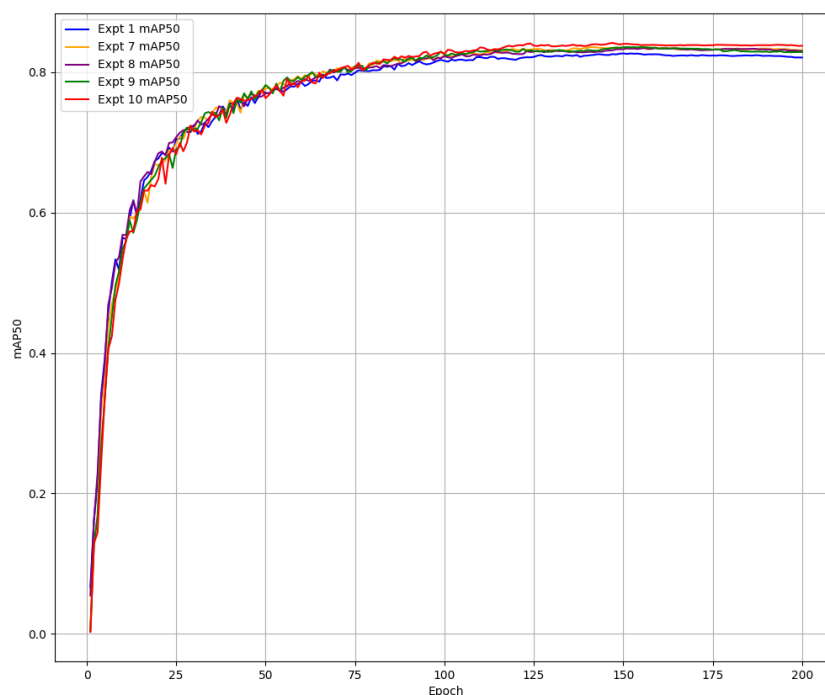


Figure 8. mAP50 changes curve.

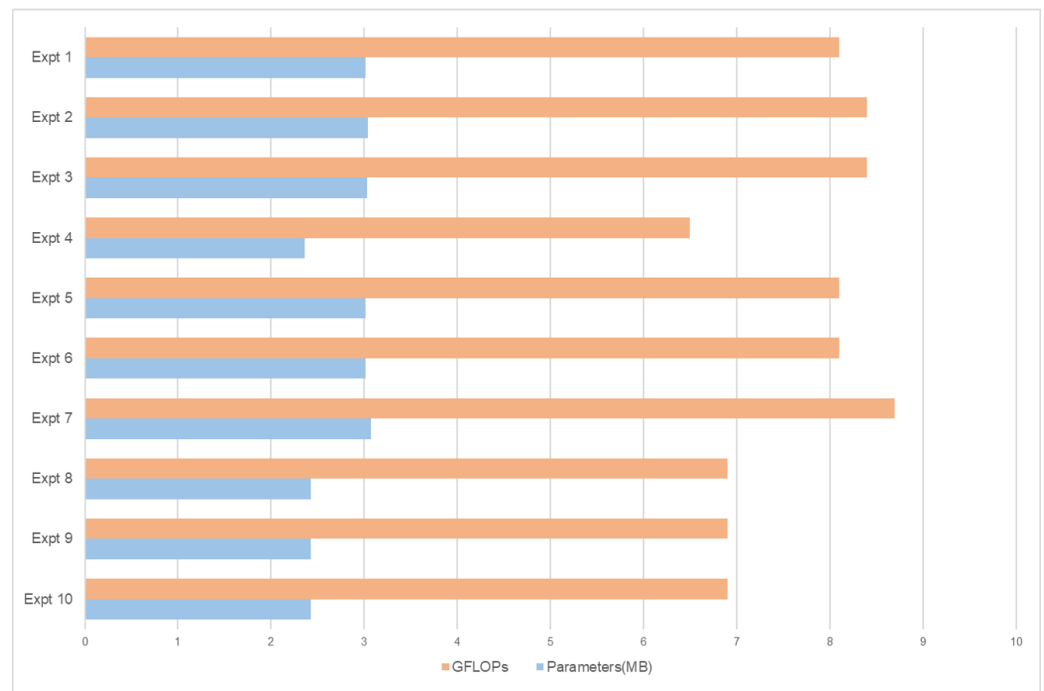


Figure 9. Comparison of parameters as well as computational effort.

3.5. Loss Function Weight Assignment Comparison

To explore the effectiveness of combining NWD with CIoU and the determination of weight α , comparison experiments were conducted on RDL-YOLO using different weight assignments for α . The experimental findings are presented in Table 2.

Table 2. Weight assignment comparison results.

Weight Assignment	Precision/%	Recall/%	mAP50/%
1.0	83.3	76.5	83.7
0.8	83.1	76.7	83.6
0.6	82.6	77.2	83.7
0.4	83.0	77.9	83.9
0.2	82.8	77.5	83.8
0	82.7	77.4	83.5

From the data in Table 2, it can be seen that as the value of α decreases, the detection accuracy of the model changes. When $\alpha = 0.4$, detection accuracy is highest. The experimental results indicate that introducing NWD allows the model to calculate the optimal solution for the similarity between the predicted and actual boxes, hence enhancing the model's accuracy in underwater target detection.

3.6. Comparison Experiment

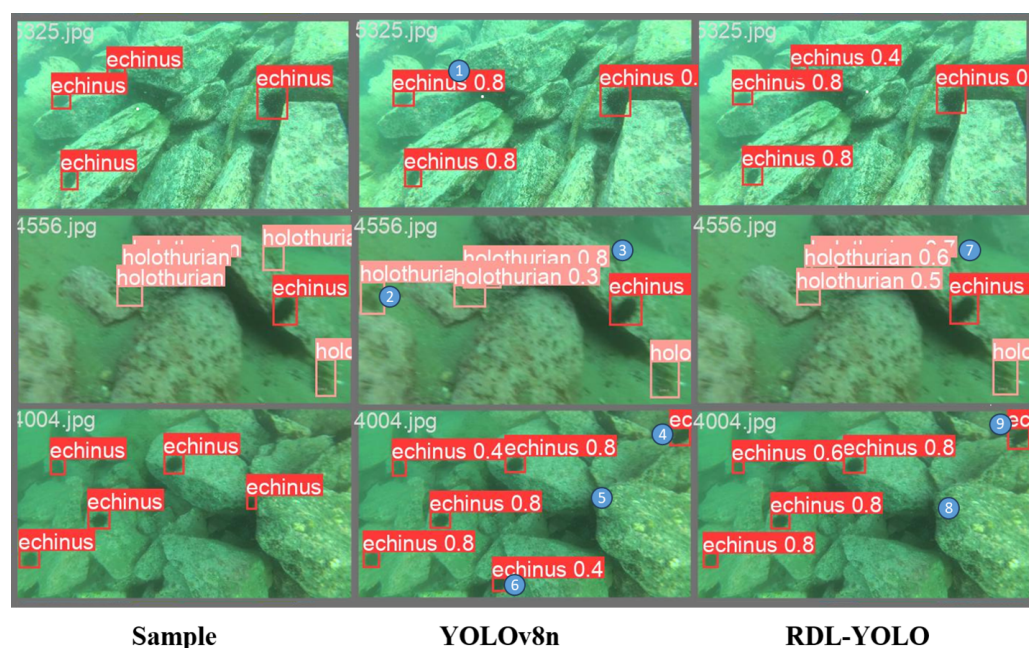
To assess the efficacy of the enhanced model for underwater target detection, comparison experiments were conducted with the mainstream general object detection models, the latest proposed underwater target detection approaches, and the most recently proposed underwater target detection models. The experimental findings are presented in Table 3.

Table 3. Comparison experiment results on the UPRC dataset.

Model	mAP50/%	Params/M	FLOPs/G
Faster R-CNN	75.2	41.14	63.3
YOLOv3-Tiny [29]	79.6	12.13	18.9
YOLOX-Tiny [30]	80.1	5.06	15.4
YOLOv5n	81.5	2.50	7.1
YOLOv8n	82.5	3.01	8.1
YOLOv10 [31]	81.6	2.69	8.2
YOLOv11n	82.0	2.60	6.5
reference [32]	84.0	19.0	6.5
reference [33]	82.0	13.7	27.3
reference [34]	83.3	3.10	8.0
reference [35]	83.6	2.55	7.5
RDL-YOLO	83.9	2.43	6.9

Analyzing Table 3 indicates that the proposed algorithm, RDL-YOLO, performs better compared to conventional model algorithms in that they offer greater detection accuracy while utilizing reduced computing complexity and fewer parameters. Compared to the baseline, the proposed model scores an 83.9% accuracy in detections, with an increase of 1.4 percentage points. The proposed model is also lighter in weight, thus appropriate for deployment in settings with restricted hardware. Compared to the most advanced underwater target detection models, the proposed model exhibits superior performance in both mAP and computational complexity. While the UODN has a higher average precision compared to the proposed model, the complexity of the proposed model is far less compared to that of UODN. As a whole, the RDL-YOLO offers an equilibrium of performance and complexity for the model, proceeding to be more appropriate in the case of underwater environments with restricted computational resources.

To more intuitively showcase and evaluate the performance differences between the RDL-YOLO algorithm and the YOLOv8 baseline algorithm in underwater target detection, Figure 10 provides a comparison of the detection outcomes of the two algorithms on identical images.

**Figure 10.** Detection results before and after model improvement on the UPRC dataset.

The figure illustrates that RDL-YOLO exhibits greater stability in target recognition compared to YOLOv8, offering more precise bounding box localization and higher confidence scores. Specifically, in the first image, YOLOv8 failed to detect the sea urchin at label 1; in the second image, it incorrectly identified the object at label 2 on the left side as a sea cucumber; and in the third image, it misclassified the object at label 6 at the bottom as a sea urchin. In contrast, the improved RDL-YOLO provided correct results in all these cases. However, both YOLOv8 and RDL-YOLO failed to detect the sea cucumber targets at labels 3 and 7 of the second image, as well as the sea urchin targets at labels 5 and 8 in the third image. This suggests that further improvements are needed in detection performance under low-light conditions.

3.7. Generalization Experiment

To enhance the assessment of the RDL-YOLO model's generalization ability, further verification using the RUOD dataset was undertaken. The RUOD dataset [36] is specially formulated for underwater detection tasks and covers a range of underwater detection difficulties. The dataset covers a broad target range that includes fish, divers, starfish, sea turtles, sea urchins, sea cucumbers, scallops, squid, and jellyfish with varying numbers of each type of target as indicated in the figure. The collection contains 14,000 pictures and 74,904 labeled items. The dataset is randomly partitioned into training, validation, and test sets in the ratio 8:1:1, comprising 11,200 images in the training set, 1400 in the validation set, and 1400 in the test set.

All models underwent training and testing under identical experimental settings, as specified in Section 3.6. The comprehensive experimental findings are presented in Table 4 and Figure 11.

Table 4. Comparison experiment results on the RUOD dataset.

Model	mAP50/%	Params/M	FLOPs/G
Faster R-CNN	65.2	41.14	63.3
YOLOv3-Tiny	80.6	12.13	18.9
YOLOX-Tiny	82.1	5.06	15.4
YOLOv5n	83.5	2.50	7.1
YOLOv8n	84.1	3.01	8.1
YOLOv10	83.7	2.69	8.2
YOLOv11n	84.0	2.60	6.5
RDL-YOLO	85.1	2.43	6.9

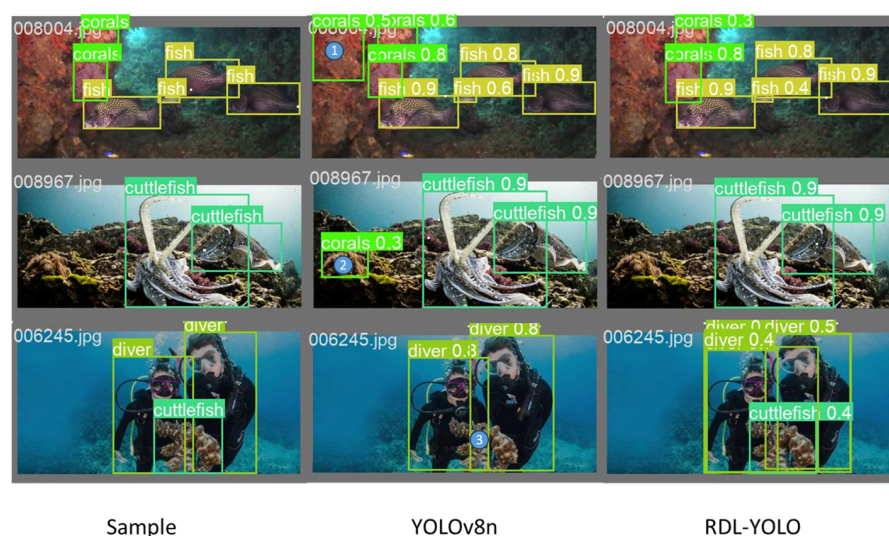


Figure 11. Detection results before and after model improvement on the RUOD dataset.

According to the table, RDL-YOLO achieves an mAP of 85.1% while maintaining a low parameter count and computational load, representing a 1.0% improvement over YOLOv8n. Furthermore, compared to YOLOv10n and YOLOv11n, it achieves mAP50 improvements of 1.4% and 1.1%, respectively. In Figure 11, RDL-YOLO demonstrates superior performance in complex backgrounds. YOLOv8n exhibits both missed and false detections, as follows: it fails to detect the cuttlefish at marker 3 and incorrectly identifies the background at markers 1 and 2 as corals. In contrast, the improved RDL-YOLO adapts better to complex scenarios, accurately localizing targets. These results indicate that RDL-YOLO enhances detection accuracy while maintaining a lightweight design, exhibiting greater stability in complex backgrounds, in particular.

4. Discussion

4.1. Findings

The experimental results show that the improved RDL-YOLO algorithm exhibits significant performance enhancement on the UPRC and RUOD datasets by introducing the RFACnv module, the DySample dynamic up-sampling operator, the NWD-CIoU loss function, and the lightweight detection header of LSC_Detect. Specifically, the model achieves a mean average precision (mAP) of 83.9% on the UPRC dataset, which is 1.4 percentage points higher than the original YOLOv8n, while the mAP on the RUOD dataset is 85.1%, which is 1.0 percentage point higher. At the same time, the algorithm remains lightweight, and the number of parameters and computational costs are kept at a low level, which can meet the practical application requirements of underwater resource-constrained scenarios.

4.2. Limitations and Future Works

Although the improved model has made significant progress in terms of detection accuracy and lightweighting, there are still limitations. As can be seen from the detection result images, RDL-YOLO still suffers from some omissions and misdetections, especially in the shadow part of the image where other objects are mistakenly detected as sea urchins. In addition, underwater images generally have quality problems. Although the goal of lightweight design was achieved, the performance of the model in real underwater deployment scenarios has not yet been verified.

In light of the attributes of underwater settings, forthcoming efforts will concentrate on the following three directions:

1. Combining lightweight image enhancement modules with the detector for joint training to improve detection accuracy under low-light and high-noise conditions.
2. Integrating optical and sonar image information to use multimodal data to enhance the model's robustness and enhance detection efficacy in intricate underwater settings.
3. Deploying and testing the model in real underwater environments to evaluate its performance and further optimize the model structure and parameter configuration.

5. Conclusions

In this work, the RDL-YOLO algorithm is proposed by enhancing the YOLOv8n model to fix issues associated with complex underwater environments, the blurring of images, and high model parameters and computational complexity. RFACnv is proposed to augment the backbone network, hence improving the model's capacity to extract features from complex backdrops. The DySample dynamic upsampling module is employed to efficiently enhance the model's proficiency in edge extraction. A lightweight shared convolution-based detection head is proposed to implement model lightweighting. Lastly, the integration of the NWD loss function with the Ciou loss function improves the model's precision in identifying small objects. Experimental results on the UPRC and RUOD

benchmarks indicate that the proposed approach boosts the mAP value by 1.4% and 1.0%, respectively, and decreases parameters and computational complexity by 19.3% and 14.8%, respectively. These illustrate that the model is more appropriate for deployment in constrained underwater scenarios.

Author Contributions: Conceptualization, D.S. and H.H.; methodology, D.S.; software, D.S.; validation, D.S. and H.H.; formal analysis, D.S.; investigation, D.S.; resources, H.H.; data curation, D.S.; writing—original draft preparation, D.S.; writing—review and editing, D.S.; visualization, D.S.; supervision, D.S.; project administration, H.H.; funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China (61672210), the Major Science and Technology Program of Henan Province (221100210500), and the Central Government Guiding Local Science and Technology Development Fund Program of Henan Province (Z20221343032).

Data Availability Statement: The data presented in this study are available upon request from the corresponding author.

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

References

- Joshi, R.; Usmani, K.; Krishnan, G.; Blackmon, F.; Javidi, B. Underwater object detection and temporal signal detection in turbid water using 3D-integral imaging and deep learning. *Opt. Express* **2024**, *32*, 1789–1801. [\[CrossRef\]](#) [\[PubMed\]](#)
- Xu, S.; Zhang, M.; Song, W.; Mei, H.; He, Q.; Liotta, A. A systematic review and analysis of deep learning-based underwater object detection. *Neurocomputing* **2023**, *527*, 204–232. [\[CrossRef\]](#)
- Shen, L.; Reda, M.; Zhang, X.; Zhao, Y.; Kong, S.G. Polarization-driven solution for mitigating scattering and uneven illumination in underwater imagery. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 4202615. [\[CrossRef\]](#)
- Yi, X.; Jiang, Q.; Zhou, W. No-reference quality assessment of underwater image enhancement. *Displays* **2024**, *81*, 102586. [\[CrossRef\]](#)
- Guo, Q.; Liu, N.; Wang, Z.; Sun, Y. Review of deep learning based object detection algorithms. *J. Detect. Control* **2023**, *45*, 10–20.
- Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
- Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [\[CrossRef\]](#) [\[PubMed\]](#)
- 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.
- Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I*; Springer: Cham, Switzerland, 2016; pp. 21–37.
- Cao, R.; Zhang, R.; Yan, X.; Zhang, J. BG-YOLO: A bidirectional-guided method for underwater object detection. *Sensors* **2024**, *24*, 7411. [\[CrossRef\]](#) [\[PubMed\]](#)
- Liu, X.; Zhao, K.; Liu, C.; Chen, L. Bi2F-YOLO: A novel framework for underwater object detection based on YOLOv7. *Intell. Mar. Technol. Syst.* **2025**, *3*, 9. [\[CrossRef\]](#)
- Zhou, S.; Wang, L.; Chen, Z.; Zheng, H.; Lin, Z.; He, L. An improved YOLOv9s algorithm for underwater object detection. *J. Mar. Sci. Eng.* **2025**, *13*, 230. [\[CrossRef\]](#)
- Guo, A.; Sun, K.; Zhang, Z. A lightweight YOLOv8 integrating FasterNet for real-time underwater object detection. *J. Real-Time Image Process.* **2024**, *21*, 49. [\[CrossRef\]](#)
- Liu, K.; Peng, L.; Tang, S. Underwater object detection using TC-YOLO with attention mechanisms. *Sensors* **2023**, *23*, 2567. [\[CrossRef\]](#)
- Zhang, M.; Wang, Z.; Song, W.; Zhao, D.; Zhao, H. Efficient small-object detection in underwater images using the enhanced yolov8 network. *Appl. Sci.* **2024**, *14*, 1095. [\[CrossRef\]](#)
- Wu, D.; Luo, L. SVGS-DSGAT: An IoT-enabled innovation in underwater robotic object detection technology. *Alex. Eng. J.* **2024**, *108*, 694–705. [\[CrossRef\]](#)
- Lian, S.; Li, H.; Cong, R.; Li, S.; Zhang, W.; Kwong, S. Watermask: Instance segmentation for underwater imagery. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 1305–1315.

18. 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, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
19. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
20. Zhao, H.; Xu, C.; Chen, J.; Zhang, Z.; Wang, X. BGLE-YOLO: A Lightweight Model for Underwater Bio-Detection. *Sensors* **2025**, *25*, 1595. [[CrossRef](#)]
21. Zhang, X.; Liu, C.; Yang, D.; Song, T.; Ye, Y.; Li, K.; Song, Y. RFACConv: Innovating spatial attention and standard convolutional operation. *arXiv* **2023**, arXiv:2304.03198.
22. Xie, G.; Liang, L.; Lin, Z.; Lin, S.; Su, Q. Lightweight Underwater Target Detection Algorithm Based on Improved YOLOv8n. *Laser Optoelectron. Prog.* **2024**, *61*, 2437006.
23. Liu, W.; Lu, H.; Fu, H.; Cao, Z. Learning to upsample by learning to sample. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 6027–6037.
24. Chen, Z.; He, Z.; Lu, Z.M. DEA-Net: Single image dehazing based on detail-enhanced convolution and content-guided attention. *IEEE Trans. Image Process.* **2024**, *33*, 1002–1015. [[CrossRef](#)]
25. Zhang, Z.; Li, Y.; Bai, Y.; Li, Y.; Liu, M. Convolutional graph neural networks-based research on estimating heavy metal concentrations in a soil-rice system. *Environ. Sci. Pollut. Res.* **2023**, *30*, 44100–44111. [[CrossRef](#)]
26. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 12993–13000. [[CrossRef](#)]
27. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.
28. Lieberman, B.; Dahbi, S.E.; Mellado, B. The use of Wasserstein Generative Adversarial Networks in searches for new resonances at the LHC. *J. Phys. Conf. Ser.* **2023**, *2586*, 012157. [[CrossRef](#)]
29. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
30. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
31. Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. Yolov10: Real-time end-to-end object detection. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 107984–108011.
32. Zhou, H.; Kong, M.; Yuan, H.; Pan, Y.; Wang, X.; Chen, R.; Lu, W.; Wang, R.; Yang, Q. Real-time underwater object detection technology for complex underwater environments based on deep learning. *Ecol. Inform.* **2024**, *82*, 102680. [[CrossRef](#)]
33. 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. [[CrossRef](#)]
34. Jia, R.; Lv, B.; Chen, J.; Liu, H.; Cao, L.; Liu, M. Underwater object detection in marine ranching based on improved YOLOv8. *J. Mar. Sci. Eng.* **2023**, *12*, 55. [[CrossRef](#)]
35. Cheng, S.; Han, Y.; Wang, Z.; Liu, S.; Yang, B.; Li, J. An Underwater Object Recognition System Based on Improved YOLOv11. *Electronics* **2025**, *14*, 201. [[CrossRef](#)]
36. Fu, C.; Liu, R.; Fan, X.; Chen, P.; Fu, H.; Yuan, W.; Zhu, M.; Luo, Z. Rethinking general underwater object detection: Datasets, challenges, and solutions. *Neurocomputing* **2023**, *517*, 243–256. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.