electronics

MDPI

Article

DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n

Qiang Liu [1], Wei Huang [1,2,*], Xiaoqiu Duan [1], Jianghao Wei [1], Tao Hu [1], Jie Yu [1] and Jiahuan Huang [1]

[1] The School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China; liuqiangwit@outlook.com (Q.L.); dxiaoqiu@outlook.com (X.D.); hjh2492696300@163.com (J.H.)

[2] Hubei Provincial Key Laboratory of Intelligent Robots, Wuhan Institute of Technology, Wuhan 430205, China

* Correspondence: huangw@wit.edu.cn

Abstract: Underwater target detection is widely used in various applications such as underwater search and rescue, underwater environment monitoring, and marine resource surveying. However, the complex underwater environment, including factors such as light changes and background noise, poses a significant challenge to target detection. We propose an improved underwater target detection algorithm based on YOLOv8n to overcome these problems. Our algorithm focuses on three aspects. Firstly, we replace the original C2f module with Deformable Convnets v2 to enhance the adaptive ability of the target region in the convolution check feature map and extract the target region's features more accurately. Secondly, we introduce SimAm, a non-parametric attention mechanism, which can deduce and assign three-dimensional attention weights without adding network parameters. Lastly, we optimize the loss function by replacing the CIoU loss function with the Wise-IoU loss function. We named our new algorithm DSW-YOLOv8n, which is an acronym of Deformable Convnets v2, SimAm, and Wise-IoU of the improved YOLOv8n(DSW-YOLOv8n). To conduct our experiments, we created our own dataset of underwater target detection for experimentation. Meanwhile, we also utilized the Pascal VOC dataset to evaluate our approach. The mAP@0.5 and mAP@0.5:0.95 of the original YOLOv8n algorithm on underwater target detection were 88.6% and 51.8%, respectively, and the DSW-YOLOv8n algorithm mAP@0.5 and mAP@0.5:0.95 can reach 91.8% and 55.9%. The original YOLOv8n algorithm was 62.2% and 45.9% mAP@0.5 and mAP@0.5:0.95 on the Pascal VOC dataset, respectively. The DSW-YOLOv8n algorithm mAP@0.5 and mAP@0.5:0.95 were 65.7% and 48.3%, respectively. The number of parameters of the model is reduced by about 6%. The above experimental results prove the effectiveness of our method.

Keywords: underwater target detection; deformable convnets v2; SimAm; Wise-IoU

check for updates

Citation: Liu, Q.; Huang, W.; Duan, X.; Wei, J.; Hu, T.; Yu, J.; Huang, J. DSW-YOLOv8n: A New Underwater Target Detection Algorithm Based on Improved YOLOv8n. *Electronics* **2023**, *12*, 3892. https://doi.org/10.3390/electronics12183892

Academic Editors: Yoonsik Cho and Taehyeon Kim

Received: 22 August 2023
Revised: 10 September 2023
Accepted: 11 September 2023
Published: 15 September 2023

1. Introduction

The efficient use of computer vision technology to explore the unknown underwater domain is one of the most active research fields for many researchers. Due to the dynamic and changeable underwater visual environment, we must promote visual recognition tracking and dynamic perception algorithms to deal with the complex challenges [1,2]. Effectively utilizing these resources can help prevent the overexploitation and destruction of terrestrial resources. In underwater engineering applications and research exploration, an efficient and accurate target detection and recognition algorithm is needed for underwater unmanned vehicles or mobile devices [3,4]. Of course, the more robust target detection algorithm can be applied not only to underwater target detection, but also to other scenarios, including automatic driving and unmanned aerial vehicles [5,6].

However, the complex underwater environment can affect the detection results. Factors such as a lack of light due to weather conditions and changes in underwater brightness caused by water depth increase the difficulty of underwater target detection [7,8]. Some researchers have considered using artificial light sources to compensate for these challenges, but this approach may result in the presence of bright spots and worsen the scattering of underwater suspended objects under certain conditions, which can have a negative impact.

Considering the complexity of the underwater environment, we need to develop a target detection algorithm suitable for underwater equipment which requires a high precision and low computation as its advantages [9–11]. The YOLO series target detection algorithm is known for achieving a good balance between detection accuracy and speed [12–15]. This paper focuses on improving and enhancing the performance of the YOLOv8n algorithm by making improvements in three aspects:

(1) To improve adaptability to object deformations and enable more precise convolutional operations, we replace certain C2f modules in the YOLOv8n backbone feature extraction network with deformable convolutional v2 modules.

(2) We introduce an attention mechanism (SimAm) to the network structure, which does not introduce external parameters but assigns a 3D attention weight to the feature map.

(3) Resolving a problem with the loss function in which discrepancies between the direction of the prediction boxes and the ground truth bounding boxes may result in oscillations in the position of the prediction box during training, slowing convergence and lowering prediction accuracy. We suggest using the WIoU v3 loss function to better improve the network structure in order to get around this.

2. Related Work

2.1. Objection Detection Algorithm

YOLOv8 can flexibly support a variety of computer vision tasks; especially in the field of target detection, the YOLOv8 object detection model stands out as one of the top-performing models. YOLOv8 was built upon the YOLOv5 model, introducing a new network structure and incorporating the strengths of previous YOLO series algorithms and other state-of-the-art design concepts in target detection algorithms [16]. While YOLOv8 still utilizes the DarkNet53 structure in its network architecture, certain parts of the structure have been fine-tuned. For instance, the C3 module in the feature extraction network is replaced by C2f with a residual connection, which includes two convolution cross-stage partial bottlenecks. This modification allows for the fusion of advanced features and contextual information, resulting in enhanced detection accuracy. Additionally, the model structure of YOLOv8 sets different channel numbers for each version to enhance the model's robustness in handling various types of detection tasks. In the Head section, YOLOv8 continues the Anchor-free mechanism found in YOLOv6 [17], YOLOv7 [18], YOLOX [19], and DAMO-YOLO [20]. This mechanism reduces the computational resources required by the model and decreases the overall time consumption. YOLOv8 draws inspiration from the design ideas of YOLOX, using Decoupled Head for decoupling, so the accuracy of model detection is improved by about 1%. This design allows each branch to focus on the current prediction task, thereby improving the performance of the model. The loss function in YOLOv8 consists of two parts, sample matching and loss calculation. The loss function includes category loss and regression loss, among which the regression loss includes two parts: Distribution Focal Loss and CIoU loss [21].

Target detection algorithms can be categorized into one-stage and two-stage algorithms. The one-stage algorithm, represented by Faster R-CNN [22], is known for its slower processing speed, which makes it unsuitable for real-time target identification and detection. On the other hand, the two-stage algorithms, including the YOLO series and DETR series, offer significant advantages, while the DETR [23] network model is large, difficult to train, and exhibits a poor detection effect on small targets. To some extent, YOLO series algorithms excel in underwater target detection. Currently, in the YOLO series of object detection algorithms, some researchers do a lot of research work. Lou et al. [24] proposed a new method of downsampling on the basis of YOLOv8, which better retains the feature information of the context and improves the feature network to better combine shallow information and deep information. Zhang et al. [25] proposed to introduce the global attention mechanism into the YOLOv5 model to strengthen the feature extraction ability of the backbone network for key regions and introduce a multi-branch reparametrized structure to improve the multi-scale feature fusion. Lei et al. [26] used Swin transform as the backbone network of YOLOv5 and then improved the PAnet multi-scale feature fusion method and confidence loss function, which effectively improved the object detection accuracy and the robustness of the model. In this paper, we improved the network structure of YOLOv8n with Deformable Convnets v2, added a parameter-free attention mechanism, and finally optimized the loss function. The DSW-YOLOv8n can be divided into three parts: Backbone, Neck, and Detect. The Backbone consists of various convolutional modules. The Neck includes upsampling and concatenation operations in addition to the convolutional module. The network provides three prediction outputs for objects of different sizes. Finally, the predicted results are used to calculate the loss. The network structure of DSW-YOLOv8n is shown in Figure 1.

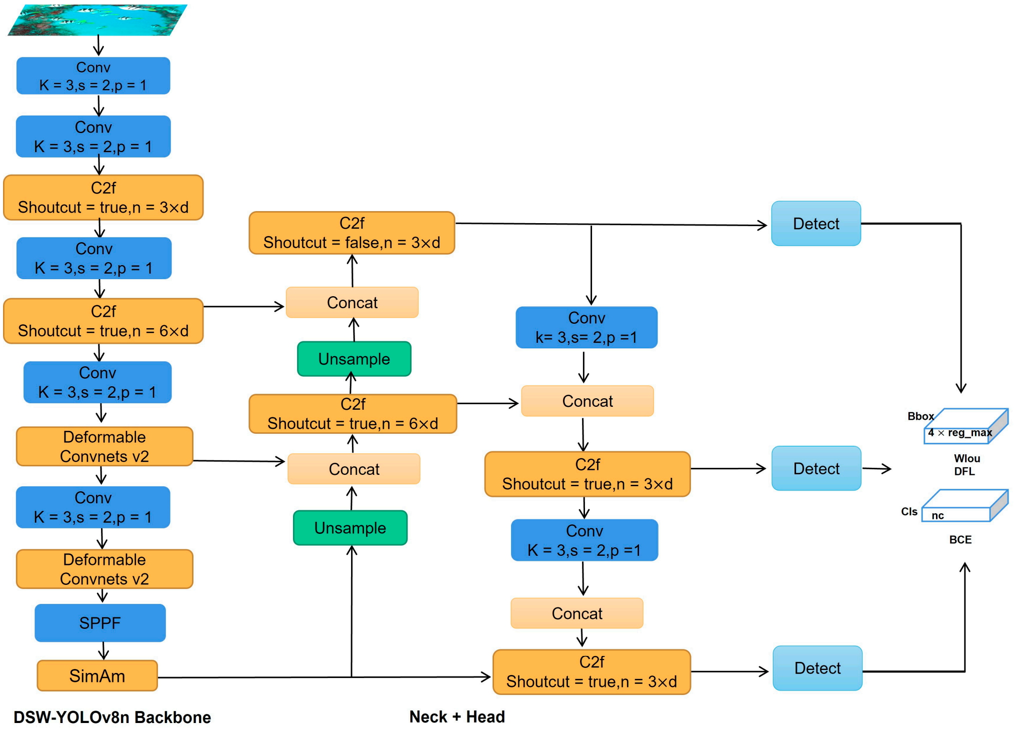

Figure 1. The network structure of DSW-YOLOv8n.

2.2. Fusion of Deformable Convolutional Feature Extraction Network

Deformable Convolution v2 [27] is an improved version of Deformable Convolution v1 [28], which further enhances and optimizes the previous method. In a common convolution module, fixed-size and shape convolution filters are used. However, during the feature extraction process, there may be interference where the convolution kernel does not align perfectly with the target region and includes excess background noise. In comparison, Deformable Convolution v2 introduces additional offsets, allowing the convolution operations to better align with the target region in the feature map. This enhancement in Deformable Convolution v2 provides improved modeling capabilities in two complementary forms. Firstly, it extends the use of deformable convolutional layers throughout the network. By incorporating more convolutional layers with adaptive learning, Deformable Convolution v2 can effectively control sampling across a wider range of feature levels. Secondly, an adjustment mechanism is introduced which not only enables each sample to experience learning shifts but also adaptively adjusts the learning target feature amplitude.

Compared with traditional convolution modules, deformable convolution is superior to traditional convolution in feature extraction accuracy. In the network structure of YOLOv8n, we adjusted some nodes in the network structure and replaced C2f modules at positions six and eight in the backbone network structure with Deformable Convnets V2 modules. The robustness of the model is effectively enhanced. The difference between the common convolution module and deformable convolution v2 is shown in Figure 2.

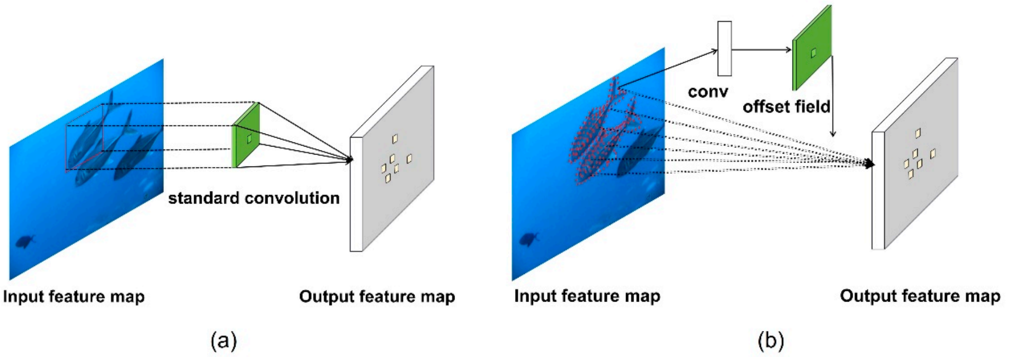

Figure 2. Common convolution and Deformable Convnets v2 are shown in (**a**,**b**).

The calculation formula for the output of the feature map obtained by the common convolution is shown in Equation (1). R represents the size of the convolution kernel, it also represents the area where convolution operations can be performed on the feature map. p_0 represents the position of the center point of the convolution kernel, while p_n represents the position of other pixel points relative to p_0. $w(p_n)$ represents the weight value at the n position, and $\mathrm{x}(p_0 + p_n)$ represents the pixel value at the n position.

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot \mathrm{x}(p_0 + p_n) \tag{1}$$

The calculation formula of Deformable Convnets v2 is shown in Equation (2). The common convolution region R is fixed, the deformable convolution region changes as the target changes so that K is a variable kernel size, p represents the position of the center point of the convolution kernel, p_k is the position of the position of other pixel points relative to p. Δp_k and Δm_k in the formula represent the learnable offset and modulation range at the k position. As Δp_k is a real number with an unconstrained range, we used Δm_k to limit it. The range of Δm_k is [0, 1]. From $p + p_k + \Delta p_k$ we may obtain a decimal, in which case, a bilinear interpolation will be used to change the number from a decimal to an integer. p_n and p_k have the same property; they, respectively, represent the position of pixels in the convolution region during their respective convolution operations.

$$y(p) = \sum_{k=1}^{k} w_k \cdot \mathrm{x}(p + p_k + \Delta p_k) \cdot \Delta m_k \tag{2}$$

2.3. Simple and Efficient Parameter-Free Attention Mechanism

Attention mechanisms are widely applied in both computer vision and natural language processing. In particular, high-resolution image processing tasks often face information processing bottlenecks. Drawing inspiration from human perception processes, researchers have been exploring selective visual attention models. We compare common attention mechanisms with SimAm, which includes CBAM [29,30], SE [31], and ECA [32]. The better attention mechanism of SimAm improves model accuracy without adding extra redundancy to the network. CBAM and SE increased by 9.23% and 9.6%, respectively, on ResNet101 [33]. Even worse, ECA's increase in the number of parameters is nearly three times that of the model. The channel attention mechanism compresses global information and learns from each channel dimension. It assigns different weights to different channels using an incentive method. On the other hand, the spatial attention mechanism combines global information to process important parts, transforming various spatial data and automatically selecting the more important area feature. Two attention mechanisms represent the 1D or 2D attention mechanisms, respectively. Underwater target detection differs from conventional target detection due to its susceptibility to illumination changes. One contributing factor is the varying light intensity caused by different weather conditions and time. Then, light transmission in the water will be affected by water absorption, reflection and scattering, and serious attenuation, which will directly result in the underwater image visible range being limited and blurred, with low contrast, color incongruity, background noise, and other problems. In order to reduce the impact of the above situation, we added the SimAm attention mechanism [34] to backbone's layer 10. The parameter-free attention mechanism is simple and efficient. Most of the operators are selected based on the energy function; no additional adjustments to the internal network structure are required [35]. The features with full 3D weights are shown in Figure 3.

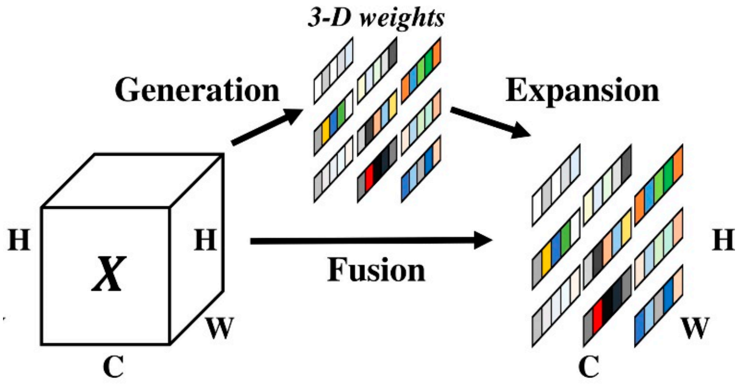

Figure 3. Full 3D weights for attention [33].

SimAm is inspired from neuroscience theory; the parameter-free attention mechanism establishes the energy function in order to obtain the importance of each neuron. The calculation formulate is shown in Equation (3).

$$\mathrm{e_t(w_t, b_t, y, x_i)} = \left(\mathrm{y_t} - \hat{t}\right)^2 + \frac{1}{\mathrm{M}-1}\sum_{\mathrm{i}=1}^{\mathrm{M}-1}\left(\mathrm{y_0} - \hat{x}_i\right)^2 \tag{3}$$

The linear transformations of t and x_i are represented by $\hat{t} = (\omega_t t + b_t)$ and $\hat{\mathrm{x}}_\mathrm{i} = \omega_\mathrm{t}\mathrm{x_i} + \mathrm{b_t}$, respectively. Here, ω_t and b_t denote the weights and biases after transformation. To simplify the formula, binary labels are used and regular terms are added to the equation. The energy function is defined as shown in Equation (4).

$$\begin{aligned} e_t(w_t, b_t, y, x_i) &= \tfrac{1}{\mathrm{M}-1}\sum_{i=1}^{\mathrm{M}-1}\left(-1 - (\omega_t x_i + b_t)\right)^2 \\ &\quad + (1 - (w_t t + b_t))^2 + \lambda w_t^2 \end{aligned} \tag{4}$$

Theoretically, each channel has M energy functions where M = H × W. However, iteratively solving this equation requires a lot of computational resources; there is a better optimization of the computation with w_t and b_t, which is shown in Equation (5).

$$\mathrm{w_t} = \frac{2(\mathrm{t} - \mu_\mathrm{t})}{(\mathrm{t} - \mu_\mathrm{t})^2 + 2\sigma_\mathrm{t}^2 + 2\lambda}, \mathrm{b_t} = -\frac{1}{2}(\mathrm{t} + \mu_\mathrm{t})\mathrm{w_t} \tag{5}$$

The mean value μ_t and variance σ_t^2 of other neurons in the channel can be calculated using the formulas $\mu_t = \frac{1}{M}\sum_{i=1}^{M-1} x_i$ and $\sigma_t^2 = \frac{1}{M}\sum_{i=1}^{M-1}(x_i - \mu_t)^2$. The λ represents the regularization parameter. The existing solution in formula (5) is obtained on a single channel, so it is reasonable to assume that the pixels in the channel all follow the same distribution. We can calculate the mean value and variance of all neurons and use it for all neurons on the channel. The method can reduce the computation amount well. Therefore, the calculation formula of minimum energy function is shown in Equation (6).

$$e^{min} = \frac{4\left(\hat{\sigma}^2 + \lambda\right)}{\left(t - \hat{\mu}\right)^2 + 2\hat{\sigma}^2 + 2\lambda} \tag{6}$$

If the value of e^{min} is lower, it means that the difference between neuron t and other neurons is more obvious; it also means that it's more important. The importance of each neuron can be obtained by e^{min}. Our approach treats each neuron individually and integrates this linear separability into an end-to-end framework, as shown in Equation (7).

$$\widetilde{\mathrm{X}} = \mathrm{sigmoid}\left(\frac{1}{\mathrm{E}}\right) \odot \mathrm{X} \tag{7}$$

The value of E is the energy function on each channel. Meanwhile, E groups are all e^{min} across channels and dimensions. Using the sigmoid activation function to prevent the value of E from getting too large. SimAm can be flexibly and easily applied to other target object algorithms, integrating it into the backbone network of YOLOv8n, effectively refining the characteristics of the channel domain and spatial domain, thereby significantly improving the accuracy of object detection without increasing the complexity and computing resources of the network [36].

2.4. Loss Function with Dynamic Focusing Mechanism

The loss function is essential for improving the performance of the model. The region between the predicted and ground truth bounding boxes is not taken into account by traditional loss functions, which only consider the overlap between them. If there is no intersection between the predicted and ground truth bounding boxes, this constraint becomes troublesome for small target identification because the loss function cannot be discriminated. Because of this, it is unable to optimize the network model, which causes variations in the evaluation results [37,38]. In the YOLOv8n network model, the Distribution Focal Loss and CIoU loss functions are employed as the loss functions. The CIoU loss function incorporates the loss in detection box scale and the loss in length and width ratio, in addition to the DIoU loss function. These enhancements contribute to improved accuracy in regression prediction. However, it is worth noting that the CIoU loss function requires more computational resources during model training within the original YOLOv8n network structure. Second, the datasets may contain low quality data samples, which may contain other background noise, an uncoordinated ratio of length to width, and other geometric factors which may further aggravate the negative impact of its training that cannot eliminate the negative impact of geometric factors. So, we improved our model by using Wise-IoU [39] to replace CIoU.

2.4.1. WIoU v1

Low quality datasets will inevitably have a negative impact on the model, which usually comes from geometric factors such as distance and aspect ratio, etc. Therefore, we used the WIoU v1 with two layers of attention based on the distance metric, as follows Equations (8) and (9) [40].

$$\mathcal{L}_{WIoUv1} = \mathcal{R}_{WIoU}\mathcal{L}_{IoU} \tag{8}$$

$$\mathcal{R}_{\mathrm{WIoU}} = \exp\left(\frac{\left(\left(x - x_{gt}\right)^2 + \left(y - y_{gt}\right)^2\right)}{\left(W_g^2 + H_g^2\right)^*}\right) \tag{9}$$

where $\mathcal{R}_{WIoU} \in$ [1, e), which can significantly enlarge the $\mathcal{L}_{IoU}$ of the anchor box. W_g and H_g are the minimum width and height of the enclosing box. By separating W_g and H_g from the computed graph, gradients that hinder convergence can be prevented without introducing new conditions such as aspect ratio. (The superscript * indicates this operation) [39].

2.4.2. WIoU v2

WIoU v2 borrows the design method of Focal Loss to construct a monotonic focusing coefficient on the basis of WIoU v1. However, it also has another problem with the introduction of this monotonic focusing coefficient, which will cause a gradient change when the model is backpropagated. The gradient gain decreases with the decrease in $\mathcal{L}_{IoU}$, which causes the model to take more time to converge at a later stage. Therefore, we take the mean of $\mathcal{L}_{IoU}$ as a normalization factor, which is a good way to speed up the later convergence of the model, where $\overline{\mathcal{L}_{IoU}}$ acts as the exponential running average with momentum [41].

$$\mathcal{L}_{WIoUv2} = \left(\frac{\mathcal{L}^{*}_{IOU}}{\overline{\mathcal{L}_{IoU}}}\right)^{\gamma} \mathcal{L}_{WIoUv1}\ ,\ \gamma > 0 \tag{10}$$

2.4.3. WIoU v3

The quality of the anchor box is reflected by defining an outlier value. A high quality anchor box has a smaller outlier value. Utilizing a higher quality anchor box to match a smaller gradient gain can better focus the bounding box regression frame more on the ordinary quality anchor box, and the small gradient gain can match the anchor frame with large outliers, which can better reduce the large harmful gradient produced by low-quality samples. Based on WIoU v1, a non-monotonic focusing coefficient β is constructed and the gradient gain is highest when the value of the β is constant. Due to $\overline{\mathcal{L}_{IoU}}$ it is dynamic, so the quality evaluation criteria of the anchor box is also dynamic, which allows WIoU v3 to dynamically adjust the gradient gain distribution strategy.

$$\mathcal{L}_{WIoUv3} = \frac{\beta}{\delta \alpha^{\beta - \delta}} \mathcal{L}_{WIoUv1} \tag{11}$$

$$\beta = \frac{\mathcal{L}^{*}_{IOU}}{\overline{\mathcal{L}_{IoU}}} \in [0, +\infty) \tag{12}$$

3. Experiment

3.1. Underwater Target Detection Dataset

Using the Pascal VOC dataset and a self-built dataset for underwater target detection, we validate our methods in this experiment. The Target Recognition Group of China Underwater Robot Professional Competition (URPC) provided the majority of the 585 photos that make up our underwater target identification dataset. The remaining images were gathered from the publicly accessible collection on the whale community platform. There are seven different categories in the dataset, including jellyfish, fish, sea urchins, scallops, sea grass, sharks, and sea turtles. LabelImg software was used to annotate every image in the collection and they are all in yolo format. The dataset is arbitrarily split into a 7:2:1 training set, test set, and validation set. We created a complete presentation of the underwater target detection dataset, which is presented in Table 1. It includes the total number of trial sets and samples for each category. Figure 4 displays a sample of the 1585 image collection for underwater target detection. We thoroughly examined the training sets in the experiment's training phase. We can see the training set in the dataset graphically in Figure 5. The quantity of samples for each category is shown in Figure 5a, and the size and quantity of ground truth boxes in the target area are shown in Figure 5b. It is evident that the dataset has a disproportionately higher percentage of small targets. Figure 5c,d, respectively, assesses the distribution of the target area's center points and the aspect ratio of the image label for the entire image.

Table 1. Quantity of images and samples in underwater target detection dataset.

Experiment Set	Train	Test	Validation	Total			
Quantity of images	1109	317	159	1585			
Category	jellyfish	fish	Sea urchin	Scallop	Sea grass	Shark	Sea turtle
Quantity of samples	356	1939	3335	1537	271	527	340

Figure 4. Some sample images of underwater target detection dataset.

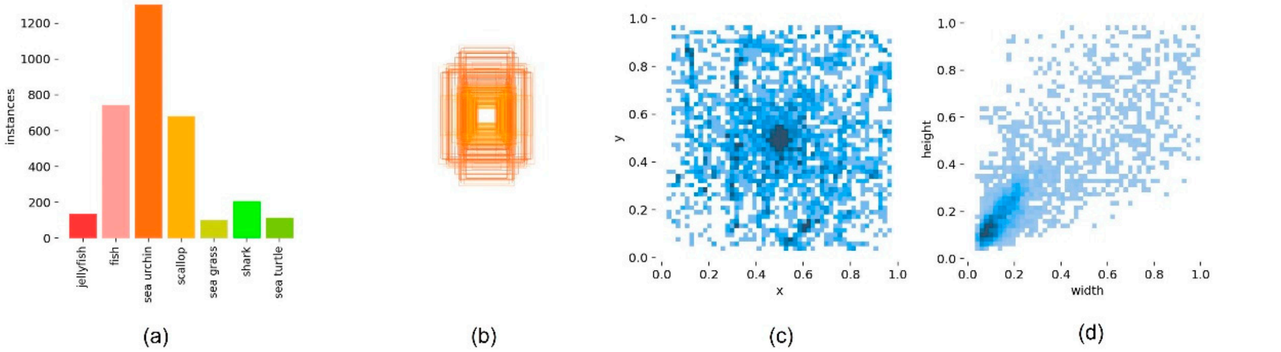

Figure 5. Analysis and presentation of underwater target detection dataset: (**a**) bar chart of the samples of each class of train set; (**b**) represents size and quantity of grand truth box; (**c**) is the position of the center point relative to the image; (**d**) represents the ratio of height and width of the object relative to the image.

3.2. Experimental Configuration and Environment

Our experiment made use of the Python programming language and the PyTorch deep learning framework, along with Ubuntu18.4 as the operating system. The hardware setup is displayed in Table 2 below. The following hyperparameters are used during training: the image's input size is 640 × 640, the training epoch total is 200, and the batch size is 16. Using the Stochastic Gradient Descent (SGD) to optimize the model, the initial learning rate is set to 0.01, the momentum is set to 0.973, and the weight attenuation is set to 0.0005. For trained dataset processing, we used a Mosaic data augment strategy and turned it off for the last ten epochs [19]. This strategy randomly cuts four images and changes the length to form an image.

Table 2. Experimental configuration and environment.

Environment	Version or Model Number
Operating System	Ubuntu18.04
CUDA Version	11.3
CPU	Intel(R) Xeon(R) CPU E5-2620 v4
GPU	Nvidia GeForce 1080Ti×4
RAM	126G
Python version	Python 3.8
Deep learning framework	Pytorch-1.12.0

3.3. Model Evaluation Metrics

We used the recall rate, average detection time, mean average precision (mAP), number of parameters, and floating-point operations per second (FLOPS) to evaluate the performance of the DSW-YOLOV8n model. *Precision* and *Recall* are as shown in Equations (13) and (14).

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

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

TP and FP are the proportion of positive samples in the dataset that are correctly predicted and incorrectly predicted, and FN is the quantity of samples in the negative sample that are incorrectly predicted. Average precision (*AP*) represents the average accuracy in the model. Mean average precision (*mAP*) is the average of *AP* values for all classes. x denotes the number of classes in the dataset. The calculation formulas are shown in Equations (15) and (16), respectively.

$$AP = \int_0^1 \mathrm{p(r)dr} \tag{15}$$

$$mAP = \frac{1}{\mathrm{x}}\sum_{\mathrm{i}=1}^{\mathrm{x}} AP_{\mathrm{i}} \tag{16}$$

4. Analysis and Discussion of Experimental Result

4.1. Comparison of Experimental Results of Different Model

To demonstrate the superiority of the DSW-YOLOv8n, we conducted a comparative experimental study using a YOLO series detection model. The performances of the DAMO-YOLO, YOLOv7, YOLOX, and the original YOLOv8n versions were specifically contrasted. Table 3's experimental findings contain metrics like Flops (the number of floating-point operations per second) and Params (the quantity of model parameters). At various IoU levels, we also assessed the mean average precision (mAP). When the IoU threshold is set to 0.5, the mAP@0.5 reflects the average across all categories and the mAP@0.5:0.95 represents the average mAP for each category at various thresholds ranging from 0.5 to 0.95 with a step size of 0.05.

Table 3. The result of comparative experiments of different models.

Model	Backbone	Flops/G	Params/M	mAP@0.5	mAP@0.5:0.95
DAMO-YOLO	CSP-Darknet	18.1	8.5	72.5	37.2
YOLOX	Darknet53	26.8	9.0	81.45	42.7
YOLOv7	E-ELAN	105.2	37.2	83.5	46.3
YOLOv8n	Darknet53	3.0	8.2	88.6	51.8
DSW-YOLOv8n	Darknet53(Our)	3.13	7.7	91.8	55.9

In the comparison experiment, all models used default parameters and the input image size for all models was set to 640 × 640. Notably, the DSW-YOLOv8n model exhibited a 3.2% increase in mAP@0.5 and a 4.1% increase in mAP@0.95, compared to the original YOLOv8n model. Furthermore, the number of parameters in the improved model was reduced by 6.1%. When compared to other mainstream target detection algorithms, the mAP@0.5 of the DSW-YOLOv8n was found to be 8.3%, 10.3%, and 19% higher than that of YOLOv7, YOLOX, and DAMO-YOLO, respectively. Similarly, the mAP@0.95 of the DSW-YOLOv8n was 9.6%, 13.2%, and 18.7% higher than that of YOLOv7, YOLOX, and DAMO-YOLO, respectively [17–20].

4.2. Comparison of Ablation Experiments

We tested each module in the DSW-YOLOv8n and examined how it affected the model in the ablation experiment. The loss function chooses the ideal WIoU v3 for the ablation experiment from among them. Table 4 presents the outcomes. According to the experiment's findings, Deformable Convnets v2, SimAm, and WIoU v3 have each increased the model's mAP@0.5 accuracy by 2.4%, 1.6%, and 3%, respectively. Additionally, by 2.2%, 3.4%, and 2%, mAP@0.5:0.95 increased. SimAm, on the basis of WIoU v3 and Deformable Convnets v2 in combination, increased the accuracy of mAP@0.5 by 2.9% and 0.1%, respectively. Additionally, mAP@0.5:0.95 went up by 2.5% and 3%, respectively [25]. Overall, Deformable Convnets v2 has increased the model detection accuracy. We used the Grad-CAM [41] image depicted in Figure 6 to visually contrast the effect before and after the SimAm module. Figure 6a depicts the initial input image, Figure 6b the standard output image, Figure 6c the thermal image following the addition of the SimAm module, and Figure 6d the thermal image output of the last layer of the backbone network. It can be seen that after adding the SimAm module, the information about the target area becomes more prominent in the output image by comparing the thermal effect plots of Figure 6b,c. Thus, it will be easier to see the heat effect [34].

Table 4. Ablation experiments of each method.

Model	Flops/G	Params/M	Average Detection Time/ms	Recall	mAP@0.5	mAP@0.5:0.95
YOLOv8n	3.0	8.2	5	84.8	88.6	51.8
YOLOv8n + DefConv2	3.13	7.7	7.4	86.1	91.0	54
YOLOv8n + SimAm	3.0	8.2	10.1	87.5	90.2	55.2
YOLOv8n + WIoUv3	3.0	8.2	5.4	85.9	91.6	53.8
YOLOv8n + DefConv2 + SimAm	3.13	7.7	10.6	80.4	91.6	53.5
YOLOv8n + DefConv2 + WIoUv3	3.13	7.7	8.1	85.8	91.5	54.3
YOLOv8n + SimAm + WIoUv3	3.0	8.2	5	81.4	88.5	54.8
DSW-YOLOv8n	3.13	7.7	8.7	85.1	91.8	55.9

In the second ablation experiment, we specifically targeted the Wise-IoU function to observe its enhancement effect on the model. Table 5 displays the outcomes of the experiment. The addition of WIoU v1, WIoU v2, and WIoU v3 was based on the addition of Deformable Convnets v2 and SimAm to the model, respectively. In comparison to WIoU v1 and WIoU v2, the experimental results demonstrate that mAP@0.5 of WIoU v3 increases by 0.86% and 0.7%, and mAP@0.5:0.95 by 0.1% and 0.6%, respectively. The average detection speed of each image is simultaneously slowed down by 0.03 ms and 1.9 ms, respectively. With the help of the aforementioned comparison analysis, WIoU v3 can enhance the performance of our model. For the purpose of displaying the prediction results, we have selected four situations that represent various object categories. Figure 7a,b shows the detection of small targets and objects with low visibility and high density, respectively. Target detection and recognition were depicted in Figure 7c,d in a general situation. The outcomes in Figure 7 demonstrate that no missed detections or errors were made, proving the reliability of DSW-YOLOv8n.

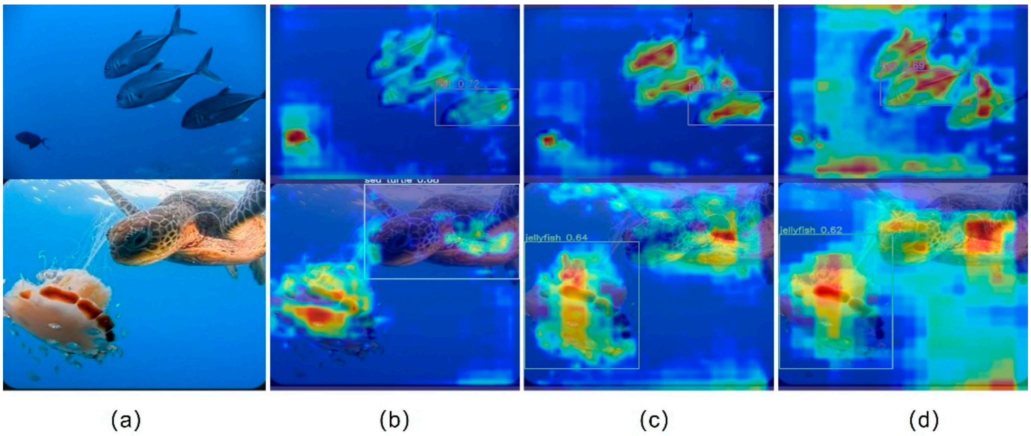

Figure 6. Grad-CAM image of the DSW-YOLOv8n. (**a**) Represents original image with the fish and sea turtle; (**b**) before adding the SimAm; (**c**) result after adding SimAm; and (**d**) the last layer output of the backbone.

Table 5. Wise-IoU ablation experiment.

YOLOv8n					Average Detection Time/ms	mAP@0.5	mAP@0.5:0.95
DefConv2	SimAm	WIoUv1	WIoUv2	WIoUv3			
√	√	√			8.73	90.94	55.8
√	√		√		10.6	91.01	55.3
√	√			√	8.7	91.8	55.9

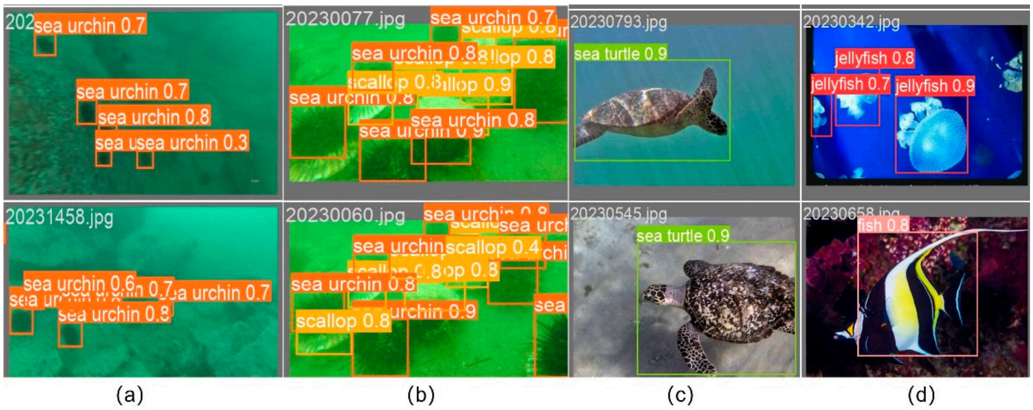

Figure 7. Results of our method's detection in DSW-YOLOv8n. The detection results for tiny targets and targets in challenging underwater environments are shown in (**a**,**b**), while the detection results for a typical situation are shown in (**c**,**d**).

4.3. Pascal VOC Dataset Experimental Results

The PASCAL Visual Object Classes is an open world-class computer vision challenge. The dataset can be applied to classification, localization, detection, segmentation, and action recognition tasks. To validate our method further, we utilized the Pascal VOC dataset which consists of 17,125 images across 20 categories. We used Pascal VOC2012 to further verify and analyze our model. Our experiment involved dividing the dataset into a training set (12,330 images), a test set (3425 images), and a validation set (1370 images), following a 7:2:1 ratio. The hyperparameters used during model training were consistent with experiment of the underwater target detection dataset. Due to the larger quantity of the Pascal VOC2012 dataset and slower model convergence, we increased the number of epochs trained to 300. The detailed experimental results are presented in Table 5, where the inclusion of the Deformable Convnets v2, SimAm, and WIoU v3 modules led to improvements of

2.5%, 1.9%, and 1.3%, respectively. This demonstrates that these three methods effectively enhance the detection accuracy. Additionally, when comparing the number of parameters in the model, the addition of Deformable Convnets v2 resulted in a 4.8% reduction, while the inclusion of the SimAm module improved the detection accuracy and recall without altering the number of parameters in the model. The effectiveness of WIoU v1, WIoU v2, and WIoU v3 on the model based on Deformable Convnets v2 and SimAm was analyzed. Table 6 shows that WIoU v3 achieved the highest detection accuracy, with mAP@0.5 and mAP@0.95 being 3.5% and 2.4% higher than the original model, respectively.

Table 6. The experimental result of Pascal VOC dataset.

Dataset	Model	Flops/G	Params/M	Recall	mAP@0.5	mAP@0.95
Pascal VOC 2012	YOLOv8n	3.0	8.2	55.1	62.2	45.9
	YOLOv8n + DefConv2	3.13	7.8	56.3	64.7	48
	YOLOv8n + SimAm	3.0	8.2	58.3	64.1	47.5
	YOLOv8n + WIoUv1	3.0	8.2	55.2	63.3	46.5
	YOLOv8n + WIoUv2	3.0	8.2	56.8	63.9	46.7
	YOLOv8n + WIoUv3	3.0	8.2	55.5	63.5	46.5
	YOLOv8n+DefConv2 + SimAm	3.13	7.8	55.8	64.4	48.2
	YOLOv8n+DefConv2 + WIoUv1	3.13	7.8	58.6	65.4	48.4
	YOLOv8n+DefConv2 + WIoUv2	3.13	7.8	56.9	65.1	48.1
	YOLOv8n+DefConv2 + WIoUv3	3.13	7.8	57.8	64.9	47.6
	YOLOv8n+SimAm + WIoUv1	3.0	8.2	57	63.8	46.8
	YOLOv8n+SimAm + WIoUv2	3.0	8.2	53.6	62.8	45.6
	YOLOv8n+SimAm + WIoUv3	3.0	8.2	54.5	64.2	46.4
	YOLOv8n + DefConv2 + SimAm + WIoUv1	3.13	7.8	56.5	64.5	47.7
	YOLOv8n + DefConv2 + SimAm + WIoUv2	3.13	7.8	59.8	64.7	47.3
	DSW-YOLOv8n	3.13	7.8	59.5	65.7	48.3

To visually observe the impact of the three versions of Wise-IoU on the model, we plotted the mAP@0.5 accuracy and DFL-loss curves in Figure 8. The red curve represents the performance after integrating Deformable Convnets v2, SimAm, and WIoU v3, indicating that the model has reached an optimal state. Compared to WIoU v1, mAP@0.5 and mAP@0.5:0.95 were improved by 1.2% and 0.6%, and there was a 1% improvement relative to WIoU v2. The experimental results on the Pascal VOC2012 dataset align with the results of our own underwater target dataset, confirming the effectiveness of DSW-YOLOv8n.

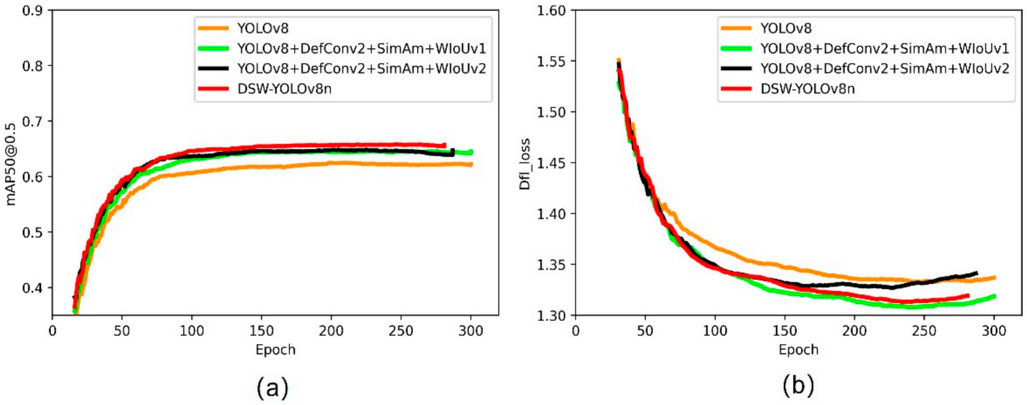

Figure 8. mAP@0.5 precision changes are shown in (**a**); DFL-Loss curve changes are shown in (**b**).

5. Conclusions

In this paper, the difficulties caused by subpar underwater image quality and intricate environmental changes are discussed in this research. We suggest three methods to YOLOv8n to address these problems. We conducted comparison studies, ablation experiments, and validation on two datasets using various methods and combinations. The experimental findings show that the three methods further optimize the model and have a good effect on its accuracy.

Firstly, we enhance the feature extraction capability of the backbone network by replacing the two-layer convolutional module with Deformable Convnets v2. This improvement has significantly increased the mAP@0.5 accuracy of both datasets by 2.4% and 2.5%, respectively. Although the number of parameters in the model has been reduced by 6%, the floating-point computation has increased by 4%, which is not the desired outcome. We provide a detailed analysis of the principle and formula of Deformable Convnets v2, in Equation (2), which involves adding an extra offset Δp_k calculated through forward inference and back propagation. The process of getting Δp_k slightly increases the computational effort of the DSW-YOLOv8n. Secondly, we introduce a flexible and efficient SimAm module in the last layer of the backbone. The core idea behind SimAm is to assign attention weight vectors to different positions of the input feature map. By refining the channel weight allocation, the model becomes more focused on the target region without adding extra parameters. The performance of the model is significantly improved, with a mAP@0.5 increase of 1.9% on both the underwater target detection dataset and the Pascal VOC dataset. Finally, we optimize the loss function by using the dynamic non-monotonically focused bounding box loss instead of the original CIoU. This modification effectively mitigates the negative impact of low-quality data on the model. Through ablation experiments, we demonstrate that WIoU v3 outperforms WIoU v1 and WIoU v2 in terms of the improvement effect, average detection speed, and detection accuracy of the model. As a result, mAP@0.5 improves, by 3.3% and 1.3%, the underwater target detection dataset and Pascal VOC dataset, respectively.

However, there are some imperfections in our work and the computational requirements of the model have slightly increased. In future work, we will continuously optimize our algorithm of DSW-YOLOv8n. Considering the potential application on mobile devices, the computational load is an important factor to consider. Therefore, our future goal is to explore methods for effectively reducing the amount of floating-point computation in the model and developing a more lightweight object detection model.

Author Contributions: Conceptualization, Q.L.; Methodology, W.H.; Software, Q.L.; Formal analysis, Q.L.; Investigation, J.W. and J.Y.; Resources, W.H. and T.H.; Data curation, X.D., T.H. and J.H.; Writing—original draft, Q.L.; Visualization, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Hubei Province Science and Technology Innovation Talents Project (grant number 2023DJC070).

Data Availability Statement: No applicated.

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

References

1. Sun, Y.; Zheng, W.; Du, X.; Yan, Z. Underwater small target detection based on yolox combined with mobilevit and double coordinate attention. *J. Mar. Sci. Eng.* **2023**, *11*, 1178. [CrossRef]
2. Zvarikova, K.; Rowland, Z.; Nica, E. Multisensor fusion and dynamic routing technologies, virtual navigation and simulation modeling tools, and image processing computational and visual cognitive algorithms across web3-powered metaverse worlds. *Anal. Metaphys.* **2022**, *21*, 125–141.
3. Kovacova, M.; Oláh, J.; Popp, J.; Nica, E. The algorithmic governance of autonomous driving behaviors: Multi-sensor data fusion, spatial computing technologies, and movement tracking tools. *Contemp. Read. Law Soc. Justice* **2022**, *14*, 27–45.
4. Yan, J.; Zhou, Z.; Zhou, D.; Su, B.; Xuanyuan, Z.; Tang, J.; Lai, Y.; Chen, J.; Liang, W. Underwater object detection algorithm based on attention mechanism and cross-stage partial fast spatial pyramidal pooling. *Front. Mar. Sci.* **2022**, *9*, 1056300. [CrossRef]
5. Wang, X.; Xue, G.; Huang, S.; Liu, Y. Underwater object detection algorithm based on adding channel and spatial fusion attention mechanism. *J. Mar. Sci. Eng.* **2023**, *11*, 1116. [CrossRef]
6. Novak, A.; Sedlackova, A.N.; Vochozka, M.; Popescu, G.H. Big data-driven governance of smart sustainable intelligent transportation systems: Autonomous driving behaviors, predictive modeling techniques, and sensing and computing technologies. *Contemp. Read. Law Soc. Justice* **2022**, *14*, 100–117.

7. Wen, G.; Li, S.; Liu, F.; Luo, X.; Er, M.-J.; Mahmud, M.; Wu, T. Yolov5s-ca: A modified yolov5s network with coordinate attention for underwater target detection. *Sensors* **2023**, *23*, 3367. [CrossRef]
8. Zhang, C.; Zhang, G.; Li, H.; Liu, H.; Tan, J.; Xue, X. Underwater target detection algorithm based on improved yolov4 with semidsconv and fiou loss function. *Front. Mar. Sci.* **2023**, *10*, 1153416. [CrossRef]
9. Lei, Z.; Lei, X.; Zhou, C.; Qing, L.; Zhang, Q. Compressed sensing multiscale sample entropy feature extraction method for underwater target radiation noise. *IEEE Access* **2022**, *10*, 77688–77694. [CrossRef]
10. Li, W.; Zhang, Z.; Jin, B.; Yu, W. A real-time fish target detection algorithm based on improved yolov5. *J. Mar. Sci. Eng.* **2023**, *11*, 572. [CrossRef]
11. Zhang, Y.; Ni, Q. A novel weld-seam defect detection algorithm based on the s-yolo model. *Axioms* **2023**, *12*, 697. [CrossRef]
12. 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.
13. 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.
14. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
15. Terven, J.; Cordova-Esparza, D. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv* **2023**, arXiv:2304.00501.
16. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W. Yolov6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
17. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
18. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
19. 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.
20. 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 Hilton Midtown, NY, USA, 7–12 February 2020; pp. 12993–13000.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149. [CrossRef]
22. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
23. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. Dc-yolov8: Small-size object detection algorithm based on camera sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]
24. 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]
25. Lei, F.; Tang, F.; Li, S. Underwater target detection algorithm based on improved yolov5. *J. Mar. Sci. Eng.* **2022**, *10*, 310. [CrossRef]
26. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316.
27. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
28. Guo, M.-H.; Xu, T.-X.; Liu, J.-J.; Liu, Z.-N.; Jiang, P.-T.; Mu, T.-J.; Zhang, S.-H.; Martin, R.R.; Cheng, M.-M.; Hu, S.-M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
29. 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; pp. 3–19.
30. 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.
31. 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.
32. Yang, L.; Zhang, R.-Y.; Li, L.; Xie, X. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International Conference on Machine Learning*; PMLR: Westminster, UK, 2021; pp. 11863–11874.
33. Lai, Y.; Ma, R.; Chen, Y.; Wan, T.; Jiao, R.; He, H. A pineapple target detection method in a field environment based on improved yolov7. *Appl. Sci.* **2023**, *13*, 2691. [CrossRef]
34. Dong, C.; Cai, C.; Chen, S.; Xu, H.; Yang, L.; Ji, J.; Huang, S.; Hung, I.-K.; Weng, Y.; Lou, X. Crown width extraction of metasequoia glyptostroboides using improved yolov7 based on uav images. *Drones* **2023**, *7*, 336. [CrossRef]
35. Mao, R.; Wang, Z.; Li, F.; Zhou, J.; Chen, Y.; Hu, X. Gseyolox-s: An improved lightweight network for identifying the severity of wheat fusarium head blight. *Agronomy* **2023**, *13*, 242. [CrossRef]
36. 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.

37. 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]
38. Tong, Z.; Chen, Y.; Xu, Z.; Yu, R. Wise-iou: Bounding box regression loss with dynamic focusing mechanism. *arXiv* **2023**, arXiv:2301.10051.
39. Zhu, Q.; Ma, K.; Wang, Z.; Shi, P. Yolov7-csaw for maritime target detection. *Front. Neurorobot.* **2023**, *17*, 1210470. [CrossRef]
40. Zhao, Q.; Wei, H.; Zhai, X. Improving tire specification character recognition in the yolov5 network. *Appl. Sci.* **2023**, *13*, 7310. [CrossRef]
41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.

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.