sensors

MDPI

Article

SD-YOLOv8: An Accurate *Seriola dumerili* Detection Model Based on Improved YOLOv8

Mingxin Liu [1,2], Ruixin Li [3], Mingxin Hou [2,4,*], Chun Zhang [1], Jiming Hu [1] and Yujie Wu [3]

[1] School of Electronics and Information Engineering, Guangdong Ocean University, Zhanjiang 524088, China; liumx@gdou.edu.cn (M.L.); 2112210031@stu.gdou.edu.cn (C.Z.); 2112210006@stu.gdou.edu.cn (J.H.)
[2] Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching, Zhanjiang 524088, China
[3] Naval Architecture and Shipping College, Guangdong Ocean University, Zhanjiang 524088, China; 2112212010@stu.gdou.edu.cn (R.L.); 2112312017@stu.gdou.edu.cn (Y.W.)
[4] School of Mechanical Engineering, Guangdong Ocean University, Zhanjiang 524088, China
* Correspondence: houmx@gdou.edu.cn

Abstract: Accurate identification of *Seriola dumerili* (SD) offers crucial technical support for aquaculture practices and behavioral research of this species. However, the task of discerning *S. dumerili* from complex underwater settings, fluctuating light conditions, and schools of fish presents a challenge. This paper proposes an intelligent recognition model based on the YOLOv8 network called SD-YOLOv8. By adding a small object detection layer and head, our model has a positive impact on the recognition capabilities for both close and distant instances of *S. dumerili*, significantly improving them. We construct a convenient *S. dumerili* dataset and introduce the deformable convolution network v2 (DCNv2) to enhance the information extraction process. Additionally, we employ the bottleneck attention module (BAM) and redesign the spatial pyramid pooling fusion (SPPF) for multidimensional feature extraction and fusion. The Inner-MPDIoU bounding box regression function adjusts the scale factor and evaluates geometric ratios to improve box positioning accuracy. The experimental results show that our SD-YOLOv8 model achieves higher accuracy and average precision, increasing from 89.2% to 93.2% and from 92.2% to 95.7%, respectively. Overall, our model enhances detection accuracy, providing a reliable foundation for the accurate detection of fishes.

Keywords: *Seriola dumerili*; attention mechanism; YOLOv8; deformable convolution; small object detection

Citation: Liu, M.; Li, R.; Hou, M.; Zhang, C.; Hu, J.; Wu, Y. SD-YOLOv8: An Accurate *Seriola dumerili* Detection Model Based on Improved YOLOv8. *Sensors* **2024**, *24*, 3647. https://doi.org/10.3390/s24113647

Academic Editor: Teruhisa Komatsu

Received: 8 April 2024
Revised: 12 May 2024
Accepted: 30 May 2024
Published: 4 June 2024

1. Introduction

Seriola dumerili (SD), commonly known as "Kanpachi", is a fish species primarily found in the upper layers of the ocean. It inhabits tropical and temperate waters of major ocean regions [1]. Species of the genus *Seriola* are targeted for aquaculture due to their fast growth and large size. Among them, *S. dumerili* is distributed in tropical seas and is actively farmed in the Mediterranean Sea and Asian seas [2–4], holding significant value in both commercial and recreational fisheries. Its firm and flavorful flesh makes it a prized target for various culinary applications, including in baking, in grilling, and as sashimi [5]. In the complex underwater environment, conventional aquaculture practices heavily rely on artificial observation and experimental judgment [6,7]. These methods are particularly challenging when dealing with schools of fish, where misdetection and missed detections are common occurrences. The limitations of human observation extend to the difficulty in accurately identifying both nearby and distant fish as well as small fish, leading to low management effectiveness and the formulation of incorrect strategies in aquaculture [8]. The need for intelligent and accurate detection of fish underwater is paramount for advancing fish behavior research, implementing intelligent feeding strategies, and monitoring fish health

and diseases. To address these challenges, this paper introduces an improved detection model designed to be robustly applied to the task of underwater fish recognition.

Fish recognition tasks are particularly challenging compared to the identification of other terrestrial organisms [9–11]. First, dataset acquisition is difficult due to the stringent requirements for collection devices in underwater environments. The existing publicly available datasets, such as LifeCLEF15 [12], Fish4Knowledge [13], LSCD1 [14], and Fish-Pak [15], are often plagued by image noise, blurriness, and suboptimal lighting conditions. Second, accurately capturing the texture, color, and shape information of fish within complex aquatic environments is a daunting task. Traditional fish recognition methods depend on feature extraction for target classification. For example, Fouad et al. [16] utilized support vector machines alongside the scale-invariant feature transform (SIFT) and speeded-up robust features (SURF) algorithms for feature extraction to classify Nile tilapia based on local features. Ravanbakhsh et al. [17] employed Haar features in conjunction with principal component analysis (PCA) for the classification of southern bluefin tuna in aquaculture settings. Bilal et al. [18] adopted the centroid–contour distance method for classifying fish species with dual dorsal fins. Cutter et al. [19] implemented a cascade of Haar features for the detection and recognition of benthic fish during unconstrained underwater surveys. Dhawal and Chen [20] generated representative feature vectors through the use of a histogram of oriented gradients (HOG) and color histograms for the identification of 10 similar fish species. While these appearance-based techniques have shown commendable detection performance in static images, they require considerable human effort in feature design and are less robust in adapting to dynamic and complex underwater environments, exacerbated by limited data availability.

The advent of deep learning has ushered in new avenues for fish object detection. Deep learning techniques can autonomously extract both low- and high-level features from extensive datasets. The inherent complexity of deep learning, with its layered architectures and self-learning algorithms, allows them to capture intricate details that may elude traditional detection methods, thus enhancing the overall accuracy and adaptability of fish recognition systems in diverse aquatic environments. For instance, Li et al. [21] designed an automated fish recognition system using the Fast R-CNN framework, achieving an average recognition accuracy of 81.2% across 24,277 images of 18 fish species. Notably, the system required only 4 h to train and processed individual images in 0.311 s, indicating a substantial enhancement in the efficiency of fish object detection. In another study, Salman et al. [22] leveraged a mixture of Gaussian mixture models (GMMs) and optical flow to isolate initial motion features from video footage of fish. These motion-centric features, combined with texture and shape data from the original frames, informed the training of the R-CNN network. This approach achieved detection accuracies of 87.44% and 80.02% on the Fish4Knowledge and LifeCLEF2015 fish datasets, respectively. Lin et al. [23] introduced the RoIMix methodology, which involves factors related to the proximity and occlusion that are typical of interactions between underwater organisms. By simulating target overlap, blending, and blurring from disparate images, the method aims to bolster the interaction representations between images and enhance model generalizability. Jalal et al. [24] proposed the YOLO-Fish detection model, which includes YOLO-Fish-1 and YOLO-Fish-2 iterations. The former iteration refined YOLOv3's performance by adjusting the upsampling stride, while the latter added a spatial pyramid pooling layer to better capture fish appearances in dynamic settings. Zhang et al. [25] reconceptualized the convolution module, network structure, loss function, and detection head for the YOLOv4 network, incorporating attention mechanisms and activation functions. The resulting algorithm not only achieved a 91.1% detection accuracy but also achieved a clock at 58.1 frames per second (FPS) on the URPC dataset, underscoring its real-time detection capabilities. Addressing the need for lightweight models, Liu et al. [26] devised Tuna-YOLO, optimized for tuna detection and suited for mobile device deployment and real-time applications. Similarly, HU et al. [27] crafted a swift and cost-effective detection system, leveraging underwater imaging coupled with deep learning frameworks to monitor fish behavior within hybrid aquaculture contexts.

Moreover, Zhou et al. [28] developed a precise detection method for oriental Takifugu rubripes based on YOLOv7, which is adept at managing multiscale detection of small targets and enhancing information extraction—imperative for smart fish farming. While these studies address various aspects of fish recognition, such as model training speed, detection speed, and feature extraction efficacy, challenges remain. The datasets employed primarily expanded the data volume without ample consideration of the image quality or the morphological details of the fish under diverse environmental conditions, illumination conditions, or viewing angles. Given the three-dimensional nature of fish movement in water, the recognition process necessitates gathering morphological data from multiple perspectives to refine the model's accuracy and generalizability. Furthermore, intricate underwater settings demand focused detection of minute targets and the discernment of granular features to minimize false positives and negatives. Crucially, an equilibrium between model compactness and detection performance is essential for informing practical applications in aquaculture and production deployment.

In this paper, the SD-YOLOv8 detection model is introduced, utilizing the advanced YOLOv8 network architecture for enhanced detection of *S. dumerili*. Given the limited availability of *S. dumerili* data in existing public datasets, this study undertakes the creation of a comprehensive dataset of *S. dumerili*. The dataset encompasses a variety of angles, lighting conditions, and resolutions. To further refine the dataset, image augmentation techniques are deployed on the blurred and low-light images, bringing the defining features of *S. dumerili* such as color and texture into sharper focus. The YOLOv8-based model architecture is meticulously reworked to improve the detection process. In this renovation, a dedicated layer for small object detection is introduced, along with sophisticated detection heads. Deformable convolutions are integrated into the architecture, enabling the model to fine-tune sampling offsets and weights, and, in turn, reducing the granularity of information loss. Additionally, the model assimilates the BAM attention mechanism and a sophisticated SPPF layer. These modifications facilitate the model's adaptability to a spectrum of angles and resolutions across both spatial and channel dimensions, which broadens the scope of information fusion and significantly improves the model's generalizability. To enhance the detection of objects that are overlapping, blurry, or small—especially at the image peripheries, a loss function is proposed. This function is designed to improve the model's convergence speed and precision. When pitted against other object detection models specializing in *S. dumerili* detection, the SD-YOLOv8 model clearly outperforms the other models in terms of detection performance, as evidenced by comparative evaluations. Such assessments underscore the model's superior capabilities and confirm the efficacy of the proposed enhancements in real-world applications.

The contributions of this paper are as follows:

(1) Creation of an *S. dumerili* dataset. We simulated real-world scenarios and constructed a dataset with diverse angles, lighting conditions, and resolutions, specifically focusing on *S. dumerili* The dataset was further enhanced using image augmentation techniques to highlight the appearance features of *S. dumerili*.

(2) Redesigned YOLOv8 network architecture. We improved the efficiency of *S. dumerili* detection by introducing new components to the YOLOv8 architecture. This includes adding a small object detection layer and detection heads, incorporating deformable convolutions for better information sampling, and integrating BAM attention and an improved SPPF layer to handle different angles and resolutions.

(3) Inner-MPDIoU loss function. To enhance the model's ability to detect overlapping, blurry, and small objects at the edges, we propose the Inner-MPDIoU loss function. This loss function improves the convergence speed and accuracy during training, leading to better detection performance.

2. Methods

2.1. SD-YOLOv8

The original YOLOv8 model consists of three components: the backbone network, the neck network, and the detection head. The backbone network is responsible for image feature extraction, the neck network handles feature fusion, and the detection head performs object detection at different scales. In this paper, improvements were made to each part of the study to enhance the efficiency of fish detection via SD-YOLOv8, as shown in Figure 1. The backbone utilizes the CSPNet processing concept from YOLOv5, built upon the DarkNet53 feature extraction network to process image features. First, a deformable convolution network (DCNv2) [29] was introduced to the C2f module, allowing for expanded sampling-point offsets and enriched multiscale sampling information, which can effectively cope with the distortion of an underwater environment. DCNv2 incorporates weight coefficients to enhance the accuracy of feature extraction. Second, the bottleneck attention mechanism (BAM) was employed to downsample the mapped image information, creating a multiparameter hierarchical attention structure to enhance channel and spatial mapping capabilities. Finally, a large-kernel separable attention (LSKA) [30] was used to strengthen the robustness of shape information encoding in feature representation, improving the long-range dependency and enhancing the feature fusion capability of the SPPF structure. The neck adopts the PANet concept to further enhance feature fusion at different scales, making it more suitable for object detection. Semantic information extracted from different levels of the backbone network is downsampled and used as input for PANet. Additionally, a small object detection layer was added to the neck to address situations where fish features are not prominent in low-light environments. It concatenates shallow and deep feature maps for detection and adds an auxiliary detection head to the head network for detecting small objects. The detection head adopts a decoupled head method to accelerate model convergence by separating the regression and prediction branches. The regression branch is evaluated using bounding box loss, which includes CIou [31] and distributional focal loss components. The prediction branch is evaluated using binary cross-entropy loss. Both branches are learned separately and then merged. To address the lack of advantages of the complete intersection over union (CIOU) metric in handling overlapping fish bodies, the Inner-MPDIoU metric, a loss function that fully utilizes geometric features of bounding boxes and includes auxiliary bounding boxes, is used to enhance feature convergence and improve model regression efficiency.

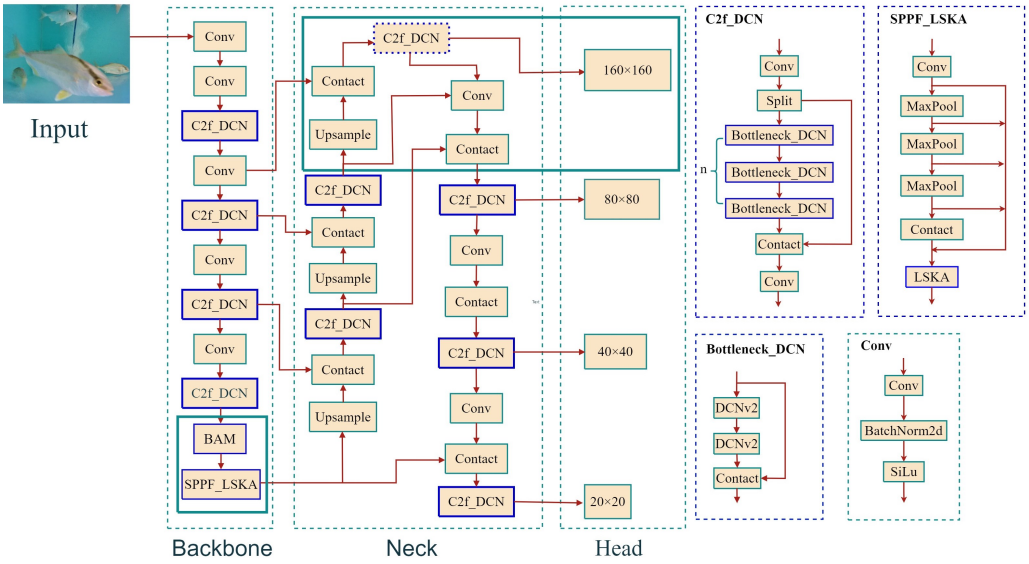


Figure 1. SD-YOLOv8 structure.

2.2. Small Object Detection Layer

The original YOLOv8 architecture, which utilizes downsampling layers with larger strides within its neck network, plays a pivotal role in reducing the model's complexity and streamlining training and inference, contributing to a more lightweight model. However, this configuration poses significant challenges for deep-level feature maps in terms of capturing details about small objects, a drawback that becomes particularly evident in the realm of fish detection. Specifically, small *S. dumerili* that are positioned on the periphery of the image or shrouded within low-light conditions may elude comprehensive detection due to these inherent limitations. The peripheral or low-light presence of these specimens underscores the need for a more nuanced approach that can effectively handle the various challenges presented by the complex, dynamic, and often unpredictable nature of aquatic environments. This paper presents an innovative redesign of the feature extraction mechanism in the neck network to remedy this. The new design incorporates a layer dedicated to small object detection along with an accompanying detection head. The restructured neck network includes supplementary upsampling and convolutional layers that not only deepen the architecture but also widen the receptive field. This enhancement bolsters the integration of features across different levels, leveraging contextual information more effectively and improving the depiction of characteristics specific to *S. dumerili*. Furthermore, the introduction of a specialized detection head for small objects, which employs shallower, higher-resolution feature maps, represents a significant step forward. This head, in combination with the preexisting three heads, culminates in a four-head detection system. This comprehensive structure facilitates the concurrent detection of objects across multiple regions, thereby broadening the scope and speed of detection and curbing the occurrence of missed targets and false positives. This multifaceted approach signifies a substantial advancement in object detection technology, particularly for the accurate identification of small objects such as *S. dumerili* under challenging conditions.

2.3. C2f_DCN

In the YOLOv8 architecture, the C2f module plays a critical role in feature fusion by combining convolutional, splitting, and bottleneck blocks to extract and merge both low-level and high-level features from the input image. This fusion process capitalizes on detailed and semantic information to amalgamate feature maps at varying depths. A limitation of the traditional bottleneck block within the C2f module, as depicted in Figure 2a, is its reliance on fixed 3 × 3 convolutions at predetermined locations on the feature map, which is not optimal for object features that vary in position and scale. This paper introduces an enhancement to the conventional C2f module aimed at overcoming the challenges of scale variation in *S. dumerili*, such as nonlinear aberrations in underwater images, varying distances of the fish from the camera, and incomplete outlines of the fish. The innovation involves the substitution of the fixed 3 × 3 convolution within the bottleneck block with deformable convolution, as described in [32]. This alternative, as visualized in Figure 2b, introduces offset adjustments to standard convolutions, enabling deformable convolution to adapt more effectively to the distinct shapes and sizes of targets. The deformable convolution network v2 (DCNv2) further refines this approach by incorporating additional deformable layers, thus enhancing the convolution kernel's sampling capacity. Moreover, DCNv2 learns the sampling points' weight information in tandem with offset learning, which serves to substantially reduce convolutional sampling disruptions caused by extraneous elements. This learned weight information is integral in minimizing the influence of irrelevant factors during the sampling process. The mathematical representation of the weight information is encapsulated in the following equation:

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

where p_n denotes a predetermined offset, Δp_n represents the learned offset for the deformable convolution, and Δm_n is a learnable weight used for end-to-end training. The terms

$w(p_n)$, $x(p_0)(p_0)$, and $y(p_n)$ refer to the weights at position p_n. The pixel feature at position x is obtained from the input feature map, and the image feature at position p_0 is obtained from the output feature map.

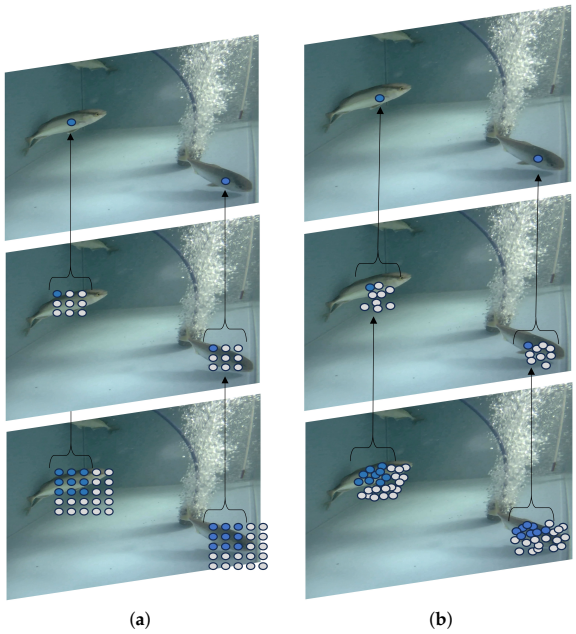


Figure 2. Visualization of standard convolution and deformable convolution. (**a**) Standard convolution with the fixed receptive field. (**b**) Deformable convolution with the adaptive receptive field.

2.4. Bottleneck Attention Module

This paper introduces the BAM to enhance the model's focus on *S. dumerili* features to mitigate interference from underwater bubbles, turbid water quality, and feed obstruction. The BAM decomposes the input image into spatial attention and channel attention components. The spatial attention component focuses on learning the content information of the image, while the channel attention component focuses on learning the positional information of the image. The structure of the BAM is shown in Figure 3. Spatial attention utilizes dilated convolutions to emphasize or suppress features at different positions, thereby expanding the receptive field and enhancing the ability to utilize contextual information, thus strengthening the spatial mapping capability. Channel attention adaptively adjusts the feature response of each channel by leveraging the relationships between channel branches. The calculation formula for the BAM is shown as follows:

$$\mathrm{M(F)}{=}M_s(\mathrm{F}) + M_c(\mathrm{F}), \tag{2}$$

where F represents the input feature map. $M_s(F)$ refers to the spatial attention processing result, $M_c(F)$ denotes the channel attention mechanism processing result, and $M(F)$ represents the overall BAM attention.

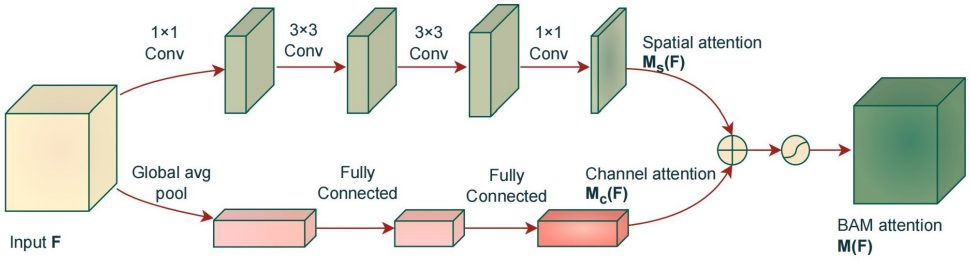


Figure 3. The structure of BAM.

2.5. SPPF_LSKA

YOLOv8 utilizes the fast spatial pyramid pooling (SPPF) technique, which applies a series of max pooling layers to different scale feature maps to generate fixed-sized vector outputs, addressing the issue of information distortion caused by traditional pooling methods. In traditional YOLOv8, the SPPF layer incorporates a 1×1 convolution for information fusion, achieving satisfactory results in tasks with a single background. However, this approach fails to balance the structural and channel adaptability of image information when dealing with complex backgrounds. Therefore, redesigning this component is of high importance for improving fish recognition capabilities.

Large separable kernel attention (LSKA) is introduced for reconstructing the SPPF layer, which is tailored specifically for vision tasks. A large-kernel convolution captures a wider range of contextual information, aiding the model in better learning object positions and relationships in the image. However, simply enlarging the convolution kernel leads to increased parameters and computational complexity, which is detrimental to model training. In light of this, large-kernel attention (LKA) decomposes large convolution kernels into a spatial local convolution (depthwise convolution), a spatial long-range convolution (depthwise dilated convolution), and a channel convolution. Specifically, the $K \times K$ convolution is reconfigured into a depthwise dilated convolution with a dilation factor of d, which results in a spatial kernel size of $\lfloor \frac{K}{d} \rfloor \times \lfloor \frac{K}{d} \rfloor$. This is followed by a depthwise convolution with a kernel size of $(2d-1) \times (2d-1)$, and subsequently, Sa 1×1 convolution is applied [33]. This decomposition strategy reduces the computational and parameter burdens associated with purely enlarging the convolution kernel. The convolution decomposition method is illustrated in Figure 4, and the LKA structure is shown in Figure 5a. The calculation of the LKA is shown as follows:

$$\bar{Z^c} = \sum_{h,w} W^c_{(2d-1)\times(2d-1)} * F^c, \tag{3}$$

$$Z^C = \sum_{H,W} W^C_{\lfloor \frac{k}{d} \rfloor \times \lfloor \frac{k}{d} \rfloor} * \bar{Z}^C, \tag{4}$$

$$A^C = W_{1\times1} * Z^C, \tag{5}$$

$$F^C = A^C \otimes F^C, \tag{6}$$

where $F \epsilon R^{c\times h\times w}$ represents the input feature map, c is the number of input channels, and h and w denote the height and width of the feature map, respectively. d represents the dilation rate, W^c represents the number of channels in the convolutional kernels, F^c represents the number of channels in the feature map, and $\bar{Z}^c$ represents the output of the depthwise convolution with the input feature map F using a kernel of size $(2d-1) \times (2d-1)$. This depthwise convolution captures local spatial information and compensates for the subsequent depth expansion convolution Z^C. k represents the kernel size, and Z^C is the output of the depth expansion convolution obtained by convolving $\bar{Z}^c$ with a kernel of size $\frac{k}{d} \times \frac{k}{d}$. The $\lfloor . \rfloor$ denotes the floor operation. A^C represents the attention map obtained by convolving the depth expansion convolution with a 1×1 kernel. The output $\bar{F}^C$ of the LKA is the Hadamard product (denoted by $\otimes$) of the attention map $A^C S$ and the input feature map F^C.

To further reduce computational complexity, LSKA splits the depthwise convolution layer and depth expansion convolution into two cascaded one-dimensional separable convolutions while preserving larger convolution kernels; the LSKA structure is shown in Figure 5b. The calculations for LSKA are shown as follows:

$$\bar{Z^c} = \sum_{h,w} W^c_{(2d-1)\times1} * \left(\sum_{h,w} W^c_{1\times(2d-1)} * F^C \right), \tag{7}$$

$$Z^C = \sum_{h,w} W^c_{\lfloor \frac{k}{d} \rfloor \times 1} * \left(\sum_{h,w} W^c_{1 \times \lfloor \frac{k}{d} \rfloor} * \bar{Z}^c \right), \tag{8}$$

$$A^C = W_{1\times 1} * Z^C, \tag{9}$$

$$\bar{F}^C = A^C \otimes F^C. \tag{10}$$

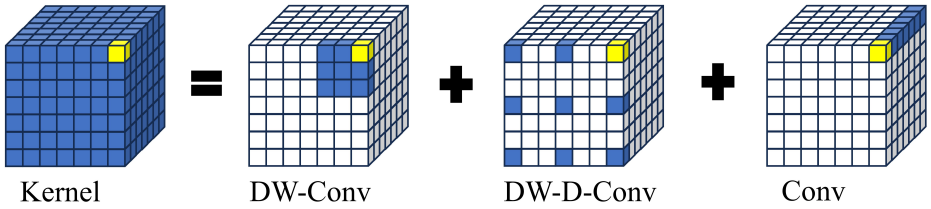
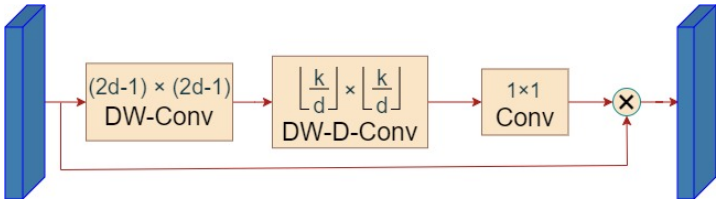
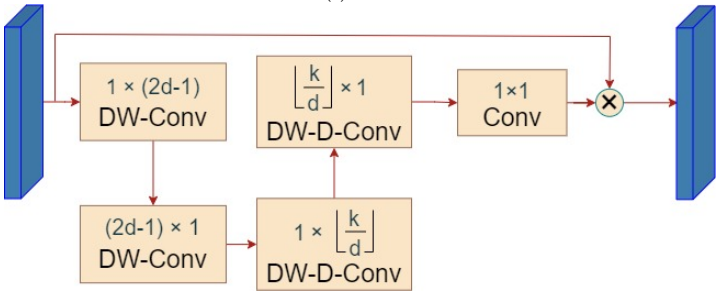


Figure 4. Illustration of large-kernel convolution. The colored grids represent the positions of the convolution kernels, while the yellow grids represent the central points of the grids. The process involves decomposing a 13 × 13 convolution into a 5 × 5 depthwise convolution, a 5 × 5 depthwise dilated convolution, and a 1 × 1 convolution.



(**a**) LKA.



(**b**) LSKA.

Figure 5. Visualization of kernel attention. (**a**) The LKA structures, and (**b**) the LSKA structures.

2.6. Inner-MDPIoU

The YOLOv8 loss function consists of two parts: classification loss and regression loss. The classification loss is calculated using binary cross-entropy loss, while the regression loss is calculated using a combination of distribution focal loss [34] and bounding box regression (BBR) [35]. However, since this study focuses only on classifying *S. dumerili*, which has only one category, the binary cross-entropy loss for that category is equal to 0. Therefore, the final loss function can be represented as follows:

$$f_{loss} = \lambda_1 f_{DFL} + \lambda_2 f_{BBR}, \tag{11}$$

The distribution focal loss (DFL) optimizes the probabilities of the left and right positions closest to the label y in the form of cross-entropy. This helps the network quickly focus on the distribution of the target position and its neighboring regions. It can be represented as follows:

$$f_{DFL}(S_i\ ,\ S_{i+1}) = -((y_{i+1} - y)\log(S_i) + (y - y_i)log(S_{i+1})), \tag{12}$$

where y_i and y_{i+1} represent the values approaching the continuous label y from the left and right sides, respectively, satisfying $y_i < y < y_{i+1}$. Additionally, the equation $y = \Sigma_{1=0}^{n} P(y_i) y_i$ describes the relationship between y and the probabilities $P(y_i)$, where $P(y_i)$ can be implemented using a softmax layer denoted as S_i in the formula.

During the data collection process, healthy fish were observed swimming actively in the water, and there was overlap and occlusion among the fish groups. Therefore, a loss function that can handle targets in a refined manner and robustly handle folding is needed. This study optimized the BBR loss function by using a new similarity metric called the Inner-MPDIoU metric to compute the IoU loss function by incorporating auxiliary bounding boxes. The Inner-MPDIoU calculation and parameters are shown in Figure 6. In the Inner-MPDIoU calculation, the MPDIoU component leverages the geometric properties of BBR to minimize the points distance d_1 and d_2 between the top-left and bottom-right corners of the predicted box and the ground-truth box. This measures the similarity between the predicted and ground-truth boxes, which can be adapted to overlapping or nonoverlapping bounding box regression. In the training phase, each bounding box $b = [x_c, y_c, w_{inner}, h_{inner}]$ predicted by the model is forced to approach its ground-truth box $b^{gt} = [x_c^{prd}, y_c^{prd}, w_{inner}^{gt}, h_{inner}^{gt}]$ by minimizing the loss function of the two points [36]. The Inner-IoU component adjusts the size of the auxiliary bounding boxes relative to the actual box using the scale ratio, which can control the scale size of the auxiliary bounding boxes [37]. Since the detection target in this experiment was fish, more detailed image information was needed. Therefore, the ratio was set to 1, meaning that the auxiliary box is smaller than the actual box. When the auxiliary box was smaller than the actual box, the effective range of the prediction was smaller than the IoU loss. However, the gradient magnitude obtained from the auxiliary box was greater than that from the IoU loss, accelerating the convergence of high IoU samples and obtaining more detailed image information by reducing the size of the ground-truth box. The combination of these two components robustly improved regression accuracy and convergence speed. The Inner-MPDIoU calculation formula can be represented as follows:

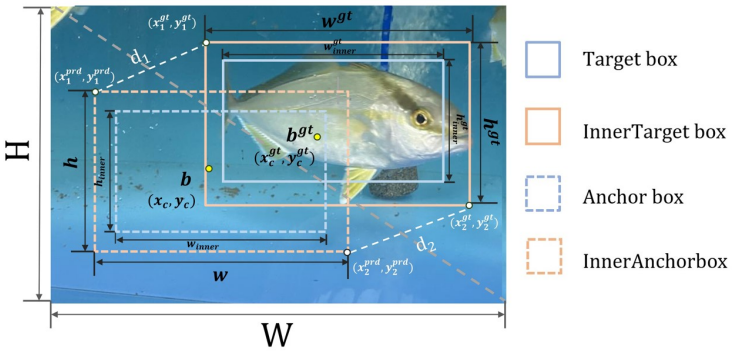


Figure 6. Calculation diagram of the Inner-MPDIou model.

$$d_1^2 = \left(x_1^{prd} - x_1^{gt}\right)^2 + \left(y_1^{prd} - y_1^{gt}\right)^2, \tag{13}$$

$$d_2^2 = \left(x_2^{prd} - x_2^{gt}\right)^2 + \left(y_2^{prd} - y_2^{gt}\right)^2, \tag{14}$$

$$IoU = \frac{Target\ box \cap Ground\ truth\ box}{Target\ box \cup Ground\ truth\ box}, \tag{15}$$

$$MPDIoU = IoU - \frac{d_1^2}{W^2 + H^2} - \frac{d_2^2}{W^2 + H^2}, \tag{16}$$

$$L_{MPDIoU} = 1 - MPDIoU, \tag{17}$$

where d_1 represents the distance between the top-left corners $\left(x_1^{prd}, y_1 prd\right)$ and $\left(x_1^{gt}, y1_g t\right)$ of the predicted bounding box and the ground-truth bounding box, respectively, while d_2 represents the distance between the top-right corners $\left(x_2^{prd}, y_2^p rd\right)$ and $\left(x_2^{gt}, y2gt\right)$. W and H represent the width and height of the input image, respectively.

$$b_l^{gt} = x_c^{gt} - \frac{w^{gt} * ratio}{2}, b_r^{gt} = x_c^{gt} + \frac{w^{gt} * ratio}{2}, \tag{18}$$

where B^{gt} and B represent the ground-truth bounding box and anchor box, respectively. $\left(x_c^{gt}, y_c^{gt}\right)$ represents the coordinates of the center point of the ground-truth box, while (x_c, y_c) represents the center point of the anchor box and the inner anchor box. W^{gt} and h^{gt} represent the width and height of the ground-truth box, respectively.

$$b_t^{gt} = y_c^{gt} - \frac{h^{gt} * ratio}{2}, b_b^{gt} = y_c^{gt} + \frac{h^{gt} * ratio}{2}, \tag{19}$$

$$b_l = x_c - \frac{w * ratio}{2}, b_r = x_c + \frac{w * ratio}{2}, \tag{20}$$

where w and h represent the height and width of the anchor box,respectively.

$$b_t = y_c - \frac{h * ratio}{2}, b_b = y_c + \frac{h * ratio}{2}, \tag{21}$$

$$inter = \left(\min\left(b_r^{gt}, b_r\right) - \max\left(b_l^{gt}, b_l\right)\right) * \left(\min\left(b_b^{gt}, b_b\right) - \max\left(b_t^{gt}, b_t\right)\right) \tag{22}$$

$$union = \left(w^{gt} * h^{gt}\right) * (ratio)^2 + (w * h) * (ratio)^2 - inter, \tag{23}$$

$$IoU^{inner} = \frac{inner}{union}, \tag{24}$$

$$L_{Inner-MPDIou} = L_{MPDIoU} + IoU - IoU^{inner}. \tag{25}$$

3. Materials

3.1. Experimental Environment and Parameter Settings

All experiments in this paper were conducted on the same computer with the following specifications: Windows 10 operating system, Intel® Xeon® Silver 4100 CPU, and NVIDIA GeForce RTX 2080 Ti GPU. PyTorch version 2.0.0 and CUDA version 11.7 were used. The experiments were conducted for 100 epochs with a batch size of 16 and a learning rate of 0.01 to evaluate the model performance. The results were optimized using the SGD optimizer.

3.2. Dataset

In order to evaluate the detection robustness of the SD-YOLOv8 model, we conducted experiments on two datasets: the *S. dumerili* dataset and a real-world dataset. Firstly, we trained the SD-YOLOv8 model using the *S. dumerili* dataset. This dataset provided a foundation for the model's initial training and performance assessment. To further validate the model's robustness, we constructed a separate dataset by gathering images from FishBase (https://www.fishbase.se, accessed on 21 April 2024) and the Aquarium dataset (https://public.roboflow.com/object-detection, accessed on 21 April 2024). The real-world dataset encompasses a broader spectrum of environmental conditions and object variations, thereby furnishing us with the means to rigorously evaluate the model's detection proficiencies across a plethora of settings. Crucially, this dataset enables an effective validation of SD-YOLOv8's capacity to discern and categorize entities beyond the realm of *S. dumerili* biology. By integrating this supplementary dataset, our objective is to augment the model's aptitude for precise detection and classification of objects, transcending the limitations imposed by the initial training dataset. In essence, the inclusion

of a diverse real-world dataset serves to fortify the robustness and versatility of the SD-YOLOv8 model. It allows for the assessment of the model's performance in scenarios that deviate from the controlled conditions of the original training environment. This is pivotal for ensuring that the model can generalize its learning to novel situations, thereby enhancing its applicability and reliability in practical, real-world applications.

3.2.1. *S. dumerili* Dataset

This paper compiled a dataset of *S. dumerili* images under laboratory conditions with variances in angles, lighting, and resolutions. Imaging data for *S. dumerili* were acquired from the Biological Breeding Center at the Southern Marine Science and Engineering Laboratory (ZHANJIANG) within recirculating aquaculture systems. These images were captured both above and beneath the water surface, under various lighting conditions spanning the early morning to the evening of a day. The dataset includes 1071 images, and it was segmented into training, validation, and testing subsets in a 7:2:1 ratio. Table 1 details the resolution of the capturing equipment, and Figure 7 shows the experimental capture angles and image collection methodology.

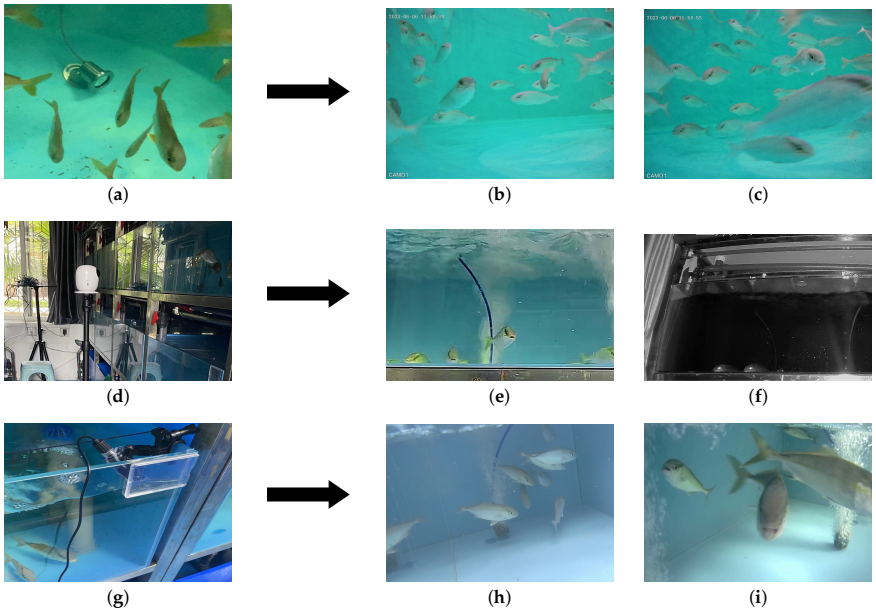


Figure 7. Images with different shooting angles. (Images (**a**,**d**,**g**) were taken by a Barlus underwater camera, a TP-Link network camera, and an SJCAM action camera, respectively. Images (**b**,**c**,**e**,**f**,**h**,**i**) are the captured images from the aforementioned cameras.)

Proper preprocessing of underwater datasets is crucial for correcting image color and enhancing image quality and clarity, which are essential for subsequent model training and prediction. Effective preprocessing steps ensure that the inherent distortions and variations in underwater images are mitigated, thereby providing a more reliable and accurate input for deep learning models [38]. Among the training set, there were 460 original images directly extracted from the captured videos and 289 images from applied augmentation software. Of the 289 images, there were 156 images with diverse lighting conditions and 133 images with blurry boundaries; we employed image augmentation software [39] to augment the training of these images. Among the augmentation methods used, we applied the sigmoid contrast technique [40] to enhance the contrast of low-light images, which rectified the images' color performance, improving their visibility and ensuring accurate object detection. Additionally, we utilized the canny method [41] to emphasize blurry boundaries, making them more distinct and aiding in the model's ability to accurately detect and delineate objects with unclear edges. These augmented images played a crucial role in training the SD-YOLOv8 model to handle a wide range of lighting conditions and object boundaries, ultimately enhancing its generalization capabilities. However, for the

validation and testing sets, no such augmentation was enacted, as their role was to gauge the model's ability with real-world applications.

Table 1. Shooting device information.

Device Name	Resolution		
Barlus Underwater Camera	5 MP	2592 × 1944	25 FPS
IPC5MPW-PBX10	4 MP	2560 × 1440	25 FPS
TP-Link Network Camera TL-IPC44B	4 MP	2560 × 1440	15 FPS
SJCAM Action Camera C300	4 K	3840 × 2160	30 FPS
	2 K	2560 × 1440	60/30 FPS
	720 P	1280 × 7201	20/60/30 FPS
	1080 P	1920 × 1080	120/60/30 FPS

To optimize the cost of manual labeling, we adopted a two-step approach for labeling our dataset. Initially, we leveraged the LabelImg [42] tool to manually label 200 images. These labeled images served as a foundation for training our model to detect and label the remaining images in an automated manner. Using the pretraining model, we applied automated labeling to the remaining images. Although this approach may introduce some inaccuracies in the initial labels, we subsequently conducted a thorough review of all labels. During this review process, we meticulously checked and modified any inaccurate labels, ensuring the highest level of accuracy and precision in the dataset. By combining manual labeling with automated labeling using the pretraining model, we were able to significantly reduce the amount of human effort and resources required for the dataset labeling process. This approach effectively balanced the need for accurate labeling while optimizing the time and manpower involved in the overall process.

3.2.2. Real-World Dataset

The real-world dataset was collected from FishBase and the Aquarium dataset. FishBase is an open-source fish database created and maintained by the Leibniz Institute of Oceanology, which provides researchers with comprehensive data on species, regional distribution, and population density [43]. We took the species of *S. dumerili* in our constructed dataset, which contained 96 wild *S. dumerili* images. The Aquarium dataset was collected by Roboflow from two aquariums in the United States, The Henry Doorly Zoo in Omaha (16 October 2020) and the National Aquarium in Baltimore (14 November 2020) [44], and is composed of 638 images that cover different classes of the following marine life: fish, jellyfish, penguins, sharks, puffins, stingrays, and starfish. The entire real-world dataset contains 735 images and covers eight classes, and it was split into a training set and a test set in an 8:2 ratio.

3.3. Evaluation Metrics

To evaluate the algorithm model in the experiments, performance metrics based on neural network models [45] are used in this paper. The precision, recall, F1 score, and average precision are adopted as evaluation metrics.

Precision represents the ratio of true-positive results to the total number of predicted samples. The formula is as follows:

$$Precision = \frac{TP}{TP + FP}, \tag{26}$$

where TP represents the number of actual *S. dumerili* samples correctly identified as *S. dumerili*, and FP represents the number of non-*S. dumerili* samples incorrectly identified as *S. dumerili*.

Recall represents the proportion of correctly identified samples to the total samples in the dataset. The formula is as follows:

$$Recall = \frac{TP}{TP + FN}, \tag{27}$$

where FN represents the number of actual *S. dumerili* incorrectly identified as *S. dumerili*. The $F1$ score is the weighted average of precision and recall and is a comprehensive metric for evaluating the accuracy and recall of a model in detection tasks. The F1 score ranges from 0 to 1, and a higher value indicates better model performance. When both the precision and recall are high, the F1 score is also high, indicating that the model performs well in the detection task. The formula is defined as follows:

$$F1 = \frac{2}{Recall^{-1} + Precision^{-1}} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{28}$$

In the context of *S. dumerili* detection, the average precision (AP) is calculated by taking the average precision at various recall levels, providing an overall assessment of the precision–recall trade-off. The mean average precision (mAP) represents the average AP for different categories. In this specific paper, as only *S. dumerili* is detected, the AP is equal to the mAP. The formulas for the AP and mAP are defined as follows:

$$AP = \int_0^1 p(R)\mathrm{d}R, \tag{29}$$

$$mAP = \frac{\sum_1^N \int_0^1 P(R)\mathrm{d}R}{N}, \tag{30}$$

where N represents the number of classes. When there is only one class, the AP is equal to the mAP.

4. Results and Analysis

4.1. Model Comparison

Different *S. dumerili* images with different backgrounds are selected to compare the improvement of the algorithm. The model detection comparison is shown in Figure 8. Figure 8 provides an illustrative comparison between the original YOLOv8n detection results (panels a, b, c, d, and e) and the proposed SD-YOLO method presented in our paper (panels f, g, h, i, and j). Specifically, the pairs of (a) and (f) and of (b) and (j) depict cases of missed detections, such as baits and splash bubbles, (c) and (h) show missed detection as well night detection performance, (d) and (i) demonstrate missed detections of small fishes, and (e) and (j) highlight fish body overlap. Due to the redesigned feature extraction and fusion blocks in the network, the model's ability to learn and generalize *S. dumerili* details is enhanced. The comparison images show that the proposed method improves the recognition ability of *S. dumerili* for complex backgrounds, especially for complex backgrounds and overlapping or occluded fish bodies.

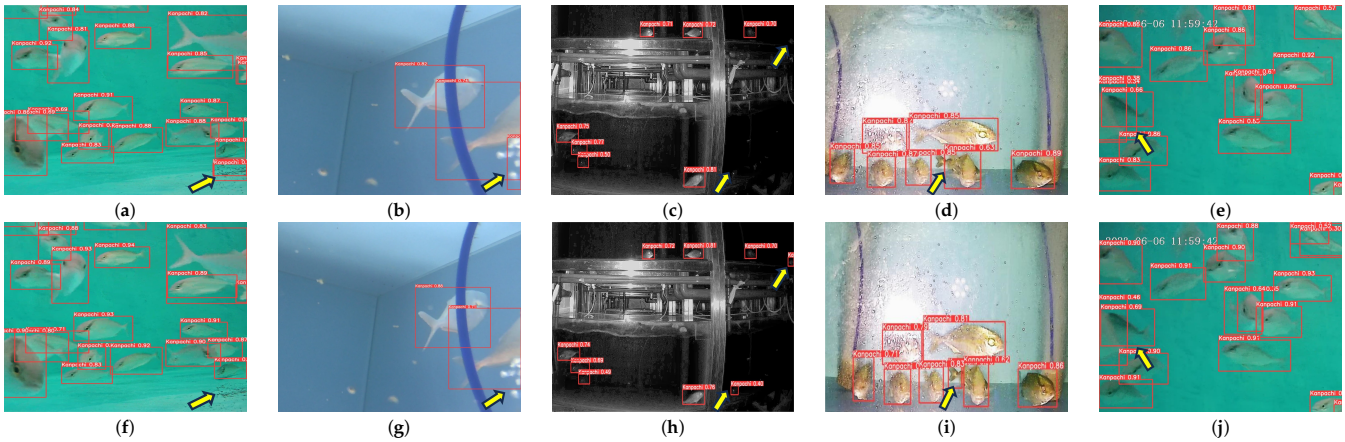


Figure 8. Comparison of the original network and improved results.The original YOLOv8n detection performance as (**a**–**e**) shown. SD-YOLOv8n detecion performance as (**f**–**j**) shown.

We conducted a comparative analysis of our proposed model against mainstream models, employing the S. dumerili annotated dataset as input for the Faster-RCNN [46], RetinaNet [47], SSD [48], YOLOv4-tiny, YOLOv5-s, YOLOX-s [49], YOLOv7, and YOLOv8n models. Our evaluation encompassed precision, recall, F1 score, $mAP_{@0.5}$, parameter count, computational complexity, and model size as performance metrics. Upon comparing our two-stage object detection model, SD-YOLOv8, with Faster R-CNN, CenterNet, and RetinaNet, it becomes apparent that SD-YOLOv8 surpasses the two-stage network in all metrics. In contrast to the one-stage object detection model, SSD, and other YOLO series models, our innovative model demonstrates notable improvements in precision. Specifically, it outperforms SSD by 8.9%, YOLOv4-tiny by 9.6%, YOLOv5 by 2.7%, YOLOX-s by 3.1%, YOLOv7 by 4.8%, and YOLOv8n by 4.1%. Moreover, it exhibits an 8.7%, 11.5%, 2.5%, 1.3%, 3.3%, and 3.5% increase in $mAP_{@0.5}$ compared to SSD, YOLOv4-tiny, YOLOv5, YOLOX-s, YOLOv7, and YOLOv8n, respectively. Furthermore, SD-YOLOv8n showcases the highest F1 score among both one-stage and two-stage models. In terms of model size, our proposed model is comparable to YOLOv5 and 1.6MB smaller than YOLOv8n, while significantly outperforming other models listed in Table 2 in terms of compactness. Additionally, Figure 9 shows the detection results of our experimentally improved model, which achieved a 4.1% improvement in accuracy compared to the YOLOv8n baseline model. It also yields higher values of precision, recall, F1 score, and $mAP_{@0.5}$ than SSD, YOLOv4-tiny, YOLOv5, YOLOX-s, and YOLOv7. In terms of parameter count, computational complexity, and model size, our proposed model competes with mainstream models.

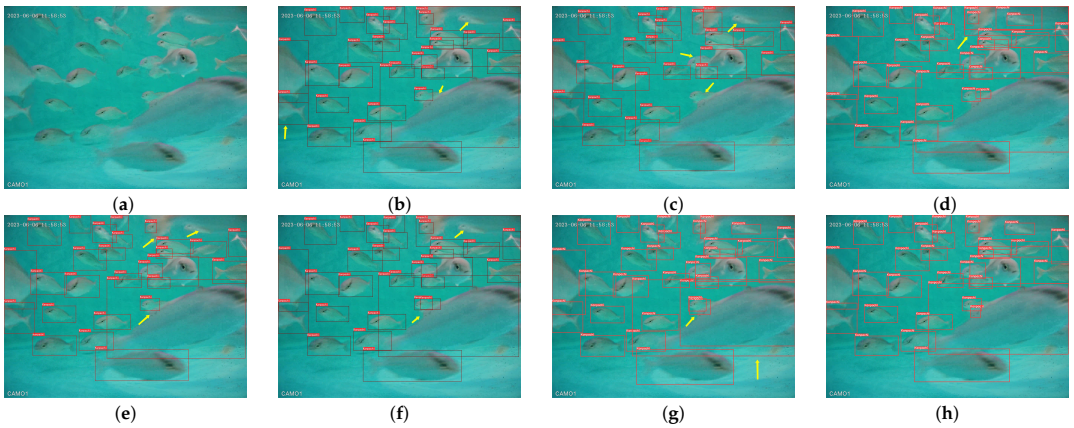
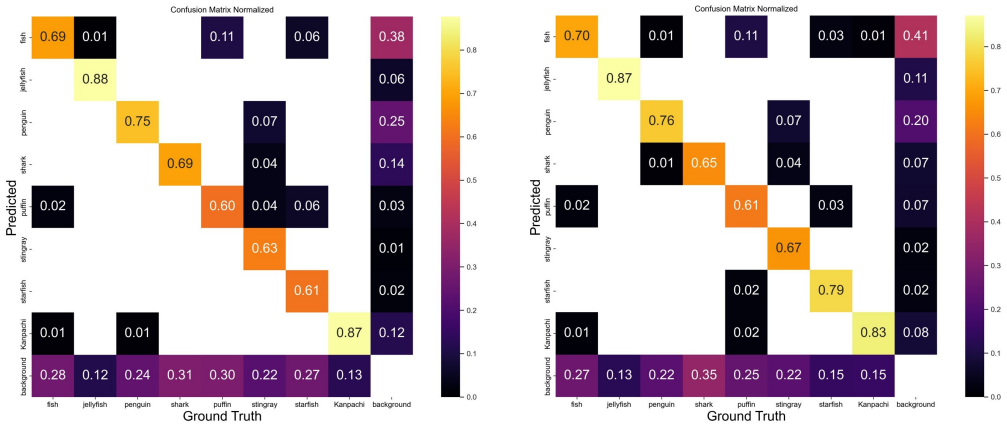


Figure 9. Comparison results of different detection model : (**a**) original picture; (**b**) SSD; (**c**) YOLOv4-tiny; (**d**) YOLOv5s; (**e**) YOLOX-s; (**f**) YOLOv7; (**g**) YOLOv8n; (**h**) SD-YOLOv8.

In order to further assess the performance of our model, we conducted experiments on a real-world dataset. The results of the confusion matrix are depicted in Figure 10. Upon examining Figure 10, it becomes evident that SD-YOLOv8 outperforms YOLOv8n in six classes, namely, fish, penguin, shark, puffin, starfish, and Kanpachi. On the other hand, YOLOv8n demonstrates superior capabilities in distinguishing jellyfish and stingrays. However, it is important to note that both models encounter challenges in accurately identifying multiple classes, resulting in occasional misclassifications into unrelated categories.



(**a**) (**b**)

Figure 10. Confusion matrices of SD-YOLOv8 and YOLOv8n on *S. dumerili* dataset and real-world dataset. SD-YOLOv8 (**a**) exhibits superior performance across six classes: fish, penguin, shark, puffin, starfish, and Kanpachi. YOLOv8n (**b**) demonstrates notable capabilities in distinguishing jellyfish and stingrays.

On the real-world dataset, as illustrated in Table 2, our SD-YOLOv8 model achieves the highest precision of 81.6%, with YOLOX-s closely following at a mere 0.1% lower. However, when it comes to the recall metric, Faster RCNN outperforms all other models with a recall rate of 75.4%. CenterNet, SD-YOLOv8, YOLOv5, YOLOv8n, RetinaNet, SSD, and YOLOv4-tiny exhibit comparable recall metrics, all hovering around 65%. YOLOX-s, on the other hand, demonstrates a lower recall rate. In terms of F1 score, SD-YOLOv8, YOLOv8n, and Faster RCNN showcase the top performances, reaching 73.1%, 71.5%, and 70.6%, respectively. In the $mAP_{@0.5}$ metric, CenterNet outperforms other models by achieving a detection accuracy improvement of 4.7% over Faster RCNN, 19.5% over RetinaNet, 4.4% over SSD, 9.8% over YOLOv4-tiny, 1.2% over YOLOv5, 6.4% over YOLOX-s, 12.1% over YOLOv7, 6.7% over YOLOv8n, and 2.3% over SD-YOLOv8. When considering the parameters metric, YOLOv8n stands out as the most lightweight model, with only 3.1 M parameters. SD-YOLOv8 remains the second smallest model among the compared detection models. In terms of floating point operations per second (FLOPs), the larger-scale detection model Faster RCNN requires 370.2 G FLOPs, similar to its performance on the *S. dumerili* dataset. YOLOv5 and YOLOv4-tiny have computational costs of 6.1 G and 6.5 G FLOPs, respectively, which are still considered low on real-world datasets. With regard to model size, both our proposed SD-YOLOv8 and YOLOv8n exhibit the smallest sizes, weighing in at a mere 7.6 MB. Overall, when compared to different object detection models, SD-YOLOv8 continues to outperform other models on the real-world dataset. Figure 11 presents the adeptness of SD-YOLOv8 in discerning relevant subjects within a real-world dataset. The visuals collectively corroborate the model's proficiency in reliably distinguishing and excluding non-*S. dumerili* marine entities. For instance, panels (b) and (e) elucidate the model's capabilities in identifying objects across varying scales with a notable emphasis on the detection of diminutive objects. Meanwhile, panels (c) and (f) exemplify the model's resilience against environmental perturbations such as light reflections while maintaining its targeting accuracy. These instances are demonstrative of SD-YOLOv8's formidable object detection competencies in complex, real-world scenarios.

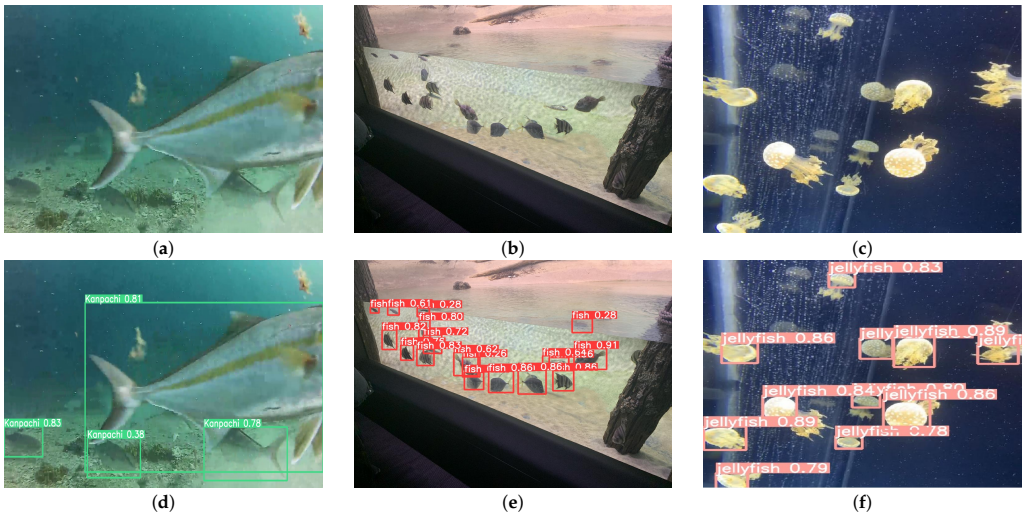


Figure 11. Detection results of SD-YOLOv8 on the real-world dataset.Image (**a**–**c**) are the original images, while images (**d**–**f**) display the SD-YOLOv8 detection results. Among these results, (**d**) demonstrates the capability in detecting wild *S. dumerili*, (**e**) showcases the ability to identify objects across varying scales with a notable emphasis on the detection of diminutive objects, and (**f**) illustrates the model's resilience against environmental perturbations such as light reflections while maintaining its targeting accuracy.

Table 2. Comparison of our proposed model with mainstream models.

Dataset	Models	Precision	Recall	F1 Score	mAP$_{@0.5}$	Parameters	FLOPs	Size
S. dumerili	Faster RCNN	77.2%	78.4%	77.8%	82.3%	137.1 M	370.2 G	108.0 MB
	CenterNet	79.2%	79.4%	79.3%	80.1%	32.7 M	70.2 G	124.0 MB
	RetinaNet	81.0%	60.5%	69.3 %	56.2%	37.9 M	170.1 G	108.0 MB
	SSD	84.4%	89.5%	86.9%	87.0%	18.4 M	15.5 G	90.6 MB
	YOLOv4-tiny	83.7%	73.4%	80.3%	84.2%	6.1 M	**6.9 G**	22.4 MB
	YOLOv5s	90.6%	87.8%	89.2%	93.3%	**2.6 M**	7.7 G	7.7 MB
	YOLOX-s	90.2%	89.6%	89.9%	94.4%	8.9 M	26.6 G	34.3 MB
	YOLOv7	88.5%	89.0%	88.8%	92.4%	37.2 M	105.1 G	74.3 MB
	YOLOv8n	89.2%	88.4%	88.8%	92.2%	3.1 M	8.1 G	**6.0 MB**
	SD-YOLOv8	**93.3%**	88.9%	**91.0%**	**95.7%**	3.5 M	12.7 G	7.6 MB
Real-world	Faster RCNN	66.4%	**75.4%**	70.6%	71.5%	137.1 M	370.2 G	108.0 MB
	CenterNet	67.8%	66.7%	67.2%	**76.2%**	32.7 M	70.2 G	124.0 MB
	RetinaNet	62.4%	65.0%	63.7%	56.7%	37.9 M	170.1 G	108.0 MB
	SSD	70.6%	64.9%	67.6%	71.8%	26.3 M	62.7 G	94.1 MB
	YOLOv4-tiny	75.3%	61.5%	67.7%	66.4%	6.4 M	**6.5 G**	22.4 MB
	YOLOv5s	70.4%	66.2%	68.2%	75.0%	6.1 M	6.9 G	27.2 MB
	YOLOX-s	81.5%	54.7%	65.5%	69.8%	8.9 M	26.8 G	34.3 MB
	YOLOv7	70.0%	64.8%	67.3%	64.1%	37.6 M	106.5 G	142.0 MB
	YOLOv8n	78.5%	65.7%	71.5%	69.5%	**3.1 M**	8.1 G	7.6 MB
	SD-YOLOv8	**81.6%**	66.5%	**73.1%**	73.9%	3.7 M	12.2 G	**7.6 MB**

4.2. Ablation Experiments

In terms of modules, this study conducted nine ablation experiments to test their performance on the validation set in terms of precision, recall, F1 score, $mAP_{@0.5}$, and $mAP_{@0.5:0.95}$. As shown in Table 3, where YOLOv8n serves as the baseline model, "structure" represents the performance of the model after modifying the network structure. Subsequent experiments gradually added DCNv2, the BAM attention mechanism, the SPPF_LSKA module, and combinations of the two modules to the modified network to evaluate the performance of the models. The improved model achieved increases in accuracy, F1 score, $mAP_{@0.5}$, and $mAP_{@0.5:0.95}$. Specifically, as the accuracy increased by 4.1 percentage points, the F1 score increased by 2.1 percentage points, the $mAP_{@0.5}$ increased by 3.2 percentage points, and the $mAP_{@0.5:0.95}$ increased by 3.2 percentage points.

Upon conducting a more nuanced analysis, it becomes evident that the integration of various modules yielded substantial enhancements to the baseline model. While the C2f_DCN and SPPF_LSKA modules demonstrated remarkable contributions, it is important to acknowledge that other improvement options should not be overlooked. The C2f_DCN module showcased considerable potential in augmenting recall, $mAP_{@0.5}$, and $mAP_{@0.5:0.95}$ metrics. This can be primarily attributed to the utilization of deformable convolutions, which empower the model to selectively capture comprehensive object features within the input images. By adopting this approach, the C2f_DCN module successfully surpasses the limitations of traditional convolutional methods, ultimately leading to improved performance across multiple evaluation criteria. Conversely, the SPPF_LSKA module demonstrated its prowess in precision and the F1 score metrics. By incorporating downsampling layers with larger strides, this module excels in capturing intricate details with finesse, thereby refining the model's ability to discern objects with exceptional accuracy. The larger strides play a pivotal role in expanding the receptive field of the network, facilitating a more holistic understanding of the underlying features and enabling precise object localization. Although the C2f_DCN and SPPF_LSKA modules made significant contributions to the performance of SD-YOLOv8, it is essential to recognize that the overall improvements are not solely attributed to these modules. Other enhancement options should not be disregarded, as they may have also played vital roles in refining the model's capabilities. The intricate interplay of various modules, along with diligent fine-tuning and optimization, collectively contribute to the notable performance advancements witnessed in SD-YOLOv8.

Table 3. Effects of module ablations on the experimental results.

Models	Precision	Recall	F1 Score	$mAP_{@0.5}$	$mAP_{@0.5:0.95}$
YOLOv8n	89.2%	88.4%	88.8%	92.5%	66.5%
YOLOv8n + Structure	90.2%	88.5%	89.4%	94.4%	67.9%
Structure + C2f_DCN	90.1%	88.8%	89.4%	94.8%	68.9%
Structure + BAM	89.0%	88.7%	88.8%	94.6%	67.6%
Structure + SPPF_LSKA	91.1%	88.0%	89.5%	94.6%	67.2%
Structure + C2f_DCN + BAM	89.6%	87.6%	88.6%	94.6%	68.2%
Structure + C2f_DCN + SPPF_LSKA	90.6%	87.2%	88.9%	94.4%	68.1%
Structure + BAM + SPPF_LSKA	90.8%	87.0%	88.9%	94.4%	67.7%
Structure + C2f_DCN + BAM + SPPF_LSKA (Ours)	**93.3%**	**88.9%**	**91.0%**	**95.7%**	**69.7%**

In the loss function section, six sets of experiments were designed to assess the impact of various loss functions on model performance. The traditional intersection over union (IoU) algorithm lacks sensitivity to the proximity of overlapping bounding boxes, which complicates the calculation of loss and the execution of gradient backpropagation. The generalized intersection over union (GIoU) [50] introduces a minimum enclosing box as a penalty term, enhancing the gradient optimization of the traditional IoU. However, the convergence with the GIoU is slower, indicating potential for progress. Consequently,

the CIoU refines the penalty by incorporating aspects such as bounding box overlap, center distance, and aspect ratio, which facilitates faster convergence and elevates detection efficiency. Nevertheless, the CIoU can be further optimized, particularly by refining aspect ratio differences under comparable conditions. The efficient intersection over union (EIoU) builds on the CIoU by dissecting the factors affecting the aspect ratio, refining the calculation of the aspect ratio loss, and rectifying the imbalance of challenging samples. The scale-IoU (SIoU) metric [51] is a novel approach for loss computations that incorporates the angle, distance, and shape between the actual and predicted boxes, notably for detecting small objects, and significantly improves the resolution of orientation mismatches. MPDIoU, a regression loss function grounded on minimum point distance, addresses the issue of disparate values with identical aspect ratios between the actual and predicted boxes by measuring the distance along two aligned axes, advancing model performance in overlapping scenarios. Despite these advancements, most IoU-based methods have focused on innovating new loss functions to quicken convergence and enhance model proficiency, often neglecting the intrinsic characteristics of the IoU. In response, the inner-IoU metric is introduced, which refines the scaling of anchor boxes by implementing a scale factor, thus bolstering the model's adaptability across various detectors and detection tasks. This research integrates the Inner-IoU metric, which concentrates on anchor boxes, with the MPDIoU metric, which refines the loss term for boundary boxes. The combined Inner-MPDIoU loss function for bounding boxes significantly improves the efficiency of *S. dumerili* detection. As detailed in Table 4, the Inner-MPDIoU achieves substantial enhancements in metrics such as precision, recall, and F1 score.

Table 4. Effects of loss function ablations on the experimental results.

Method	Precision	Recall	F1 Score	$mAP_{@0.5}$	$mAP_{@0.5:0.95}$
GIoU	91.7%	89.5%	90.4%	94.7%	69.6%
CIoU	90.5%	87.6%	89.0%	94.3%	68.5%
EIoU	91.0%	89.5%	90.2%	94.8%	68.7%
SIoU	90.7%	88.5%	89.4%	94.8%	68.7%
MPDIoU	91.4%	87.1%	89.2%	94.6%	68.6%
InnerIoU	91.8%	**89.8%**	90.8%	95.3%	69.1%
Inner-MPDIoU (Ours)	**93.3%**	88.9%	**91.0%**	**95.7%**	**69.7%**

5. Conclusions

This study created an *S. dumerili* dataset with multiple angles, multiple light sources, and multiple resolutions, providing data support for accurate recognition of *S. dumerili*. The SD-YOLOv8 model was improved by modifying the network structure and adding a small object detection layer and detection head to enhance the efficiency of identifying small fish objects. The DCNv2 fusion with C2f with deformable convolution was introduced into the backbone network to enhance the model's ability to handle complex information and improve its resistance to interference, while also improving the extraction of fine-scale fish body features. The bottleneck attention module (BAM) was introduced to focus on both spatial and channel information, further enhancing the ability to acquire information on *S. dumerili*. The large separable kernel attention (LSKA) algorithm was used to fuse the spatial pyramid pooling fast (SPPF) algorithm for multiscale feature fusion in the backbone network. Finally, the Inner-MPDIoU metric was used to achieve rapid convergence and regression of the bounding box loss function. The experimental results showed that the improved YOLOv8n model achieved an increase in accuracy from 89.2% to 93.3% and an increase in average detection accuracy from 92.2% to 95.7%, corresponding to increases of 4.1 points and 3.5 points, respectively. The real-world dataset further validated the performance in scenarios that deviate from the controlled conditions of the original training environment.

This study preliminarily established a recognition model for *S. dumerili* in complex environments. In future research, we plan to collect more images under varied environ-

mental conditions and implement tailored preprocessing methods to enhance the dataset's robustness and applicability. Furthermore, adding more detection of other underwater objects also has great significance in enhancing the model's comprehensive ability.

Author Contributions: Conceptualization, M.L.; methodology, M.L., R.L. and M.H.; software, M.L., R.L., M.H., C.Z., J.H. and Y.W.; formal analysis, M.L. and C.Z.; investigation, M.L., R.L. and J.H.; resources, M.L., R.L., M.H. and C.Z.; data curation, M.L., R.L., M.H. and J.H.; writing—original draft, M.L., R.L., M.H., C.Z. and Y.W.; writing—review and editing, M.L., R.L. and C.Z.; visualization, M.L. and R.L.; supervision, M.H. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by the National Natural Science Foundation of China (62171143), the Natural Science Foundation of Guangdong Province (2021A1515011948), special projects in key fields of ordinary universities in Guangdong Province (2021ZDZX1060) and the Guangxi Key Research and Development Plan Project (2022AB20112), and the Fund of Guangdong Provincial Key Laboratory of Intelligent Equipment for South China Sea Marine Ranching (2023B1212030003).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Conflicts of Interest: The authors declare that there are no conflict of interest.

References

1. Shi, H.; Li, J.; Li, X.; Ru, X.; Huang, Y.; Zhu, C.; Li, G. Survival pressure and tolerance of juvenile greater amberjack (Seriola dumerili) under acute hypo- and hyper-salinity stress. *Aquac. Rep.* **2024**, *36*, 102150. [CrossRef]
2. Corriero, A.; Wylie, M.J.; Nyuji, M.; Zupa, R.; Mylonas, C.C. Reproduction of greater amberjack (Seriola dumerili) and other members of the family Carangidae. *Rev. Aquac.* **2021**, *13*, 1781–1815. [CrossRef]
3. Tone, K.; Nakamura, Y.; Chiang, W.C.; Yeh, H.M.; Hsiao, S.T.; Li, C.H.; Komeyama, K.; Tomisaki, M.; Hasegawa, T.; Sakamoto, T.; et al. Migration and spawning behavior of the greater amberjack Seriola dumerili in eastern Taiwan. *Fish. Oceanogr.* **2022**, *31*, 1–18. [CrossRef]
4. Rigos, G.; Katharios, P.; Kogiannou, D.; Cascarano, C.M. Infectious diseases and treatment solutions of farmed greater amberjack Seriola dumerili with particular emphasis in Mediterranean region. *Rev. Aquac.* **2021**, *13*, 301–323. [CrossRef]
5. Sinclair, C. *Dictionary of Food: International Food and Cooking Terms from A to Z*; A&C Black: London, UK, 2009.
6. Li, D.; Du, L. Recent advances of deep learning algorithms for aquacultural machine vision systems with emphasis on fish. *Artif. Intell. Rev.* **2022**, *55*, 4077–4116. [CrossRef]
7. Yang, L.; Liu, Y.; Yu, H.; Fang, X.; Song, L.; Li, D.; Chen, Y. Computer vision models in intelligent aquaculture with emphasis on fish detection and behavior analysis: a review. *Arch. Comput. Methods Eng.* **2021**, *28*, 2785–2816. [CrossRef]
8. Islam, S.I.; Ahammad, F.; Mohammed, H. Cutting-edge technologies for detecting and controlling fish diseases: Current status, outlook, and challenges. *J. World Aquac. Soc.* **2024**, *55*, e13051. [CrossRef]
9. Fayaz, S.; Parah, S.A.; Qureshi, G. Underwater object detection: Architectures and algorithms–a comprehensive review. *Multimed. Tools Appl.* **2022**, *81*, 20871–20916. [CrossRef]
10. Li, J.; Xu, W.; Deng, L.; Xiao, Y.; Han, Z.; Zheng, H. Deep learning for visual recognition and detection of aquatic animals: A review. *Rev. Aquac.* **2023**, *15*, 409–433. [CrossRef]
11. Lin, C.; Qiu, C.; Jiang, H.; Zou, L. A Deep Neural Network Based on Prior-Driven and Structural Preserving for SAR Image Despeckling. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 6372–6392. [CrossRef]
12. Li, X.; Shang, M.; Hao, J.; Yang, Z. Accelerating fish detection and recognition by sharing CNNs with objectness learning. In Proceedings of the OCEANS 2016—Shanghai, Shanghai, China, 10–13 April 2016; pp. 1–5. [CrossRef]
13. Boom, B.J.; Huang, P.X.; He, J.; Fisher, R.B. Supporting ground-truth annotation of image datasets using clustering. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 1542–1545.
14. Li, J.; Xu, C.; Jiang, L.; Xiao, Y.; Deng, L.; Han, Z. Detection and Analysis of Behavior Trajectory for Sea Cucumbers Based on Deep Learning. *IEEE Access* **2020**, *8*, 18832–18840. [CrossRef]
15. Shah, S.Z.H.; Rauf, H.T.; IkramUllah, M.; Khalid, M.S.; Farooq, M.; Fatima, M.; Bukhari, S.A.C. Fish-Pak: Fish species dataset from Pakistan for visual features based classification. *Data Brief* **2019**, *27*, 104565. [CrossRef] [PubMed]
16. Fouad, M.M.M.; Zawbaa, H.M.; El-Bendary, N.; Hassanien, A.E. Automatic Nile Tilapia fish classification approach using machine learning techniques. In Proceedings of the 13th International Conference on Hybrid Intelligent Systems (HIS 2013), Gammarth, Tunisia, 4–6 December 2013; pp. 173–178. [CrossRef]

17. Ravanbakhsh, M.; Shortis, M.R.; Shafait, F.; Mian, A.; Harvey, E.S.; Seager, J.W. Automated Fish Detection in Underwater Images Using Shape-Based Level Sets. *Photogramm. Rec.* **2015**, *30*, 46–62. [CrossRef]
18. Iscimen, B.; Kutlu, Y.; Uyan, A.; Turan, C. Classification of fish species with two dorsal fins using centroid-contour distance. In Proceedings of the 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, Turkey, 16–19 May 2015; pp. 1981–1984. [CrossRef]
19. Cutter, G.; Stierhoff, K.; Zeng, J. Automated Detection of Rockfish in Unconstrained Underwater Videos Using Haar Cascades and a New Image Dataset: Labeled Fishes in the Wild. In Proceedings of the 2015 IEEE Winter Applications and Computer Vision Workshops, Waikoloa, HI, USA, 6–9 January 2015; pp. 57–62. [CrossRef]
20. Dhawal, R.S.; Chen, L. A copula based method for the classification of fish species. *Int. J. Cogn. Inform. Nat. Intell. (IJCINI)* **2017**, *11*, 29–45. [CrossRef]
21. Li, X.; Shang, M.; Qin, H.; Chen, L. Fast accurate fish detection and recognition of underwater images with Fast R-CNN. In Proceedings of the OCEANS 2015—MTS/IEEE Washington, Washington, DC, USA, 19–22 October 2015; pp. 1–5. [CrossRef]
22. Salman, A.; Siddiqui, S.A.; Shafait, F.; Mian, A.; Shortis, M.R.; Khurshid, K.; Ulges, A.; Schwanecke, U. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. *ICES J. Mar. Sci.* **2019**, *77*, 1295–1307. [CrossRef]
23. Lin, W.H.; Zhong, J.X.; Liu, S.; Li, T.; Li, G. ROIMIX: Proposal-Fusion Among Multiple Images for Underwater Object Detection. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 2588–2592. [CrossRef]
24. Jalal, A.; Salman, A.; Mian, A.; Shortis, M.; Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. *Ecol. Inform.* **2020**, *57*, 101088. [CrossRef]
25. 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]
26. Liu, Y.; Chu, H.; Song, L.; Zhang, Z.; Wei, X.; Chen, M.; Shen, J. An improved tuna-YOLO model based on YOLO v3 for real-time tuna detection considering lightweight deployment. *J. Mar. Sci. Eng.* **2023**, *11*, 542. [CrossRef]
27. Hu, J.; Zhao, D.; Zhang, Y.; Zhou, C.; Chen, W. Real-time nondestructive fish behavior detecting in mixed polyculture system using deep-learning and low-cost devices. *Expert Syst. Appl.* **2021**, *178*, 115051. [CrossRef]
28. Zhou, S.; Cai, K.; Feng, Y.; Tang, X.; Pang, H.; He, J.; Shi, X. An Accurate Detection Model of Takifugu rubripes Using an Improved YOLO-V7 Network. *J. Mar. Sci. Eng.* **2023**, *11*, 1051. [CrossRef]
29. 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 (CVPR), Long Beach, CA, USA, 15–20 June 2019.
30. Lau, K.W.; Po, L.M.; Rehman, Y.A.U. Large Separable Kernel Attention: Rethinking the Large Kernel Attention design in CNN. *Expert Syst. Appl.* **2024**, *236*, 121352. [CrossRef]
31. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
32. 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 (ICCV), Venice, Italy, 22–29 October 2017.
33. Guo, M.H.; Lu, C.Z.; Liu, Z.N.; Cheng, M.M.; Hu, S.M. Visual attention network. *Comput. Vis. Media* **2023**, *9*, 733–752. [CrossRef]
34. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; Yang, J. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Glasgow, UK, 2020; Volume 33, pp. 21002–21012.
35. 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]
36. Siliang, M.; Yong, X. MPDIoU: A loss for efficient and accurate bounding box regression. *arXiv* **2023**, arXiv: 2307.07662.
37. Zhang, H.; Xu, C.; Zhang, S. Inner-IoU: More Effective Intersection over Union Loss with Auxiliary Bounding Box. *arXiv* **2023**, arXiv: 2311.02877.
38. Maharana, K.; Mondal, S.; Nemade, B. A review: Data pre-processing and data augmentation techniques. *Glob. Transitions Proc.* **2022**, *3*, 91–99. [CrossRef]
39. Fafa, D.L. Image-Augmentation. Available online: https://github.com/Fafa-DL/Image-Augmentation (accessed on 21 April 2024).
40. Hassan, N.; Akamatsu, N. A new approach for contrast enhancement using sigmoid function. *Int. Arab J. Inf. Technol.* **2004**, *1*, 221–225.
41. Ali, M.; Clausi, D. Using the Canny edge detector for feature extraction and enhancement of remote sensing images. In Proceedings of the IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217), Sydney, Australia, 9–13 July 2001; Volume 5, pp. 2298–2300.
42. HumanSignal. LabelImg. Available online: https://github.com/HumanSignal/labelImg (accessed on 21 April 2024).
43. Wang, F.; Zheng, J.; Zeng, J.; Zhong, X.; Li, Z. S2F-YOLO: An Optimized Object Detection Technique for Improving Fish Classification. *J. Internet Technol.* **2023**, *24*, 1211–1220. [CrossRef]

44. Karthi, M.; Muthulakshmi, V.; Priscilla, R.; Praveen, P.; Vanisri, K. Evolution of YOLO-V5 Algorithm for Object Detection: Automated Detection of Library Books and Performace validation of Dataset. In Proceedings of the 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 24–25 September 2021; pp. 1–6. [CrossRef]
45. Goutte, C.; Gaussier, E. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In *Advances in Information Retrieval*; Losada, D.E., Fernández-Luna, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 345–359.
46. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
47. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *arXiv* **2018**. [CrossRef]
48. 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, Proceedings of the 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016, Proceedings, Part I 14*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
49. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. YOLOX: Exceeding YOLO Series in 2021. *arXiv* **2021**. [CrossRef]
50. 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.
51. Gevorgyan, Z. SIoU Loss: More Powerful Learning for Bounding Box Regression. *arXiv* **2022**. [CrossRef]