ITC 4/53

Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 1291-1310
DOI 10.5755/j01.itc.53.4.38761

Fish Catch Sorting and Detection Model Improved Based on YOLOv8 Model

Received 2024/09/10 | Accepted after revision 2024/12/02

HOW TO CITE: Yang, P., Shi, T., Yuan, Y., Jiang, H. (2024). Fish Catch Sorting and Detection Model Improved Based on YOLOv8 Model. *Information Technology and Control*, 53(4), 1291-1310. https://doi.org/10.5755/j01.itc.53.4.38761

Fish Catch Sorting and Detection Model Improved Based on YOLOv8 Model

Ping Yang, Tiange Shi, Youdong Yuan, Hanbing Jiang

School of Mechanical and Electrical Engineering, Lanzhou University of Technology, Lanzhou 730050, China; e-mails: YANGping_lz@163.com (Yang); 222085501039@lut.edu.cn (Shi); 222080204006@lut.edu.cn (Yuan); 222080204004@lut.edu.cn (Jiang)

Ping Yang and Tiange Shi contributed equally to this work, they are both first authors.

Corresponding author: Tiange Shi, e-mail: 222085501039@lut.edu.cn

The primary challenge in trawl fishing lies in its limited selectivity, resulting in highly diverse fish catches with severe mixed-species issues. The catches from trawl fishing require manual sorting, leading to low work efficiency and high labor demands. In order to tackle this issue, the present study introduces an enhanced version of the DWR module by utilizing DilatedReparamBlock convolution, introducing a novel dilated convolution module. This module is integrated into the YOLOv8 model, enhancing its capacity to capture features from the enlarged receptive field in the upper network layers. Furthermore, a new attention mechanism based on a multi-branch structure with the CA attention mechanism is incorporated into the YOLOv8 model. This attention mechanism fully extracts image features, enhances feature representation, strengthens the generation of offsets and sampling weights, and improves the accuracy of target recognition. Lightweight improvements to the detection head are achieved through the use of shared convolutions, ultimately resulting in a significant reduction in the number of model parameters. Our empirical findings indicate that the refined model exhibits a 2.2% enhancement in mAP@0.5 when benchmarked against the initial YOLOv8 model, Offering a significant reference for the advancement of an effective embedded system for fish sorting.

KEYWORDS: fish catch sorting, object detection, YOLOv8 model, improved YOLOv8 model.

1. Introduction

Presently, trawling stands as the dominant marine fishing technique, comprising around 60% of the entire marine catch. The catch of trawling is extremely diverse, encompassing not only various bottom-dwelling fish but also mid-water and surface fish, crustaceans, cephalopods, shellfish, and more.

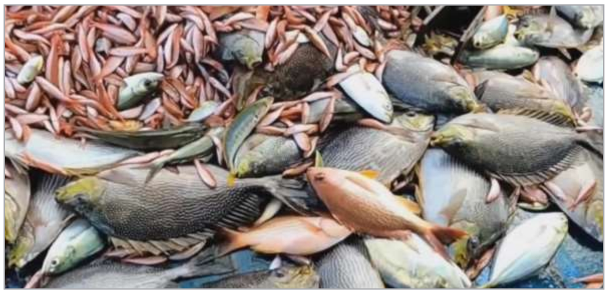
Additionally, the composition of catches varies significantly across different marine regions [1,16]. The most significant issue with trawling is its low selectivity, which results in a highly diverse catch composition and severe mixed-species catches. The harvested catches require manual sorting [28], and the process of sorting fish catches often consumes considerable time and labor.

Deep learning techniques have found extensive application in item detection assignments, including research focused on the classification and detection of fish, shellfish, and crayfish, as exemplified by studies conducted by Villon et al. [24] used a Convolutional Neural Network (CNN) to detect underwater fish, achieving an accuracy rate of 94.6%, which is greater than the hand-operated detection rate of 89.3%. Cui et al. [2] applied deep learning techniques to the detection of fish in murky seawater and established a detection model suitable for underwater fish. Feng et al. [4] employed an enhanced Faster RCNN for the purposes of classifying and detecting shellfish, enabling it to detect shellfish in different scenarios with an identification accuracy nearly 4% higher than the original model. Clearly, deep learning-based object detection algorithms have attained satisfactory outcomes in the domains of classification and grading recognition. In contrast, research on applying deep learning algorithms to fish catch sorting and detection remains a gap. After investigation, it is found that the industry currently lacks automatic and efficient fish catch sorting equipment and related research based on deep learning networks.

A central challenge in object detection lies in accurately identifying multiple objects of various categories within an image, while also providing precise location information pertaining to their bounding boxes [29]. Previously, object detection primarily relied on handcrafted features and conventional machine learning techniques, including Support Vector Machines (SVM) [13] and Haar cascades [9]. Nonetheless, advancements in deep learning have led to notable achievements in object detection, especially following the introduction of algorithms such as Faster RCNN [19], which was proposed by Ren et al., and the SSD algorithm proposed by Liu et al. [15]. These advancements have attracted considerable attention, owing to their remarkable recognition outcomes on publicly available datasets.

As shown in Figure 1, the fish catch dataset is significantly different from public datasets, and traditional approaches struggle with accurately classifying and detecting fish catches. This is primarily due to the issue of varying scales in different images, where severe occlusion between fish catches is a major problem. Additionally, the complexity of the environment poses numerous challenges for recognition, as fish catch images often include water bodies, which confuse differentiation [17]. The interaction of natural light fluctuations further exacerbates these issues, leading to changes in image properties that hinder the extraction and detection of image features. Secondly, as the sorting system serves as an embedded system, it is necessary to decrease the quantity of algorithm parameters to meet the requirements of the embedded system.

Figure 1
Trawl fishing catch



This paper focuses on the problem of fish catch sorting and detection. Based on the YOLOv8 network framework, a new dilated convolution module and attention mechanism are designed and integrated into the backbone network, along with a lightweight detection head. This improved YOLOv8 network architecture is integrated into a custom fish catch sorting dataset to obtain an enhanced YOLOv8 classification and detection model algorithm, which enables the detection of target fish catches, thereby accurately and rapidly detecting different fish catches.

The algorithm proposed in this paper is more effective than traditional algorithms in extracting image features from multiple scales, enhancing feature representation, and is better suited for scenes with dense objects and severe occlusion. In response to the application requirements of embedded systems, the algorithm presented in this paper has fewer parameters and faster detection speed compared to traditional algorithms.

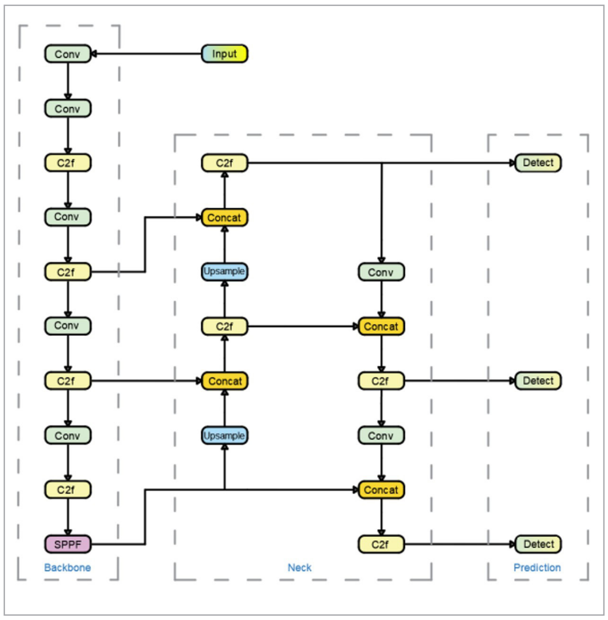
2. Improve the YOLOv8 Model

2.1. Overall Framework Improvement

2.1.1. YOLOv8 Framework

Illustrated in Figure 2, the network architecture of YOLOv8 primarily comprises three components: Backbone, Neck, and Head. The Backbone mainly extracts features, while the Neck fuses these feature maps from various Backbone stages to boost their representational power; in YOLOv8, the Head component primarily handles the assignments of detecting and categorizing objects.

Figure 2
YOLOv8 network framework



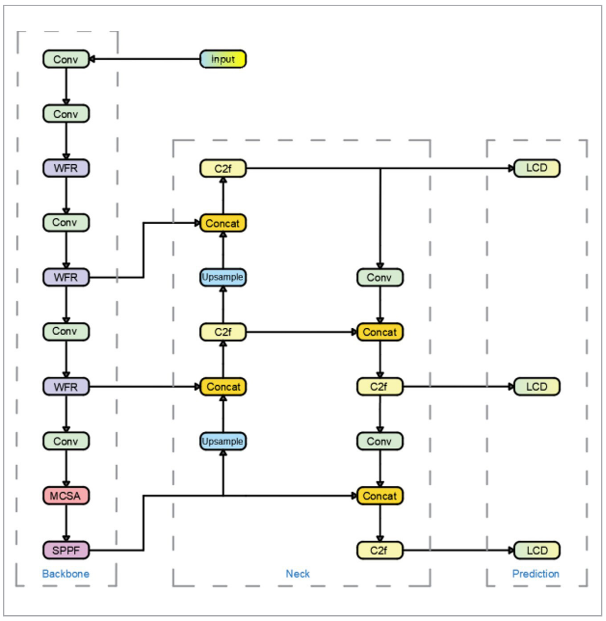
2.1.2. Improved YOLOv8 Framework

In actual fish sorting and detection, the YOLOv8 network framework still has some deficiencies in aspects such as dense fish targets, mutual occlusion between catches, feature extraction of multiple targets, and limitations on parameter quantity in embedded systems. This study presents a range of enhancements to the YOLOv8 network framework, aiming to tackle these issues.

To enhance detection accuracy, this study introduces YOLOv8-WML, an enhanced model for fish sorting and detection that builds upon YOLOv8. This study redesigns the object detection framework by merging the DilatedReparamBlock convolution with the DWR module to construct a novel dilated convolution module. This module replaces all C2f modules in the backbone of the original YOLOv8 network framework. The attention mechanism module devised in this study is incorporated into the backbone, and the detection head receives lightweight enhancements. Figure 3 displays the enhanced network framework, where the purple-shaded box highlights the substituted dilated convolution module (WFR module), the red background box represents the introduced attention mechanism module(MCSA module), and the yellow-green background box denotes the replaced lightweight detection head.

When an image is input into the WFR module, the WFR module first performs initial processing on the image through its multi-scale feature extraction capabilities, generating a feature map rich in contextual information. Subsequently, the MCSA attention mechanism processes the feature map, adjusting the network's focus on different regions, enhancing important features, and strengthening the generation of offset values for each convolutional window and sampling weights at different positions within the window, making the model more accurate in subsequent

Figure 3
Improved YOLOv8 network framework



target recognition and classification tasks. Finally, the lightweight detection head further processes the feature map using shared convolutions and normalization layers to generate category predictions for the targets.

Addressing the issue of mutual occlusion between objects, the framework proposed in this paper replaces the C2f module with the WFR module in the backbone network. The WFR module utilizes dilated re-parameterized convolution to expand the receptive field, enabling the model to extract features over a larger spatial range and thus better handle densely packed and occluded targets. Additionally, the MCSA attention mechanism is introduced into the backbone network. This mechanism helps the model more accurately locate targets in cases of occlusion and overlap by capturing attention information in the spatial dimension. For the problem of multi-target feature extraction, the WFR module enhances the model's feature extraction capabilities at different scales through a multi-branch structure design and semantic residualization methods, enabling the model to better handle multi-target scenarios. The MCSA attention mechanism improves the model's ability to recognize multiple targets in complex scenes through global and local feature fusion. To address the limitations on the number of parameters in embedded systems, the framework in this paper lightens the detection head. The improved detection head significantly reduces the number of parameters and computational load by using shared convolutions, and increases detection speed by employing normalization layers.

2.2. Replacement of Backbone Network Framework Module

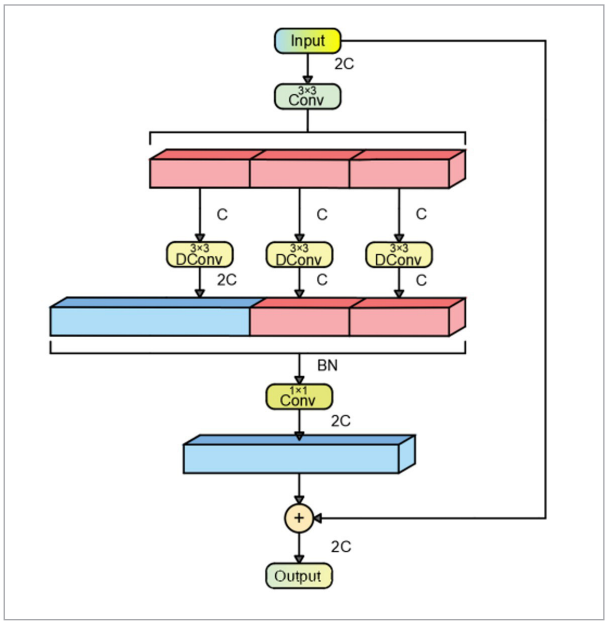
2.2.1. Dilation-wise Residual Module

Figure 4 illustrates the DWR (Dilation-wise Residual) module, a multi-branch design aimed at improving multi-scale feature extraction. This module integrates the principles of dilated convolutions alongside residual connections, targeting the difficulties of extracting multi-scale features in applications like real-time semantic segmentation [25].

2.2.2. Wide-Field Residual Module Proposed in This Study

The DWR module is primarily applicable to the high-level structures of a network and has limited effectiveness in enhancing feature extraction capabilities in the lower and middle levels of the network. Therefore, this paper introduces dilated re-parameterized convolutions [3] in combination with the DWR module to design a new multi-branch dilated convolution module, intended to enhance the model's multi-scale feature extraction capability across various network levels.

Figure 4
DWR module



As shown in Figure 5, the newly designed module is named the "Wide-Field Residual (WFR)" module. Based on the multi-branch architecture of the DWR module [25], in this module, dilated convolutions with dilation rates of 3 and 5 are substituted with dilated re-parameterized convolutions that employ dilation rates of 5 and 7. The latter introduces dilation factors, allowing the convolution kernels to expand the receptive field without elevating computational complexity. This facilitates the extraction of features from a larger range, thereby capturing richer contextual information [3].

The module initially uses three 3×3 convolutions for preliminary feature extraction. Subsequently, each branch generates relevant residual features from the input features through a Batch Normalization (BN) layer and a ReLU layer, which is referred to as regional residualization [6]. Regional residualization produces a series of concise regional feature maps in simplified form for the subsequent semantic residualization. The feature representation obtained through regional residualization is denoted as f, and its expression is given by:

$$f = \mathrm{ReLU}(\mathrm{BN}(F_3)). \tag{1}$$

The formula includes F_3, which denotes a 3×3 convolution function, BN, representing Batch Normalization, and ReLU, which serves as the activation function.

Thereafter, dilated depthwise convolutions with a dilation rate of 1 and dilated reparameterized convolutions are applied separately to perform morphological filtering on regional features of varying sizes. This process is referred to as semantic residualization, which aids in reducing redundant information in the feature maps and highlights key features. The expressions for obtaining the feature representations f_1, f_2, and f_3 through semantic residualization are:

$$f_1 = \mathrm{d}_1\mathrm{DConv}(f) \tag{2}$$

$$f_2 = \mathrm{d}_5\mathrm{DRBConv}(f) \tag{3}$$

$$f_3 = \mathrm{d}_7\mathrm{DRBConv}(f). \tag{4}$$

In Equation (2), d_1DConv represents a dilated depthwise convolution with a dilation rate of 1; in Equation (3), d_5DRBConv represents a dilated re-parameterized convolution with a dilation rate of 5; in Equation (4), d_7DRBConv represents a dilated re-parameterized convolution with a dilation rate of 7.After acquiring multi-scale context information through morphological filtering, multiple outputs are concatenated and processed with BN. Pointwise convolution is employed to combine features and generate the ultimate residual. The ultimate residual is incorporated into the input feature map to formulate a stronger and more extensive feature representation. The output y is represented as:

$$y_{(x)} = F_1\big(\mathrm{BN}([f_1, f_2, f_3])\big) + x. \tag{5}$$

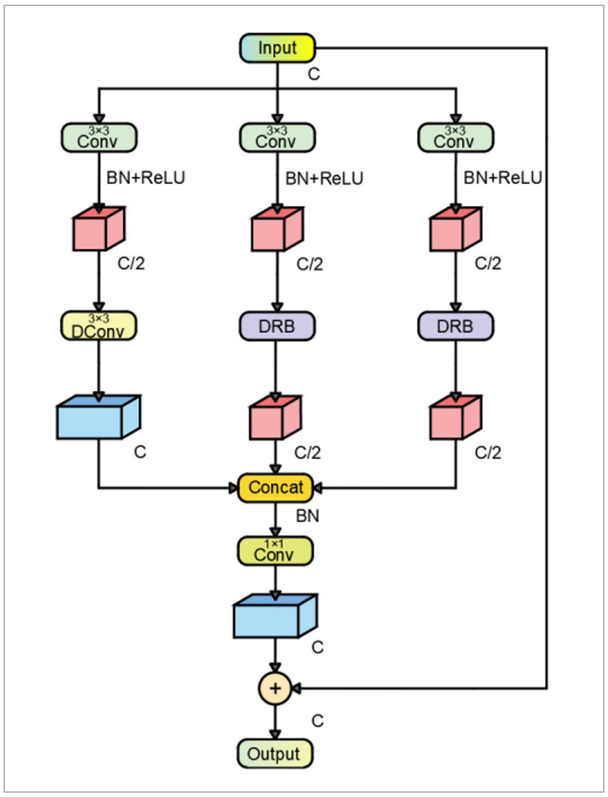
In Equation (5), F_1 represents a 1×1 convolution function, BN stands for Batch Normalization, and $[f_1, f_2, f_3]$ indicates the concatenation of the three feature maps f_1, f_2, and f_3 along the channel dimension, with x being the given input.

In this module, three branches generate feature maps f_1, f_2, and f_3 containing information at different scales through two steps of regional residualization and semantic residualization. By concatenating these feature maps, features from different scales are fused together to form a feature map that includes richer contextual information. This fusion helps the model capture key features at different scales, enhancing the feature representation capability. Subsequently, the feature map undergoes Batch Normalization (BN) processing, which normalizes the channels of the feature map so that the input data distribution of each batch has the same mean and variance, thereby accelerating the training speed. Finally, point-wise convolution is applied to further fuse the feature map, combining information from multiple channels to form a new feature representation. This process can be regarded as cross-channel information exchange, which aids in generating more representative features.

Since establishing connections directly over large spatial spans through convolution is always more

Figure 5

Wide-Field Residual module



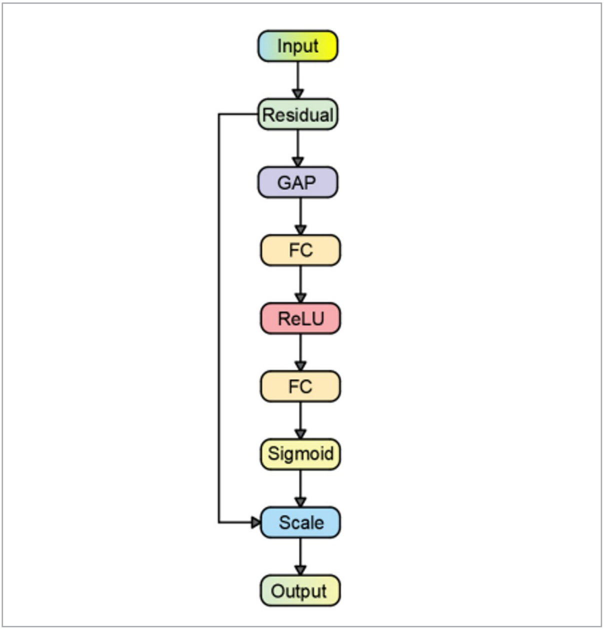
challenging, and long-span connections require the assistance of short-span connections, small receptive fields at each stage are always crucial. Therefore, the first branch's output channel count is increased to double the amount of the other branches.

2.3. Introduction of Attention Mechanism

2.3.1. Squeeze-and-Excitation Attention

Illustrated in Figure 6, the SE attention mechanism, alternatively referred to as the Squeeze-and-Excitation mechanism, revolves around the core idea of reallocating information between channels by learning an attention weight vector that represents the relationships among channels. This enhances the network's focus on important feature channels [10]. Nevertheless, the SE attention mechanism solely focuses on the channel dimension for attention and fails to capture attention in the spatial dimension.

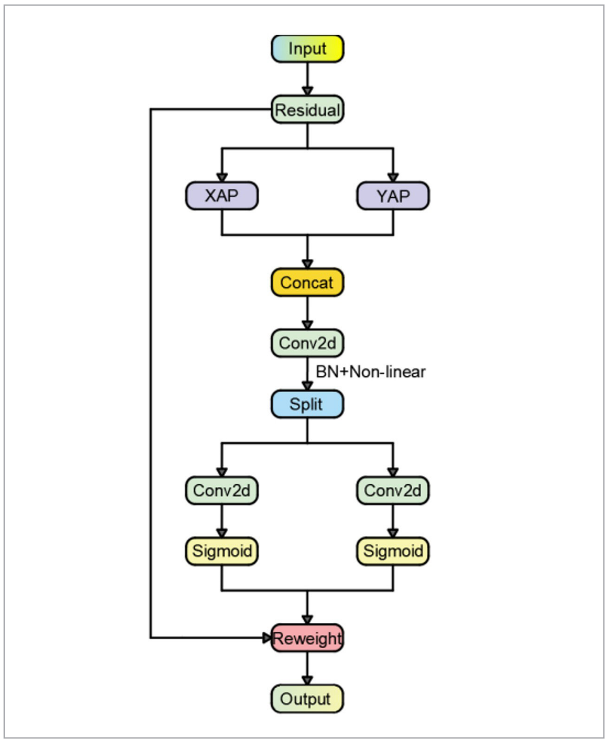
Figure 6
SE attention module



2.3.2. Coordinate Attention

Illustrated in Figure 7, the CA attention mechanism, alternatively termed as the Coordinate Attention mechanism, is a spatial attention mechanism that can effectively enhance the neural network model's focus on important features in the feature map[8].

Figure 7
CA attention module



Despite its potential, the CA attention mechanism is burdened by an extensive parameter count, rendering it impractical for deployment in embedded systems.

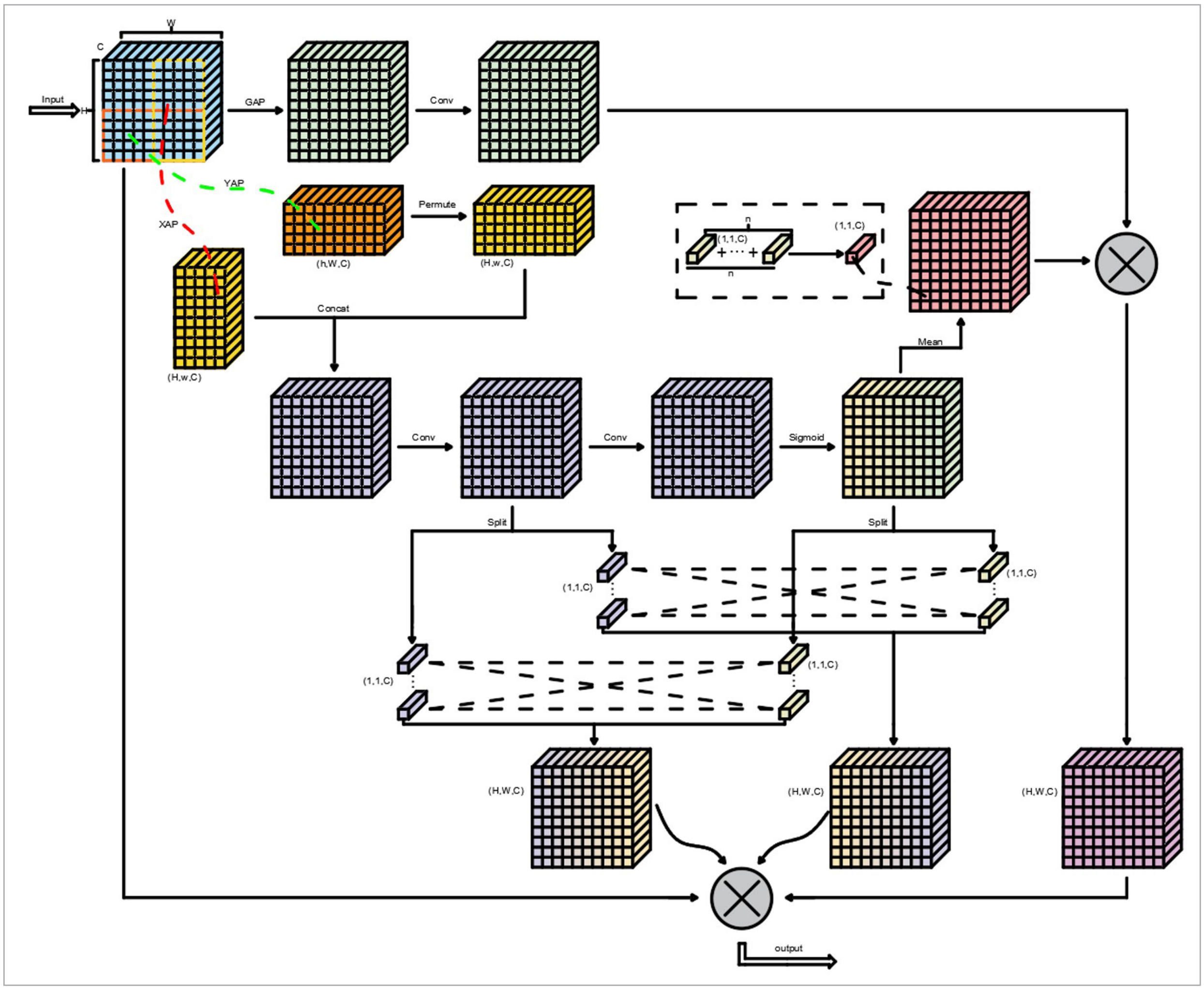
2.3.3. Multi-channel Spatial Attention Proposed in This Study

Attention mechanisms have become a key technology in deep learning models, optimizing model performance by adjusting the network's focus on different parts. However, current attention mechanisms, like CBAM (Convolutional Block Attention Module) and SE attention, often employ global max pooling or average pooling in handling channel attention, potentially resulting in the loss of spatial details [18]. On the other hand, despite the CA attention mechanism taking spatial information into account, its substantial computational complexity restricts its utilization in large-scale networks [5, 22, 27].

In response to the aforementioned deficiencies, as illustrated in Figure 8, this paper introduces an innovative attention mechanism coined as "Multi-channel Spatial Attention" (MCSA). This attention mechanism can effectively extract features from each channel with

Figure 8

Multi-channel Spatial Attention module



a relatively small quantity of parameters, accurately capture the spatial relationships between different channels, and enhance the generation of offsets and sampling weights. Unlike the CA and SE attention mechanisms, the MCSA attention mechanism embeds precise positional information at the initial stage, retains the global features of spatial information through global pooling. Additionally, it diminishes the overall count of model parameters. This allows the subsequent attention calculations to fully utilize these spatial details and global features with a smaller number of parameters, and enhances the generation of offsets for each convolutional window and sampling weights for different positions within the window.

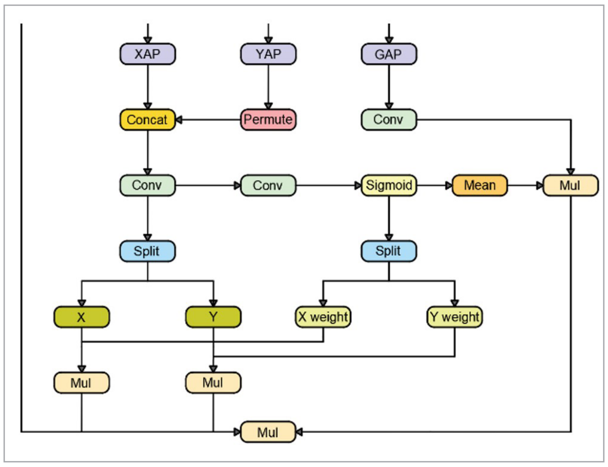
As shown in Figure 9, this attention mechanism retains global average pooling while decomposing it into one-to-one feature encoding operations. Specifically, for a given input *x*, while performing global average pooling, two spatial pooling kernels, specifically (H, 1) and (1, W), are utilized to encode each channel along the horizontal and vertical axes, respectively. For a given height h in channel c, the output is formulated as follows:

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} x_c(h, i). \tag{6}$$

Likewise, the representation of the output for the c channel at a specific width w is given as:

Figure 9

Simplified Diagram of MCSA Module



$$z_c^w(w) = \frac{1}{H}\sum_{0\le j\le H} x_c(j, w). \tag{7}$$

Equations (6)-(7) individually accumulate features across two distinct spatial directions, producing a set of feature maps that are sensitive to direction. By doing so, these transformations enable the attention module to grasp extensive dependencies in one spatial direction while retaining exact positional details in the other. This capability assists the model in precisely pinpointing objects of interest. Equations (6)-(7) attain a comprehensive receptive field while embedding accurate positional details. The attention module sends the aggregated feature map generated by Equation (7) into the Permute function to rearrange the dimensions of the feature vectors, adapting to subsequent convolution operations. This facilitates improved organization and processing of feature data, allowing the model to learn the inherent structure of the data more efficiently [26]. Subsequently, it concatenates this with the aggregated feature map generated by Equation (6) and sends them to a shared 1×1 convolutional transformation function F_1 for further transformation and fusion of information from these two feature maps, obtaining the feature representation f:

$$f = F_1([z^h, \mathrm{Permute}(z^w)]) \tag{8}$$

In Equation (8), F_1 represents a 1×1 convolutional function. The role of the Permute function is to adjust the dimension order of the tensor. The notation $[z^h$, Permute$(z^w)]$ indicates the concatenation of the two feature maps, z^h and z^w, along the channel dimension. f is a feature representation that integrates precise positional information and inter-channel relationships.

The attention module functions in two steps on the feature representation f, which encompasses spatial information extending both horizontally and vertically. Initially, the split function is utilized to divide the feature representation f into two separate tensors according to the spatial dimension. Subsequently, the feature representation f is passed through a common 1×1 convolutional transformation function, F_1, which serves to thoroughly extract feature data both horizontally and vertically by processing along these two directions, we obtain the intermediate feature representation f_1:

$$f_1 = F_1(f). \tag{9}$$

In Equation (9), F_1 represents a 1×1 convolutional function.

The attention module feeds the obtained intermediate feature representation f_1 into a sigmoid activation function and then uses a split function to divide f_1, obtaining the attention weights g^h on the height dimension and g^w on the width dimension of the feature map. The expressions for g^h and g^w are shown in Equations (10)-(11), respectively:

$$g^h = \mathrm{Sigmoid}(f_1^h) \tag{10}$$

$$g^w = \mathrm{Sigmoid}(f_1^w) \tag{11}$$

In Equations (10)-(11), the activation function employed is the Sigmoid function.

The obtained attention weights g^h and g^w are then multiplied with the two separate tensors obtained in the first step to achieve weighted processing, resulting in more refined attention weights G^h and G^w. The expressions for G^h and G^w are shown in Equations (12)-(13), respectively:

$$G^h = f^h \odot g^h \tag{12}$$

$$G^w = f^w \odot g^w \tag{13}$$

In the third branch, the attention module utilizes global average pooling on the feature map, aiming to capture the essential spatial information while effectively decreasing the number of model parameters [20, 21]. For a given input x, the squeeze step for the c channel can be represented as:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j). \quad (14)$$

In Equation (14), H represents the height and W indicates the width of the feature map. The element x_c(i,j) is positioned on the input feature map, where i and j correspond to the horizontal and vertical coordinates of this element, individually.

The obtained feature representation is then fed into a shared 1×1 convolutional transformation function F_1 to obtain the feature representation f_2:

$$f_2 = F_1(z_c). \quad (15)$$

In Equation (15), F_1 represents a 1×1 convolutional function.

The intermediate feature representation f_1 undergoes processing through the Sigmoid activation function, followed by a mean operation. This is done to normalize the intermediate feature representation f_1, resulting in a tensor with the same quantity of channels as the input x. Subsequently, this normalized tensor is multiplied with the feature representation f_2 to obtain the global attention weight g_c, which is expressed in Equation (16) as follows:

$$g_c = f_2\left(\text{Mean}\left(\text{Sigmoid}(f_1)\right)\right). \quad (16)$$

In Equation (16), the Mean function represents the operation of calculating the mean, and Sigmoid is the activation function.

This attention module multiplies the attention weights G^h, G^w, and g_c obtained from the three branches to produce the output y_c of the attention module, which is represented as:

$$y_c(i, j) = x_c(i, j) \times G_c^h(i) \times G_c^w(j) \times g_c(i, j). \quad (17)$$

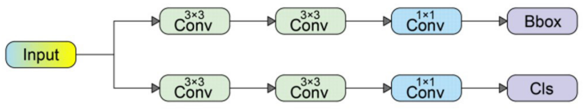
2.4. Lightweight Detection Head

2.4.1. YOLOv8 Detection Head

The detection head in YOLOv8 is vital for object detection and accounts for 1/5 of the computational load within the model framework. The “Detect” component corresponds to the detection head part of YOLOv8, comprising two distinct branches, as depicted in Figure 10.

Figure 10

YOLOv8 Detection Head branch

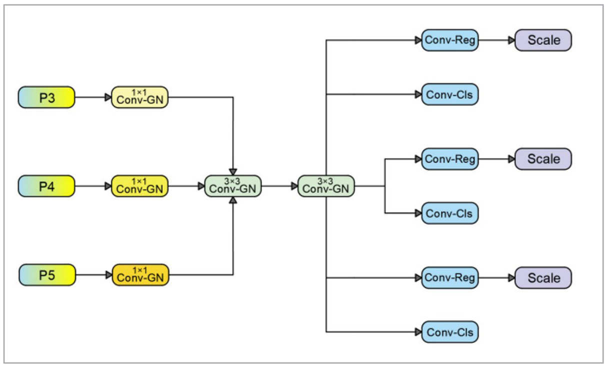


The two branches of the YOLOv8 detection head extract information through two 3×3 convolutions and a 1×1 convolution, respectively. Ultimately, they determine the bounding box regression loss (denoted as Bounding Box.loss) and the classification loss (denoted as Cls.loss or Classification Loss), each being calculated separately. After the three layers of convolution, a for loop is employed to traverse each of the three channels, which markedly raises the computational load and parameter count of the detection head.

2.4.2. Light Conv Detect Proposed in This Study

As depicted in Figure 11, the aim is to decrease both the number of parameters and the computational burden associated with the detection head, this paper designs a new detection head named Light Conv Detect (LCD). In each branch, the detection head feeds the feature map into a 1×1 convolution (Conv-GN), which consists of a convolutional layer and a normalization layer. The convolutional layer is tasked with learning the spatial relationships among the input features. Meanwhile, the normalization layer standardizes the output of the convolutional layer, aiming to expedite the training process and enhance the model’s generalization capabilities. The Conv-GN convolution can enhance the detection head’s ability in localization and classification. After being transformed by the 1×1 Conv-GN convolution, the feature maps on each branch are input into a 3×3 shared convolution (Conv-GN). After being transformed by this convolution, the feature maps are then fed into another 3×3 shared convolution (Conv-GN). These two 3×3 convolutions share the same convolutional parameters, thereby achieving efficient feature extraction and object detection. This procedure has the ability to significantly reduce the number of parameters, all while preserving a high degree of precision[7]. Upon extracting the feature information from the feature maps, three shared classification convolutions (Conv-Cls) are utilized to produce the category predictions for the targets, and the feature maps are convert-

Figure 11
Light Conv Detect



ed into category probability distributions through this convolutional transformation function. By using classification convolutions (Conv-Cls) for convolutional transformation at different scales, the detection head achieves accurate classification of targets of different sizes. The detection head uses three shared regression convolutions (Conv-Reg) to predict the position information of the targets (such as bounding box coordinates), and the classification convolutions (Conv-Cls) and regression convolutions (Conv-Reg) used here share parameters. While using shared convolutions, to tackle the problem of varying target scales identified by individual detection heads, a scaling layer (Scale) is employed to adjust the feature maps, ensuring uniformity in the target scales detected across all heads. By using shared convolutions and convolutional parameter sharing, the detection head significantly decreases the parameter count and computational load with minimal loss of accuracy, making the model lighter and improving the speed of object detection.

3. Self-Defined Dataset

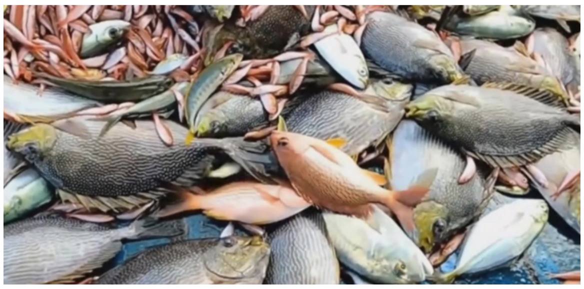
The effectiveness of the fish catch classification and detection algorithm introduced in this study is assessed using a customized dataset (https://github.com/tiange120/self-dataset). This dataset initially comprises 9,000 images and is augmented to 12,000 images through methods such as flipping and cropping. It covers ten different major economic fish catches: golden pomfret, silver pomfret, Pacific saury, swimming crab, whelk, red snapper, codfish, sardine, Pacific bluefin tuna, and skipjack tuna. Using Make Sense software, manually classify and annotate the given objects, and mark their bounding boxes. As shown in Figure 12, it presents an example from this dataset. The dataset is partitioned into training, validation, and testing subsets with a ratio of 7:2:1, aimed at evaluating the performance of the YOLOv8-WML network framework.

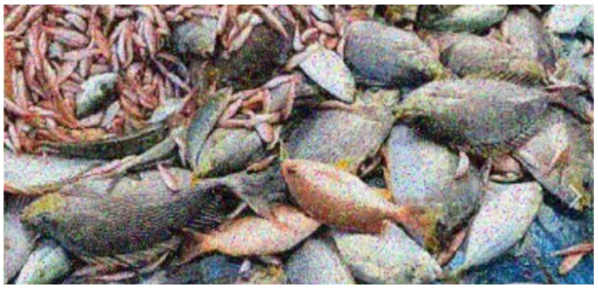
3.1. Add Noise Processing

To prevent overfitting, noise is added to the images transformed through methods such as flipping and cropping. As shown in Figure 13, salt and pepper noise, Gaussian noise, and Poisson noise are added to some images to simulate the potential interferences that images may encounter in real-time detection. By adding different types of noise, the complexity and diversity of the dataset can be increased. This aids the model in moving beyond mere dependence on specific features of the training data, enabling it to learn how to extract useful feature information from a broader range of data, thereby enhancing the model's generalization ability and robustness.

Figure 12
Some fish catch images from the Self-Defined dataset

Figure 13
Comparison of Image with Noise Addition Processing

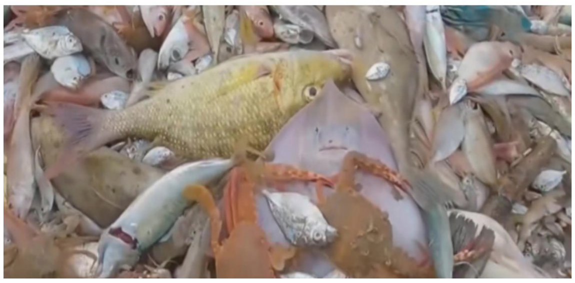

(a) Original fish catch image

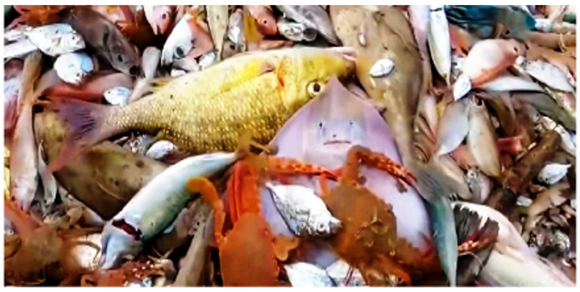

(b) Image after noise addition processing

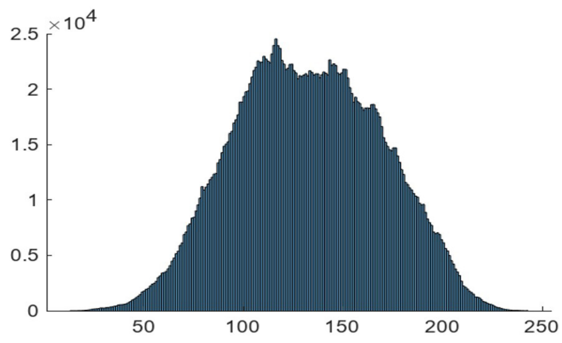
3.2. Histogram Equalization Processing

Histogram equalization processing is applied to images with uneven grayscale distribution due to lighting issues, which can cause blurred object textures, color distortion, and weakened features. As shown in Figure 14, histogram equalization processing can enhance the image's contrast by adjusting its grayscale distribution, thereby expanding its dynamic range [11]. This enhancement of contrast reduces color distortion and shadowing in the image, making it clearer and helping the model capture more useful feature information [17].

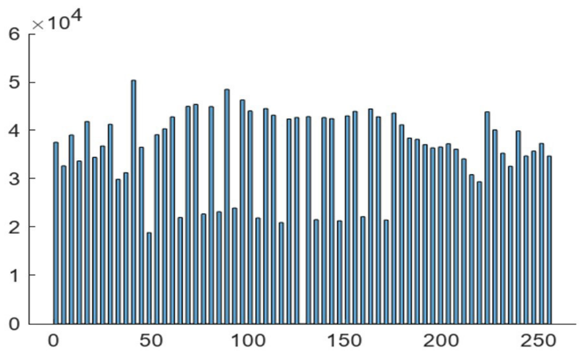
Figure 14
Image contrast after histogram equalization processing and its histogram comparison


(a) Original fish catch image


(b) Image after histogram equalization processing



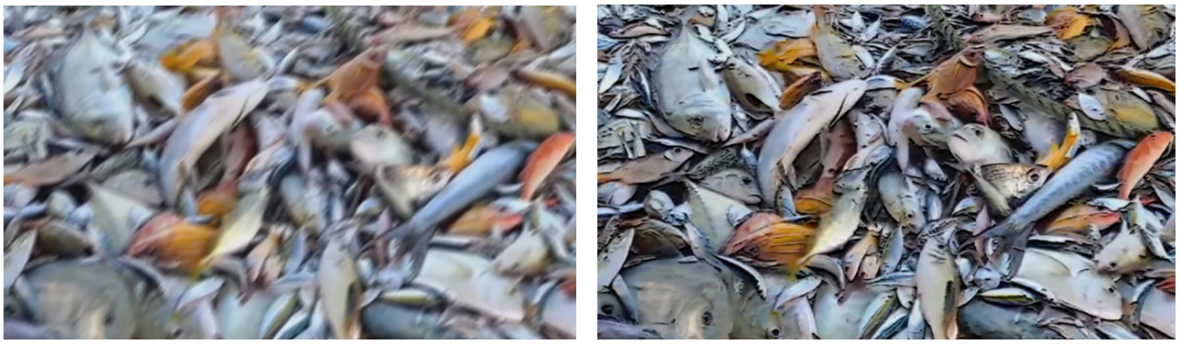
(c) Original image histogram



(d) Equalized Histogram

Figure 15

Enhance image contrast through sharpening processing



(a) Original fish catch image (b) Image after sharpening process

3.3. Sharpening Process

Enhance the sharpness of the image, particularly in areas where object contours are indistinct and features appear blurred due to shaking. Figure 15 illustrates that sharpening can accentuate the edges and contours within the image, thereby enhancing the high-frequency information [12]. This augmentation facilitates the model's ability to capture and detect pivotal features in the image, as well as to distinguish between various objects and regions within intricate backgrounds.

4. Model Training

4.1. Environmental Setup

The hardware environment used in this study is an NVIDIA GeForce RTX 3070Ti graphics card with Ubuntu 20.04 GPU drivers; the software environment used is Python 3.8.16 and torch 1.13.0 + cu117. All experiments are set to train for 500 epochs, and training will be stopped early if there is no significant improvement in average precision after 50 epochs. In the process of model training, specific parameters are established: the batch size is designated as 24, the learning rate is fixed at 0.01, the SGD momentum is assigned a value of 0.937, and the optimizer weight decay is set to 0.0005. All remaining training parameters adhere to the default settings specified in the YOLOv8s model.

4.2. Training Results of the Model Validation Set

The model is configured for 500 training rounds, and the training process will cease automatically if no notable enhancement in average accuracy is observed. After the 152nd training round, the YOLOv8-WML model achieved training results on the custom dataset. Figure 16 displays various performance metrics for both the training set and the validation set.

Figure 16 illustrates the bounding box loss curve, the object loss curve, and the classification loss curve for the enhanced YOLOv8 model, depicted in the first three columns. The three curves depicted in the initial three columns exhibit the loss trend, with the horizontal axis denoting the training rounds and the vertical axis indicating the aggregate loss value. As can be seen from the curves, with the progression of training rounds, the overall loss value continues to decrease and eventually stabilizes. The findings of the experiments indicate that the YOLOv8-WML model presented in this paper displays satisfactory levels of fitting capability, stability, and precision. The last two columns display the accuracy curve and the mean accuracy curve. The horizontal axis signifies the number of training rounds, while the vertical axis denotes both the accuracy rate and the average accuracy rate. The last two columns feature curves that assess object detection performance based on varying confidence thresholds; a value closer to 1 on these curves signifies a higher level of confidence demonstrated by the model.

Figure 17 displays a confusion matrix that depicts the prediction accuracy of the YOLOv8-WML model for the 10 categories of fish catches in the custom dataset. Furthermore, it demonstrates the connection between the predicted categories and their corresponding accuracy rates. Figure 16 displays a confusion matrix where the columns denote the predicted

Figure 16

Performance metrics of the YOLOv8-WML model

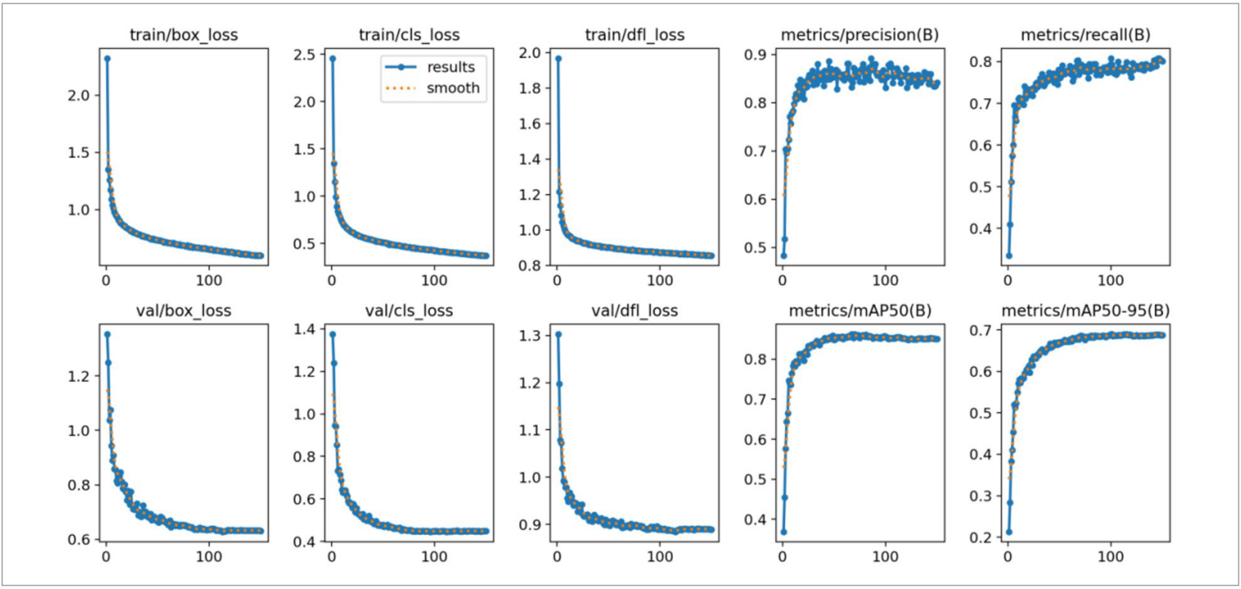
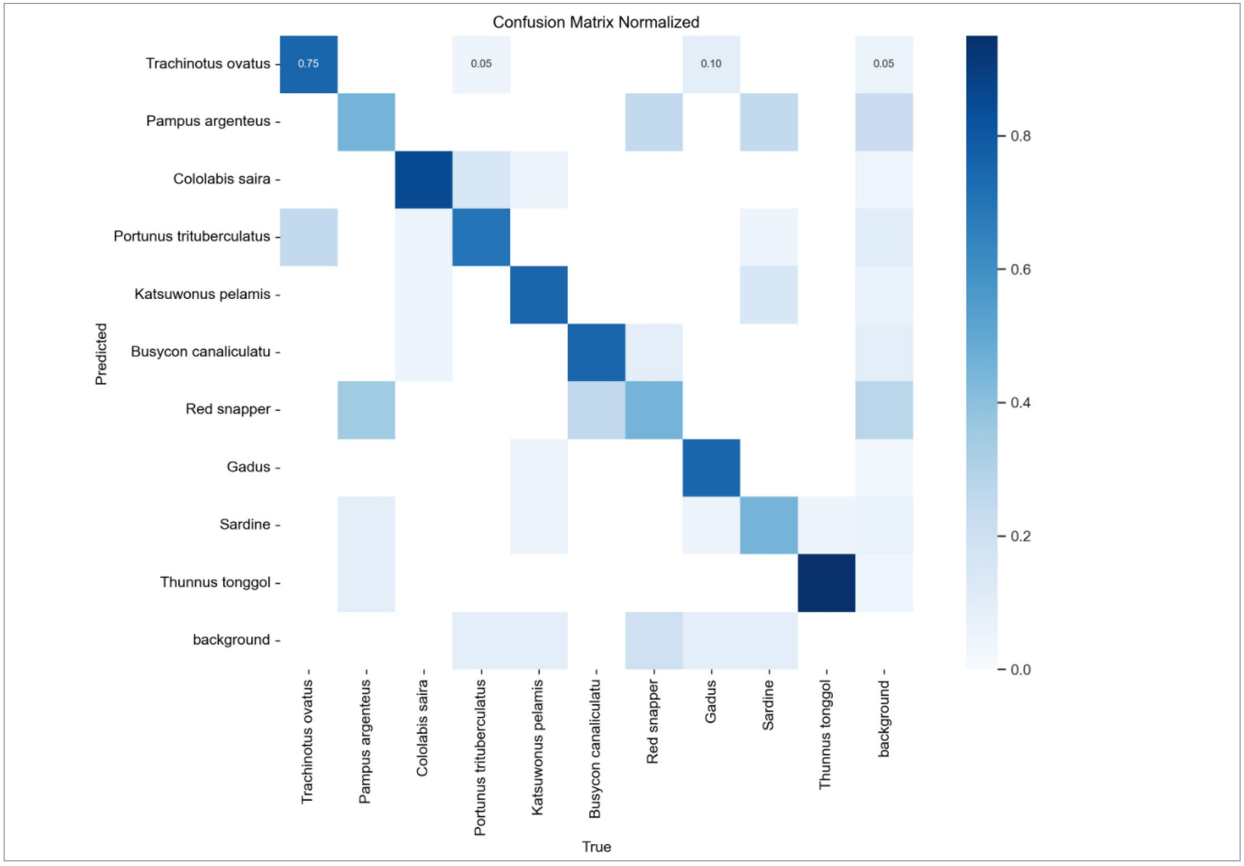


Figure 17

The confusion matrix of the YOLOv8-WML model

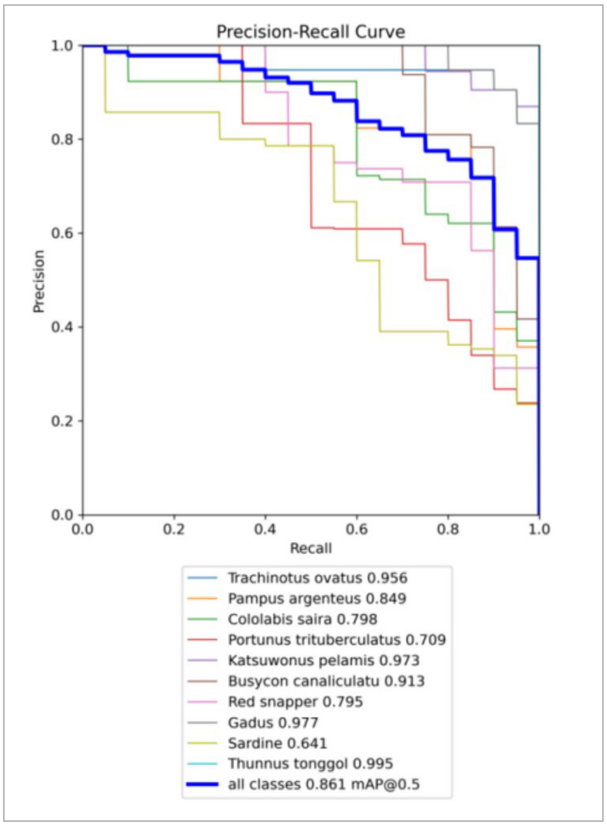


categories, the rows denote the actual labels, and the diagonal elements indicate the prediction accuracy. Figure 16 reveals that the YOLOv8-WML model attains elevated accuracy levels across all categories.

Figure 18 illustrates that as the recall rate rises, the precision also experiences an accelerated rate of change. The PR curve of the YOLOv8-WML model closely hugs the upper right corner, signifying its high recall and precision rates. The considerable area beneath the PR curve indicates that the YOLOv8-WML model exhibits good performance. The PR curve's smoothness suggests a consistent balance between recall and precision within the YOLOv8-WML model.

Figure 18

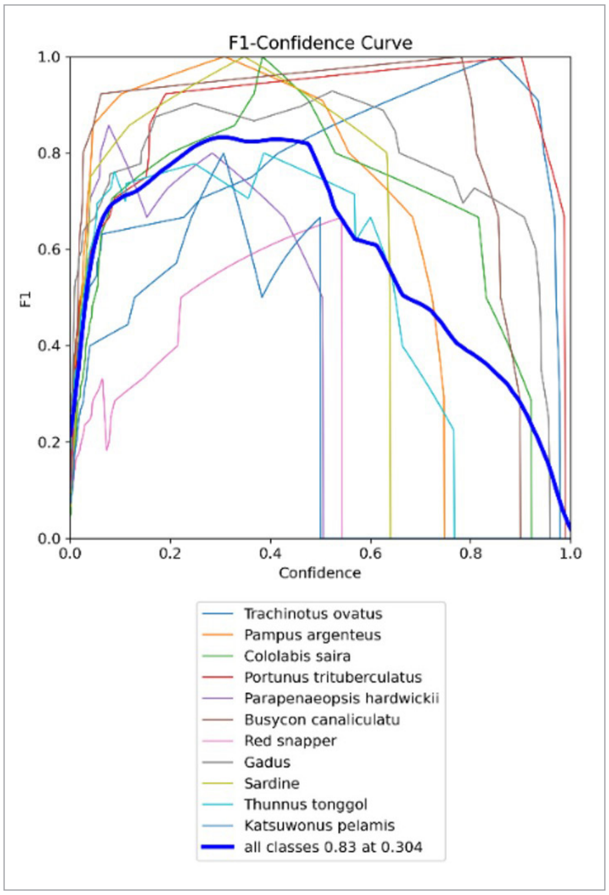
The PR curve of the YOLOv8-WML model



As illustrated in Figure 19, the F1 score typically rises and then falls with an increase in confidence, reflecting the model's transition from being overly cautious to overly confident. The F1-confidence curve in Figure 19 comprehensively evaluates the model's precision and recall in classification tasks by calculating the F1 score at different confidence thresholds. This aids in assessing the model's performance at various confidence levels and the differences in classification difficulty among different categories. The horizontal axis represents confidence, ranging from 0 to 1, indicating the level of certainty in the model's prediction that a sample belongs to a certain category. The vertical axis denotes the F1 score, which is the harmonic mean of precision and recall, used to gauge the accuracy of the model's classification. The multiple colored lines in Figure 19 depict the trends of F1 scores for different categories as confidence varies. For all classes, when the confidence level is 0.304, the F1 score reaches 0.83, indicating that the model performs exceptionally well at this confidence level.

Figure 19

The F1-Confidence Curve of the YOLOv8-WML Model



4.3. Ablation Experiment

To evaluate the effectiveness and reliability of the YOLOv8 enhancement scheme across various aspects, and to gauge how they contribute to performance improvements by systematically withdrawing these enhancements, we carried out an ablation study. Table 1 displays the experimental outcomes obtained from the validation set, utilizing the Self-Defined dataset.

From Table 1, although the performance of the YOLOv8-WML model decreased due to the replacement of the LCD detection head, the combination of the WFR module and the MCSA module brought performance improvements compared to the baseline. This is attributed to the complementarity of these three technologies. The WFR module can extract more detailed features at multiple scales, while the MCSA module can accurately capture spatial relationships between different channels and enhance feature representations, thereby compensating for the performance degradation caused by the LCD detection head. The lightweight convolutional detection head is designed to meet the parameter constraints of embedded systems. By minimizing the use of shared convolutions, it effectively reduces the number of model parameters. Throughout this process, efforts are made to minimize the impact on model performance, thereby achieving an ideal balance between model complexity and accuracy. By leveraging the complementarity of these three technologies, the enhanced model boosts overall performance and accelerates computational speed, proving advantageous for real-time object detection.

Table 1's first row showcases the detection results of the original YOLOv8 model, serving as the experimental baseline. The second row displays the outcomes following the integration of the WFR module, capable of capturing intricate features across various scales and enhancing the receptive field for each network. When the WFR module is integrated, it leads to an enhancement of 4.7% in mAP@0.5 and 5.6% in mAP@0.5:0.95 compared to the baseline YOLOv8 model. The third row presents the findings after introducing the MCSA module, an attention module that divides channels into three, decomposes global pooling into one-to-one feature encoding operations while retaining global pooling, aiming to fully extract features from each channel with a smaller number of parameters and accurately capture spatial relationships between different channels. In contrast to the initial YOLOv8 model, there is a 1.8% increase in mAP@0.5 and a 2.3% boost in mAP@0.5:0.95. The fourth row shows the findings of replacing the detection head with the LCD detection head. This findings in a notable reduce in the quantity of parameters and computational load, achieved through the utilization of shared convolutions. When comparing the modified YOLOv8 to its initial counterpart, it's noted that while mAP@0.5 decreases by 2.9% and mAP@0.5:0.95 by 2.6%, there is a significant 18% reduction in parameter count and a 16.9% decrease in computational load. The fifth row presents the experimental findings when using both the WMR module and the MCSA module. When compared to the baseline, there is a 6.2% enhancement in mAP@0.5 and a 6.3% improvement in mAP@0.5:0.95. The last row displays the ultimate outcomes of the fish sorting detection model introduced in this paper, achieving improvements of 2.2% and 3.7% in mAP@0.5 and mAP@0.5:0.95 individually, and the quantity of parameters and computational load reduced by 16.63% and 13.03% respectively. The experimental results demonstrate herein validate the assertion of this paper, namely that the

Table 1
Ablation experiment table for the YOLOv8-WML model

WFR	MCS	LCD	mAP@0.5	mAP@0.5:0.95	parameters	GFLOPs
			0.839	0.649	11127906	28.4
√			0.886	0.705	11844252	29.5
	√		0.857	0.672	11293605	29.0
		√	0.810	0.623	9124781	23.6
√	√		0.901	0.712	12010151	30.2
√	√	√	0.861	0.686	9277570	24.7

YOLOv8-WML model enhances detection accuracy and expedites the detection process, while simultaneously diminishing the parameter count and alleviating the computational burden.

5. Comparative Experiment

5.1. Analysis of Experimental Outcomes Across Various Datasets

In order to assess the model's generalization capability, comparative experiments were performed utilizing the publicly available dataset FishKnowledge. This dataset bears resemblance to the custom dataset, as they both depict scenarios characterized by densely packed biological entities, targets at multiple scales, and instances that may partially obscure one another[14]. The FishKnowledge dataset consists of 8723 training images, 2111 validation images, and 1123 testing images. The YOLOv8-WML model, which we propose, underwent training and testing in conjunction with the YOLOv8 model. The findings are illustrated in Figure 20.

The results are presented in Figure 20, the YOLOv8-WML model achieved favorable findings on the pub-

Figure 20

Performance comparison between the YOLOv8-WML model and the YOLOv8 model

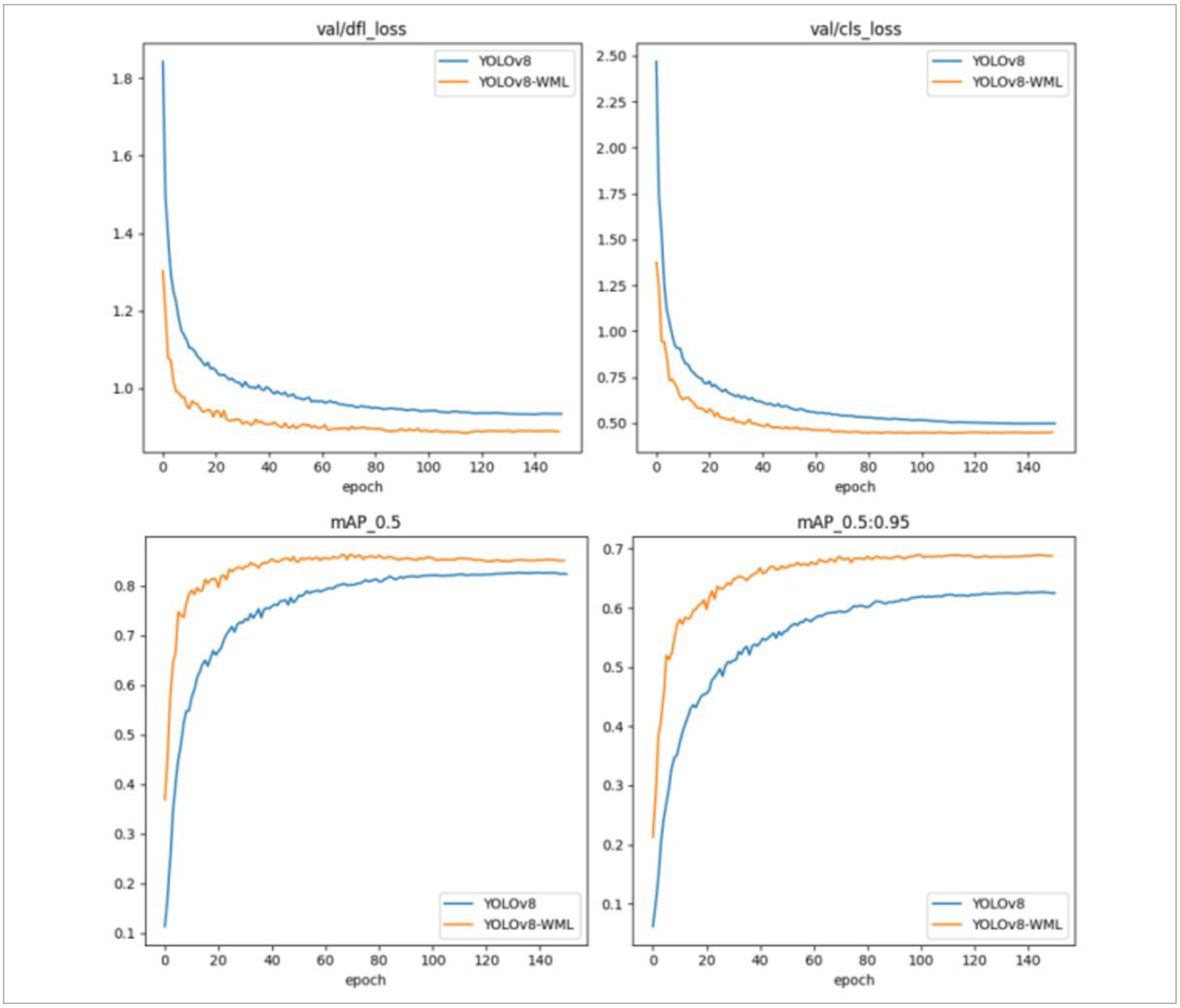
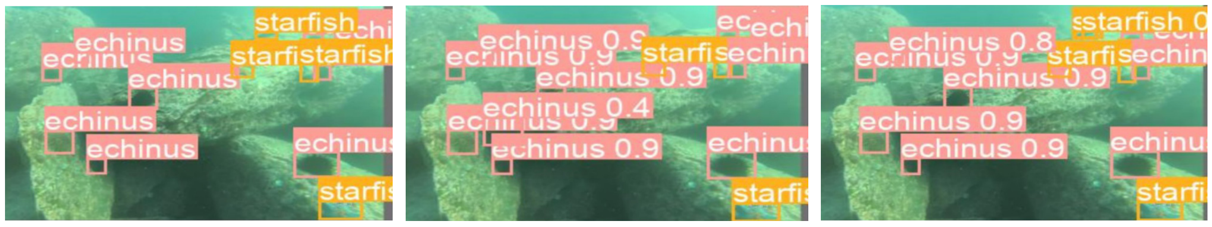


Figure 21

Comparison of Image Detection Results on Public Datasets



(a) Real Image Labels (b) Image Detected by YOLOv8s Model (c) Image Detected by YOLOv8s-WML Model

lic dataset. Compared with the YOLOv8 model, it not only exhibited higher mAP@0.5 and mAP@0.95 function curves but also demonstrated significantly lower distribution focal loss (dfl_loss) and classification loss (cls_loss) function curves. This suggests that the YOLOv8-WML model exhibits superior detection accuracy, characterized by diminished positional disparities between the anticipated bounding boxes and the actual ones, as well as reduced inconsistencies between the predicted categories and the genuine categories. The findings of the experiment indicate that the YOLOv8-WML model shows outstanding performance and can be efficiently utilized in scenarios characterized by dense biological entities, multiple scales, and potential occlusion.

As shown in Figure 21, a comparison is made between the recognition results of real-world scenarios, the YOLOv8s model, and the YOLOv8s-WML model on a public dataset. The results clearly demonstrate that the YOLOv8s-WML model outperforms the YOLOv8s model in terms of performance. The YOLOv8s-WML model effectively reduces the miss rate and false alarm rate, enhances accuracy and computational speed, and decreases the number of parameters. The experiments indicate that the proposed YOLOv8s-WML model possesses good generalization capabilities [23] . Through the synergistic effects of the WFR module and the MCSA attention mechanism, it can effectively identify other marine species in different environments.

5.2. Contrasting Experiments and Outcomes Across Various Models

This paper will compare the findings of different models in the experiment from aspects such as precision, recall values, mAP values, F1 values and fps values. All experiments were conducted on a Self-Defined dataset, with the YOLOv8s model as the baseline. Table 2 displays the experimental results, indicating that the YOLOv8s-WML model, designed for fish sorting detection, reached an mAP@0.5 score of 86.1%. This score notably surpasses that of the original YOLOv8s model, demonstrating a significant advantage. It is 1.1% higher than the Faster-RCNN model [19], 0.4% higher than the SSD model [15], and 3.7% higher than the YOLOv5s model. Furthermore, the YOLOv8s-WML model

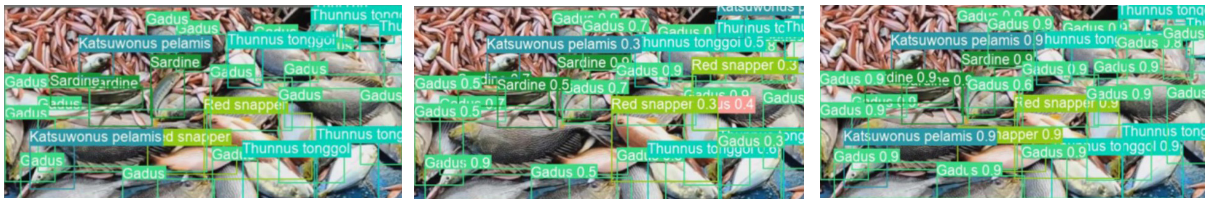
Table 2

Performance comparison table of different models

Model	Precision	Recall	mAP@0.5	mAP@0.95	F1	FPS(f/s)
Faster-RCNN	0.728	0.885	0.850	0.661	0.799	37.2
SSD	0.750	0.871	0.857	0.668	0.806	88.5
YOLOv5s	0.779	0.802	0.824	0.636	0.800	162.7
YOLOv8s	0.803	0.799	0.839	0.649	0.801	140.8
YOLOv8s-WML	0.825	0.820	0.861	0.686	0.822	223.1

Figure 22

Comparison of Image Detection Results



(a) Real Image Labels (b) Image Detected by YOLOv8s Model (c) Image Detected by YOLOv8s-WML Model

achieved an F1 score of 0.822, which is 2.6% higher than that of the YOLOv8s model, 2.5% higher than the YOLOv5s model, 1.9% higher than the SSD model, and 2.8% higher than the Faster-RCNN model. The experimental results indicate that the model proposed in this paper exhibits better performance.

In order to showcase the superior accuracy of the YOLOv8s-WML model presented in this study, a selection of images was made from the Self-Defined dataset. As shown in Figure 22, the fish sorting detection results for real scenes, the YOLOv8s model, and the YOLOv8s-WML model are presented. When dealing with dense, overlapping, and varied-scale targets, the YOLOv8s-WML model notably surpasses the YOLOv8s model. It excels at minimizing missed detections and false positives, enhancing precision and processing speed, while also reducing parameter count and computational burden. This essentially meets the requirements of fish sorting detection tasks and has more practical application value.

6. Conclusions and Future Work

The fish catch sorting and detection model proposed in this paper addresses common challenges in trawl fishing catch sorting video images, such as dense scenes, mutual occlusion, and multi-scale scenarios. Addressing these Difficulties, the present paper proposes a range of novel methodologies. Specifically, In the YOLOv8 network architecture, the C2f module located in its backbone is substituted by the WFR module. The WFR module enhances the model's ability to extract features in multi-scale scenarios, thereby improving the detection precision of the model. An innovative attention mechanism, named MCSA, is introduced. The MCSA attention mechanism can effectively derive features from various channels with a smaller quantity of parameters, accurately capture the spatial relationships between different channels, and simultaneously enhance the generation of offsets and sampling weights. The detection head has undergone lightweight optimizations, resulting in enhanced model performance, a decreased number of parameters, reduced computational load, and faster detection speed. Techniques for enhancing images are utilized to tackle the problem of low-quality original video images, ultimately leading to improved input image quality and subsequent enhancement of the model's performance.

However, despite the numerous advantages demonstrated by the YOLOv8-WML model in experiments, there are still some limitations that need further improvement and refinement in future work. In the experiments, while the Lightweight Convolutional Detection Head (LCD) significantly reduced the model parameters, it also led to a decrease in classification accuracy. Future work can explore more efficient model lightweighting methods, striving to maintain the model's classification accuracy while reducing the number of parameters. For instance, more sophisticated pruning strategies, knowledge distillation techniques, or low-rank factorization methods can be considered to find a better balance between model lightweighting and performance preservation. Additionally, while the current custom dataset covers various types of economically important fish catches, the diversity and scale of the dataset are still limited, potentially failing to comprehensively reflect the complex situations encountered in actual marine fishing, thereby affecting the model's generalization ability. In the future, the dataset can be further expanded to increase its complexity and diversity, enhancing the model's generalization capability.

References

1. Cai, Y., Xu, S., Chen, Z. Community Structure and Diversity Status of Fisheries Resources in the Offshore Waters of the Northern South China Sea. South China Fisheries Science, 2018, 14(2), 10-18.
2. Cui, S., Zhou, Y., Wang, Y., Zhai, L. Fish Detection Using Deep Learning. Applied Computational Intelligence and Soft Computing, 2020, 3738108. https://doi.org/10.1155/2020/3738108
3. Ding, X., Zhang, Y., Ge, Y., Zhao, S., Song, L., Yue, X., Shan, Y. UniRepLKNet: A Universal Perception Large-Kernel ConvNet for Audio, Video, Point Cloud, Time-Series and Image Recognition. Computer Vision and Pattern Recognition, 2024. https://doi.org/10.1109/CVPR52733.2024.00527
4. Feng, Y., Tao, X., Lee, E.-J. Classification of Shellfish Recognition Based on Improved Faster R-CNN Framework of Deep Learning. Mathematical Problems in Engineering, 2021, 1966848. https://doi.org/10.1155/2021/1966848
5. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90
6. He, K., Zhang, X., Ren, S., Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Computer Vision and Pattern Recognition, 2015. https://doi.org/10.1109/ICCV.2015.123
7. He, K., Zhang, X., Ren, S., Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824
8. Hou, Q., Zhou, D., Feng, J. Coordinate Attention for Efficient Mobile Network Design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 13708-13717. https://doi.org/10.1109/CVPR46437.2021.01350
9. Hu, G., He, W., Sun, C., Zhu, H., Li, K., Jiang, L. Hierarchical Belief Rule-Based Model for Imbalanced Multi-Classification. Expert Systems with Applications, 2023, 216. https://doi.org/10.1016/j.eswa.2022.119451
10. Hu, J., Shen, L., Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, 7132-7141. https://doi.org/10.1109/CVPR.2018.00745
11. Huo, J., Chang, Y., Wang, J., Wei, X. Robust Automatic White Balance Algorithm Using Gray Color Points in Images. IEEE Transactions on Consumer Electronics, 2006, 52, 541-546. https://doi.org/10.1109/TCE.2006.1649677
12. Jingchun, Z., Xiaojing, W. E. I., Jinyu, S. H. I. Underwater Image Enhancement Algorithm Based on Blue-green Channel Color Compensation. JEIT 2022, 44, 2932-2939. doi:10.11999/JEIT211444.
13. Jolicoeur-Martineau, A., Mitliagkas, I. Gradient Penalty from a Maximum Margin Perspective. Machine Learning, 2020. arXiv preprint arXiv:1910.06922.
14. Liang, H., Song, T. Lightweight Marine Biological Target Detection Algorithm Based on YOLOv5. Frontiers in Marine Science, 2023, 10. https://doi.org/10.3389/fmars.2023.1219155
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C. SSD: Single Shot MultiBox Detector. Computer Vision and Pattern Recognition, 2016, 9905, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
16. Li-Zhe, C.A.I., De-Yuan, Y., Xiao-Yu, Z., Jing-Xiang, L.I.N., Xin-Wei, C., Xi-Ping, Z., Yi-Yong, R.A.O., Li, M.A., He-Shan, L.I.N., Su-Jing, F. U. Species Composition and Diversity of Marine Organisms from Benthic Trawling in Daya Bay of the Northern South China Sea. Biodiversity, 2017, 25(9), 1019-1030. https://doi.org/10.17520/biods.2017103
17. Mohan, S., Simon, P. Underwater Image Enhancement Based on Histogram Manipulation and Multiscale Fusion. Procedia Computer Science, 2020, 171, 941-950. https://doi.org/10.1016/j.procs.2020.04.102
18. Pang, B. Classification of Images Using EfficientNet CNN Model with Convolutional Block Attention Module (CBAM) and Spatial Group-Wise Enhance Module (SGE). In Proceedings of the International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2022), Sirkemaa, S., Agyeman, M.O., Eds., SPIE: Guilin, China, April 29 2022, 26. https://doi.org/10.1117/12.2636811
19. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39, 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031
20. Sánchez, J., Perronnin, F., Mensink, T., Verbeek, J. Image Classification with the Fisher Vector: Theory and

Practice. International Journal of Computer Vision, 2013, 105, 222-245. https://doi.org/10.1007/s11263-013-0636-x

21. Shen, L., Sun, G., Huang, Q., Wang, S., Lin, Z., Wu, E. Multi-Level Discriminative Dictionary Learning with Application to Large Scale Image Classification. IEEE Transactions on Image Processing 2015, 24, 3109-3123. https://doi.org/10.1109/TIP.2015.2438548
22. Simonyan, K., Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition 2015.
23. Sung, M., Yu, S.-C., Girdhar, Y. Vision Based Real-Time Fish Detection Using Convolutional Neural Network. In Proceedings of the OCEANS 2017 - Aberdeen, IEEE: Aberdeen, United Kingdom, June 2017, 1-6. https://doi.org/10.1109/OCEANSE.2017.8084889
24. Villon, S., Mouillot, D., Chaumont, M. A Deep Learning Method for Accurate and Fast Identification of Coral Reef Fishes in Underwater Images. Ecological Informatics, 2018, 48, 238-244. https://doi.org/10.1016/j.ecoinf.2018.09.007
25. Wei, H., Liu, X., Xu, S., Dai, Z., Dai, Y., Xu, X. DWRSeg: Rethinking Efficient Acquisition of Multi-Scale Contextual Information for Real-Time Semantic Segmentation. Computer Vision and Pattern Recognition, 2023.
26. Woo, S., Park, J., Lee, J.-Y., Kweon, I. S. CBAM: Convolutional Block Attention Module. In Proceedings of the Computer Vision - ECCV 2018, Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds., Springer International Publishing: Cham, 2018, 3-19. https://doi.org/10.1007/978-3-030-01234-2_1
27. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K. Aggregated Residual Transformations for Deep Neural Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 5987-5995. https://doi.org/10.1109/CVPR.2017.634
28. Zhang, X., Zhang, X., Gao, T. Comparative Analysis on Catch Composition with Two Fishing Gears at Yellow River Estuary in Spring. South China Fisheries Science, 2010, 6, 59-67. doi:10.3969/j.issn.1673-2227.2010.01.011.
29. Zou, Z., Chen, K., Shi, Z., Guo, Y., Ye, J. Object Detection in 20 Years: A Survey. Proceedings of the IEEE 2023, 111, 257-276. https://doi.org/10.1109/JPROC.2023.3238524