Aquaculture International (2023) 31:2513–2530
https://doi.org/10.1007/s10499-023-01095-7

Fish detection method based on improved YOLOv5

Lei Li[1] · Guosheng Shi[1] · Tao Jiang[1]

Received: 22 November 2022 / Accepted: 15 March 2023 / Published online: 31 March 2023

Abstract
In the field of fisheries, detecting the distribution of fish underwater is an important task for achieving accurate bait feeding. However, the current deep neural networks for fish detection are significantly more computationally intensive than previous methods due to their increased network depths. Additionally, drawbacks such as the difficulty of balancing accuracy and real-time performance limit the deployment of these algorithms in fishery end devices. To address this problem, this paper proposes an improved You Only Look Once version 5 (YOLOv5)-based underwater fish detection method called RC_YOLOv5. First, the Res2Net residual structure is introduced to represent multiscale features at a finer granularity and increase the perceptual field of the network while reducing the computational power of the model. Second, a coordinate attention mechanism is introduced to suppress the interference of the background and help the network locate its target more accurately. Finally, coordinate attention is embedded into the tail of Res2Net to form a residual attention structure, and this structure is used to replace the original bottleneck structure in the YOLOv5 model to improve its accuracy. Experiments show that the proposed model has good performance on a self-built fish dataset, reaching 95.7% and 95.4% precision and mean average precision (mAP), respectively. Compared with those of the original model, the precision of the proposed approach improves by 1.6%, the mAP improves by 0.6%, the number of computations is reduced by 22.2%, the model size is reduced by 23.5%, the detection rate reaches 263 frames per second (FPS) and the performance is better than that of other mainstream detection models. This method enables accurate and rapid fish detection in fisheries.

Keywords Fish detection · YOLOv5 · Residual structure · Attention mechanism

Introduction

Fisheries, important components of aquaculture, provide large sources of protein for humans, and they provide at least 15% of the average per capita consumption of animal protein for more than half of the global population (Béné et al. 2015). According to

Handling Editor: Michael Hartnett

✉ Lei Li
lilei0064@sina.com

[1] School of Mechanical Engineering, Jiangsu University of Science and Technology, Zhenjiang 212000, China

statistics, human beings directly consume at least half of the global annual production of fisheries, while the trends of fishery production and consumption are also increasing each year (Yang et al. 2020). However, with the rapid development of fisheries and the expansion of the farming scale, some challenging problems have emerged. In large fishery farms, the underwater distributions of fish populations are very uneven. Irrational bait feeding can result in large amounts of bait residue, which can have an impact on the feeding behavior of fish and even kill them (Harsij et al. 2020). By monitoring the distribution of fish in real time, a reasonable bait feeding plan can be formulated for different underwater areas, which can effectively reduce bait waste and improve economic efficiency (Hu et al. 2021). Therefore, a detection method is proposed in this paper to detect fish in fisheries.

Traditional fish detection methods are based on machine learning methods, such as sonar technology (Boswell et al. 2008), light detection and ranging (LIDAR) technology (Zavalas et al. 2014; Jalali et al. 2015) and RGB image vision technology. Among them, sonar technology and LIDAR technology have high detection accuracy, but their disadvantages, such as high equipment costs and large sizes, limit their application in fishery farms. RGB imaging technology has the advantages of simple operation and light system weight, and it can detect fish targets based on features such as color, texture and geometry. For example, color and shape features were used to identify fish under RGB image cameras (White et al. 2006). Combining texture and shape features to classify fish species, an average accuracy of 92% was obtained (Spampinato et al. 2010). Contour feature matching methods were used to identify fish in fish tanks (DahJye et al. 2004). However, all these methods are essentially shallow learning methods, and none of them are efficient. In summary, in previous studies, fish features were particularly dependent on manual production and extraction, which can seriously affect the accuracy of fish detection if relevant features are ignored. With the development of deep topologies and artificial intelligence (AI) technology, these traditional detection methods are gradually being replaced by deep learning methods.

Deep learning, as an important branch of machine learning, has made remarkable achievements in image processing (target detection, semantic segmentation, etc.) in recent years, and people are becoming increasingly inclined to use deep learning-based target detection methods to solve most problems in many fields. Deep learning-based target detection methods are divided into two main categories. One includes region-based two-stage detection models, such as R-CNN (Girshick et al. 2014), SPP-Net (He et al. 2015) and Fast R-CNN (Girshick 2015). These algorithms first generate several regions that may contain the target in the first step and then perform sample classification on these regions in the second step. However, this class of methods suffers from the disadvantage that the process of proposing candidate regions is complicated. In response, an alternative regression-based single-stage detection model has emerged; this approach directly abandons the candidate region generation step, so it is much faster than the two-stage detection methods in terms of detection speed. Common single-stage models include the single-shot detector (SSD) (Liu et al. 2016), EfficientDet (Tan et al. 2020), RetinaNet (Lin et al. 2017) and the You Only Look Once (YOLO) series (Redmon and Farhadi 2018; Bochkovskiy et al. 2020) have been used by many scholars in the field of fish detection (Zhang et al. 2016; Rekha et al. 2020; Sung et al. 2017; Liu et al. 2018). However, these deep learning-based detection models greatly increase the number of required computational parameters as the network depth expands, making it difficult to deploy end devices to achieve detection tasks in complex underwater environments.

In response to the above problems, the MobileNet series of lightweight networks was proposed to greatly contribute to lightweight model improvements (Howard et al. 2017).

Numerous scholars have fused the MobileNet family of networks with target detection algorithms to reduce the computational effort required by their models and achieve high-speed detection. For example, MobileNetV1 was used to replace the original backbone network of YOLOv3 to detect fish (Cai et al. 2020). This reduced the computational effort of the model and increased its detection speed, but at the same time, the loss of a large number of computations led to a decrease in detection accuracy. Therefore, the authors compensated for the loss of accuracy by reselecting the feature maps according to the target scale of the images and the changes observed in the perceptual field. Similarly, MobileNetV2 was used in combination with the SSD detection model to reduce the computational effort required by the model and improve the frame rate of detection (Cao et al. 2020). Furthermore, the authors employed a feature pyramid network to compensate for the accuracy degradation due induced by the use of low computational power. In a recent study, the latest MobileNet version (MobileNetV3) was used to improve the YOLOv4 model (Zhao et al. 2022). Different from the ideas of the previous two scholars, the author used deformable convolution to enhance the target extraction ability of the model and thus improve its detection accuracy. The results showed that the proposed model could better detect dead fish in real time.

In summary, the current lightweight improvement methods have certain shortcomings in the field of fish detection. On the one hand, the previously developed YOLO series models are limited by perceptual field variations and multiscale features and perform poorly in fish detection scenarios. On the other hand, the introduction of lightweight networks to the target detection model can greatly reduce the computational effort required by the model and can help the model achieve a greater advantage in real time. However, at the same time, it can cause the model to have a significantly insufficient ability to extract image features, which makes it difficult to guarantee the detection accuracy of the model.

To meet the requirements of high accuracy and low computational effort for fish detection in the special underwater environments of fishery farms, the latest version of the YOLOv5 model is selected for optimization and improvement in this paper. The model uses Res2Net, a structure that is capable of multiscale feature reuse, to reduce the required computational effort. Coordinated attention is introduced to help the network find the location information of the target. By fusing the Res2Net structure with embedded coordinated attention to compensate for the decrease in accuracy caused by the reduction in the number of model computations, real-time fish detection is finally achieved.

Methods

YOLOv5 model

By controlling the size of the network, YOLOv5 can be divided into four models with different complexity levels: YOLOv5x, YOLOv5l, YOLOv5m and YOLOv5s, which utilize the same network architecture in all four versions. In this case, the network structure is controlled by a depth factor (the size of n in C3_n) and a width factor (the number of convolutional kernels). For example, the depth and width factors of the YOLOv5s model are 0.33 and 0.50, respectively. Its first C3 module in the backbone network has an *n* value of 1, and the number of convolutional kernels is 32. For the YOLOv5m model, its depth and width factors are both 1.0, so it has three times the depth and two times the width of the

YOLOv5s model. Therefore, n is extended to 3, and the number of convolutional kernels reaches 64.

Among these four versions, the model with the smallest size and the highest speed is YOLOv5s. This paper improves upon the YOLOv5s model, whose structure is divided into three main parts: a backbone network (backbone), a bottleneck layer network (neck) and a detection head (head). The backbone network consists of a convolutional layer with 6×6 convolutional kernels (Conv in red boxes), a standard convolutional module (Conv), a C3 module and a spatial pyramid pooling module (SPPF). The bottleneck layer network includes the standard convolution module (Conv), an upsampling operation (Upsample), a splicing operation (Concat) and a C3 module. The final feature maps with three different sizes are output to the detection head (Detect) for target detection. Among them, “True” and “False” in the bottleneck structure within the C3 module indicate whether the add operation is executed or not, and the value n in C3_n indicates the number of times the bottleneck module has been executed. The specific structure of this network is shown in Fig. 1.

Res2Net residual structure

In 2021, the Res2Net model was proposed on the basis of ResNet (Gao et al. 2021), whose basic structure is shown in Fig. 2. First, a new parameter s is introduced to represent the division of the input into s groups. Then, for the output features of the first 1×1 convolutional layer, assuming that the number of channels is n, Res2Net divides them equally into s groups of features according to the number of channels, and the number of channels in each group of features is w; that is, $n = s \times w$. For example, in Fig. 2, the features are divided into 4 groups, and each group of features after equal division is denoted as x_i, where $i \in \{1, 2 \dots \dots s\}$. Then, the 3×3 convolution operation of the second layer is denoted as K_i. For each group of features x_i obtained after grouping, each group corresponds to convolution operation K_i except for the first group, which does not

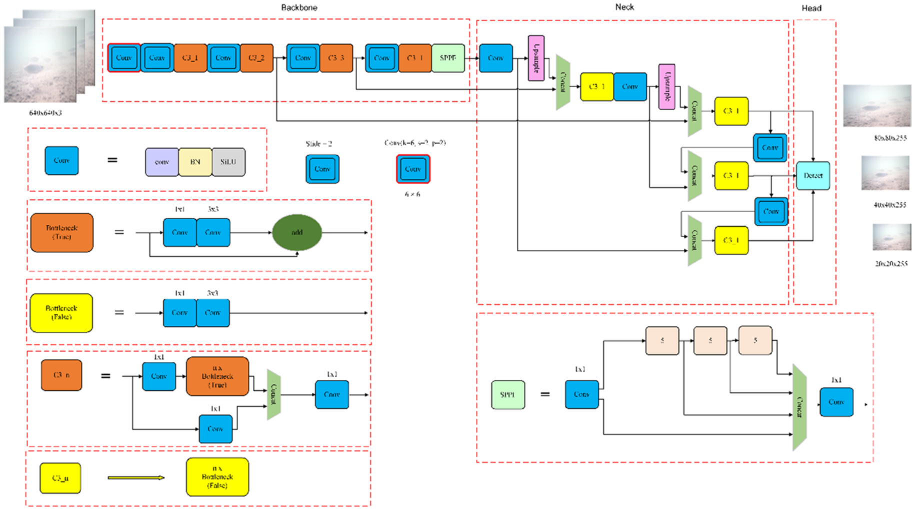
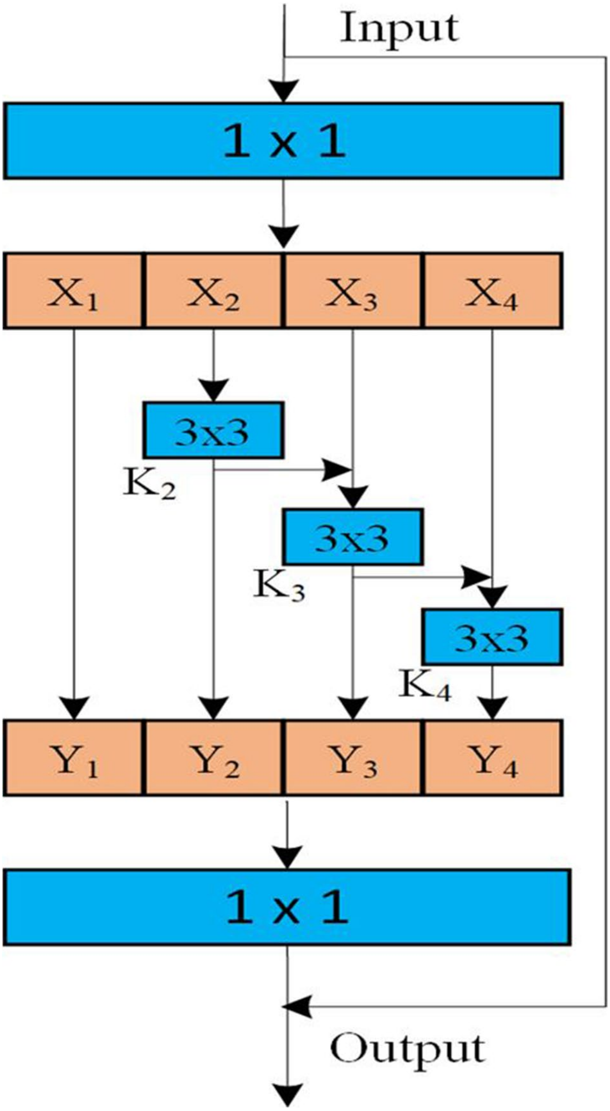


Fig. 1 YOLOv5s network structure

Fig. 2 Res2Net residual structure



perform the convolution operation, and y_i is the output after convolution operation K_i. Starting from the third group, each K_i operation associates the y_{i-1} of the previous layer with the current x_i residuals, thus increasing the perceptual field of the model. This continues until the last group. In short, y_i can be expressed as shown in Eq. (1).

$$y_i = \begin{cases} \mathrm{x}_i & i = 1 \\ \mathrm{K}_i(x_i) & i = 2 \\ \mathrm{K}_i(y_{i-1} + x_i) & 2 < i \leq 4 \end{cases} \tag{1}$$

Finally, the output s-group feature maps are stitched and fed into a 1×1 convolution for feature fusion. Through this cascading method, Res2Net represents multiscale features at a finer granularity and obtains feature combinations with different receptive fields. The advantage of incorporating this structure into the YOLOv5 model is that while reducing the number of model calculations, it can better help the model reuse features, thereby enhancing the feature extraction ability of the model.

Coordinate attention (CA)

Currently, many lightweight networks mostly use squeeze-and-excitation (SE) attention (Hu et al. 2018). However, this attention only considers the information between channels and ignores the location available information. Although the convolutional block attention module (CBAM) (Woo et al. 2018), which appeared later, also tries to extract location information via convolution after reducing the number of channels, the convolution operation only focuses on local relations and fails to pay attention to long-distance relations. In response, a coordinated attention mechanism was proposed (Hou et al. 2021). This is a new attention module that helps the network focus on a large range of location information without requiring too much computational effort. Its structure is shown in Fig. 3; specifically, it encodes channel relations and long-range relations through the following two steps.

Coordinate information embedding

The global pooling method is typically used for globally encoding channel attention and encoding spatial information, but this approach compresses the global information into a scalar, which makes it difficult to retain important position information. The global pooling operation is then decomposed according to the following formula:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{2}$$

Global pooling is transformed into an encoding operation for two 1-dimensional vectors. For an input X, using the pooling kernels (H,1) and (1,W) to encode horizontal and vertical features, respectively, the output of the cth-dimensional feature can be expressed by Eqs. 3 and 4.

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < \mathrm{W}} x_c(h, i) \tag{3}$$

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < \mathrm{H}} x_c(j, w) \tag{4}$$

The above transformation integrates features from different directions to obtain a pair of direction-aware feature maps. Compared with the global pooling and compression approach, the advantage of this method is that focusing on the long-distance relationships in a single direction while preserving the spatial information in the other direction helps the network achieve accurate target localization.

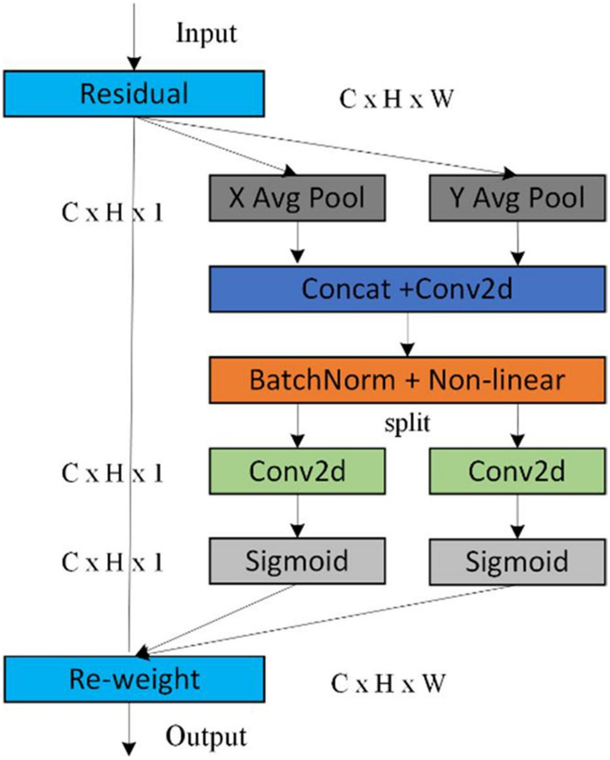


Fig. 3 CA

CA generation

The abovementioned transformations enable good access to the global sensory field and encode precise positional information. To make better use of the resulting representations, the authors proposed a 2nd transformation, called coordinate attention generation.

First, the outputs of Eqs. 3 and 4 are concatenated, and feature transformation is performed using a 1×1 convolution, batch normalization (BN) and nonlinear activation:

$$f = \delta(F_1([z^h, z^w])) \tag{5}$$

$f \in \mathbb{R}^{C/r \times (H+W)}$ is an intermediate feature containing both horizontal and vertical spatial information, and r is the scaling factor. No intense fusion is performed on the features in the two directions, and the main purpose of the connection is to perform a uniform BN operation. Subsequently, f is divided into two independent features $f^h \in \mathbb{R}^{C/r \times W}$ and $f^w \in \mathbb{R}^{C/r \times H}$, and the feature transformation operation is performed using two other 1×1 convolution and sigmoid functions to make their dimensions consistent with those of the input X:

$$g^h = \sigma(F_h(f^h)) \tag{6}$$

$$g^w = \sigma(F_w(f^w)). \tag{7}$$

The outputs g^h and g^w are combined into a weight matrix for computing the overall output, and this final output can be written as

$$y_c = (i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j). \tag{8}$$

Through the above two steps, CA makes each weight contain interchannel information, horizontal spatial information and longitudinal spatial information, which not only helps the network accurately locate the target information but also enhances its recognition ability.

Improved YOLOv5 model

The latest YOLOv5 model has undergone several structural optimization processes and has formed a more mature system, achieving good performance in terms of both detection accuracy and detection speed. In particular, the YOLOv5s model is the smallest in the YOLO series, so it has been able to meet the real-time requirements of applications with respect to detection speed. In this paper, we try to optimize the YOLOv5s model in terms of the number of computational parameters and accuracy by using a lightweight structure under the premise of ensuring the accuracy of the model.

Incorporating the Res2Net residual structure into the YOLOv5 model may result in accuracy degradation while reducing the required computational effort. Therefore, embedding coordinate attention into the tail of the Res2Net residual structure and rescaling the original features in the channel dimension can improve the accuracy at the cost of a small number of computations. The composed structure is referred to as the RC residual attention structure, as shown in Fig. 4a. The RC structure is used to replace the bottleneck structure (shown in Fig. 4b) within all C3 modules in YOLOv5.

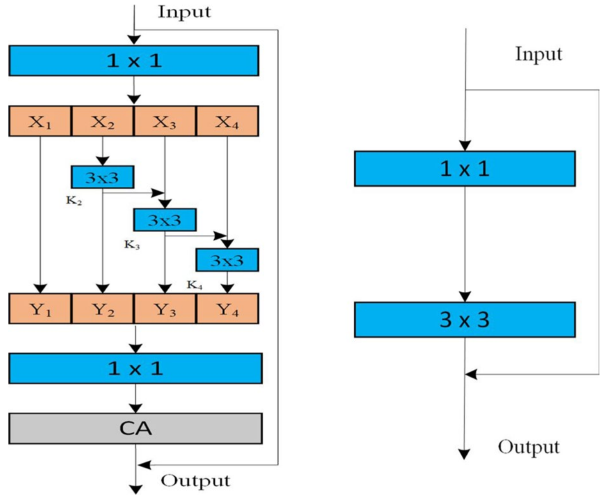
The proposed model is named RC_YOLOv5. The specific structure of the improved model is shown in Fig. 5, where the input of each layer is derived from the output of the previous layer. Take the standard convolution module Conv(64,3,2) of the second layer as an example. The value "64" indicates the number of output channels and is used as the input for the next layer. The values "3" and "2" denote the convolution kernel size and convolution step size, respectively. The C3 module obtained after embedding the RC structure is represented by C3_RC_*n*, and the value *n* indicates the number of times the RC structure is executed. Finally, the model performs target detection at layers 17, 20 and 23 for three different sizes of feature maps.

Dataset and evaluation parameters

Data acquisition

The experimental samples were collected from April to July 2022 at the bottom of a lake at Jiangsu University of Science and Technology, Zhenjiang, Jiangsu Province (Fig. 6a). The collection equipment (Fig. 6b) included an underwater monitoring device (model HK90) developed by Shenzhen Haxter. The device had an image resolution of 1920 pixels × 1080 pixels and a frame rate of 25 FPS, and the focal length of the camera was adjusted between 30 and 50 cm depending on the turbidity of the water body. To ensure condition diversity, underwater fish video data were collected on sunny days, on cloudy days, on rainy days

Fig. 4 RC residual attention structure substitution



(a) RC residual attention structure (b) Bottleneck structure of YOLOv5

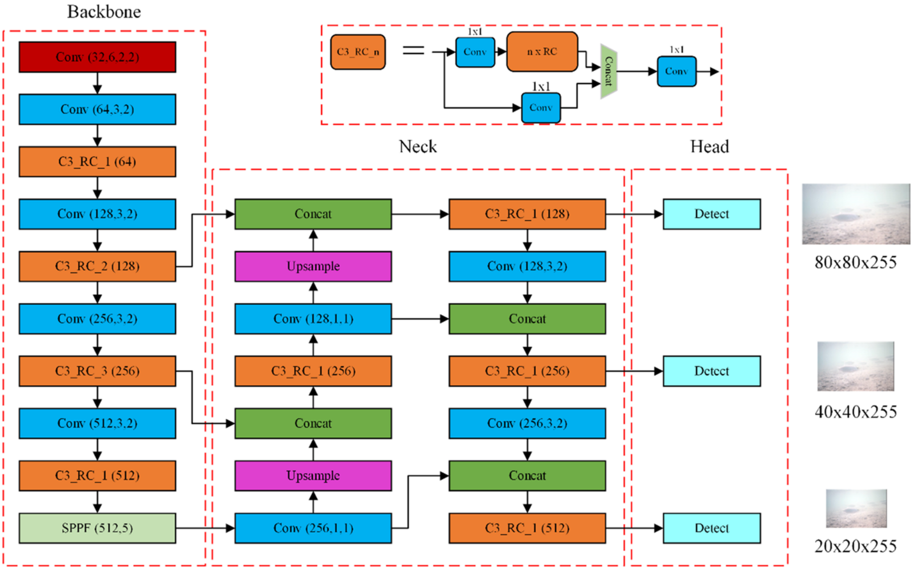


Fig. 5 RC_YOLOv5 model

and in the evening. The captured fish video samples were saved in AVI format to a memory card, which was subsequently read by a computer to obtain fish images in a frame-by-frame manner. The captured images were blurred due to the turbidity of the water body and the acquisition conditions. Therefore, 2627 fish images were initially obtained by selecting the clearest images and eliminating the useless ones.

A single underwater image sample (Fig. 7a) tends to cause model overfitting during the training process, which further leads to the model not having better accuracy. To reduce

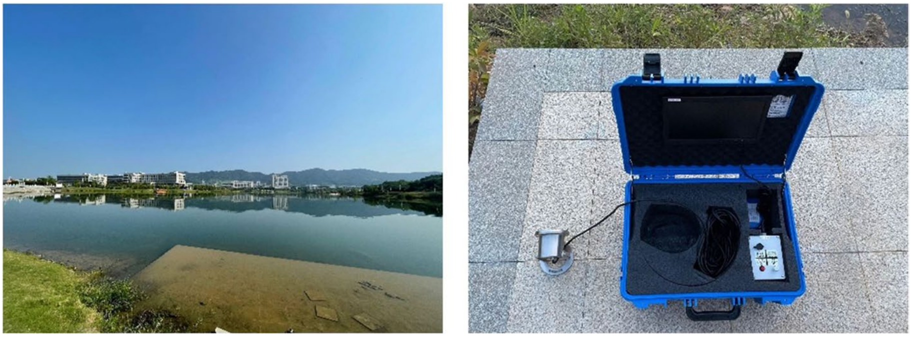

(a) Collection site (b) Acquisition equipment

Fig. 6 Sample collection

the probability of this phenomenon, we added two additional sources of data samples. One source was a collection of 1000 fish images taken in a laboratory tank using a smart device (Fig. 7b), and the other source contained 500 fish images from the internet (Fig. 7c). The diversity of data samples has thus been greatly increased to improve the generalization ability of the model. To meet the requirements of the training data volume, we further performed geometric operations such as random rotation, random cropping and random masking on the images from these three sources, resulting in 4868 image samples.

To facilitate model training, LabelImg software was used to label the target objects in each image sample using 2D rectangular boxes, and the labeled files were saved as XML files in PASCAL VOC format. After all the images were labeled, the label data in PASCAL VOC format were converted into txt label text conforming to the YOLO model training process using python scripts. Finally, all the images and label files were disordered and divided into training and test sets at a ratio of 8:2. At this point, the whole fish dataset has been produced.

Evaluation parameters

The performance metrics used in this paper mainly include precision, recall, and mean average precision (mAP). Among them, the precision represents the proportion of samples that are actually positive among all samples predicted to be positive, and recall represents the proportion of samples that are predicted to be positive among all samples that are actually positive. Equations (9) and (10) give the formulae for calculating the precision and recall metrics, respectively.

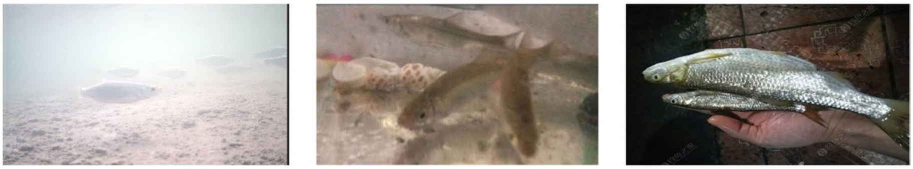

(a) Underwater image (b) Laboratory image (c) Internet image

Fig. 7 Sample data acquired from different sources

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

In the above equation, TP is the number of positive samples correctly classified as positive; FN is the number of positive samples incorrectly classified as negative; and FP is the number of negative samples incorrectly classified as positive.

AP is the integrated area enclosed by the precision-recall (PR) curve and the coordinate axis corresponding to a single class of detected targets, which is the average accuracy. The mAP is the mean value of the summation of the average detection accuracy for each category. *k* denotes the number of detected target categories, and *K* is 1 in the experiments in this paper. The AP and mAP are calculated as shown in Eqs. (11) and (12), respectively.

$$AP = \int_0^1 P(R)dR \tag{11}$$

$$mAP = \frac{1}{K}\sum\nolimits_{K=1}^{K} AP \tag{12}$$

In addition, gigaflops per second (GFLOPs) are used to measure the parametric size of the model, and frames per second (FPS) are used to measure the detection speed of the model.

Results and discussion

Experimental results of the improved model

The experimental environment utilized in this paper contains the Windows 10 operating system, an Intel Core i5 12490f processor (CPU), an NVIDIA GEFORCE RTX3060 graphics card (GPU), 12 GB of display memory, 16 GB of system running memory, the Pytorch deep learning framework and the Pycharm development platform. During the training period, the batch size is 4, the weight decay is 0.005, the momentum is 0.9, the initial value of the learning rate is 0.001 and the model training period (number of epochs) is 300 iterations.

The mAP curve and loss curve produced during the training process of the improved model are shown in Fig. 8. From the training graph, it can be seen that the loss function value decreases smoothly as the number of iterations increases. When the number of iteration epochs reaches 200, the learning efficiency of the model almost saturates, and the mAP basically stops changing. The training plots show that the improved model has good accuracy on the self-built fish dataset.

To verify whether the performance of the improved model is actually improved, the performance of the proposed model is compared with that of the original model and previous-generation versions. The proposed model improves the precision values by 2.7% and 2.4%, the recall values by 11.1% and 10.2% and the mAP values by 5.2% and 3.6% over those of

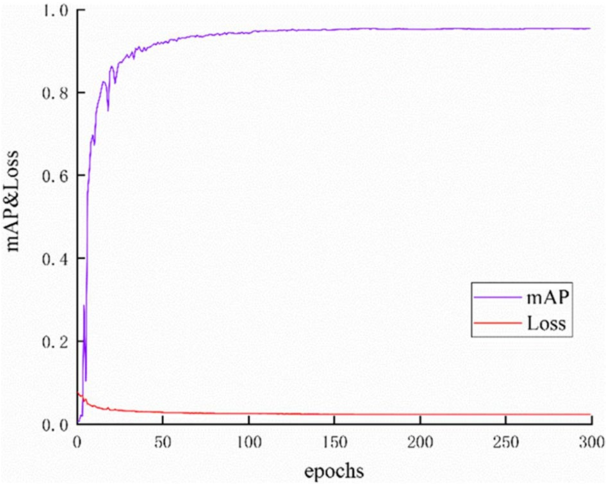


Fig. 8 Training graph of the improved model

YOLOv3 and YOLOv4, respectively. Compared to YOLOv5, the precision of the proposed model is improved by 1.6%, and the mAP is improved by 0.6%, as shown in Table 1.

Furthermore, the proposed model is 22.2% less computationally intensive (3.7 GFLOPs) and 23.5% smaller (3.2 MB) than the original model, as shown in Table 2.

To further verify that the model proposed in this paper outperforms the original model, several images in the validation set are randomly selected for a detection effect comparison. Figure 9 shows the difference between the detection effects of the original model and the model proposed in this paper, where the green circles indicate targets that are missed, and the black circles indicate targets that are falsely detected.

Regarding the detection effect, it can be seen that the original model does not detect the small target on the right side of the first test image, the original model falsely detects the blank area as the target in the second test image and the original model has both missed and false detections in the third test image. In contrast, the proposed model can better identify the targets that are present in the test images and effectively reduce the rates of missed detections and false detections, thus verifying that the proposed model outperforms the original model in terms of performance and its detection effect.

Table 1 Comparison of model training results

Model	Precision/%	Recall/%	mAP/%
YOLOv3	93	76.9	90.2
YOLOv4	93.3	77.8	91.8
YOLOv5	94.1	88.1	94.8
RC_YOLOv5	95.7	88	95.4

Table 2 Comparison of the parameters before and after implementing the proposed model

Model	Parameters/GFLOPs	Model size/MB
YOLOv5	16.6	13.6
RC_YOLOv5	12.9	10.4

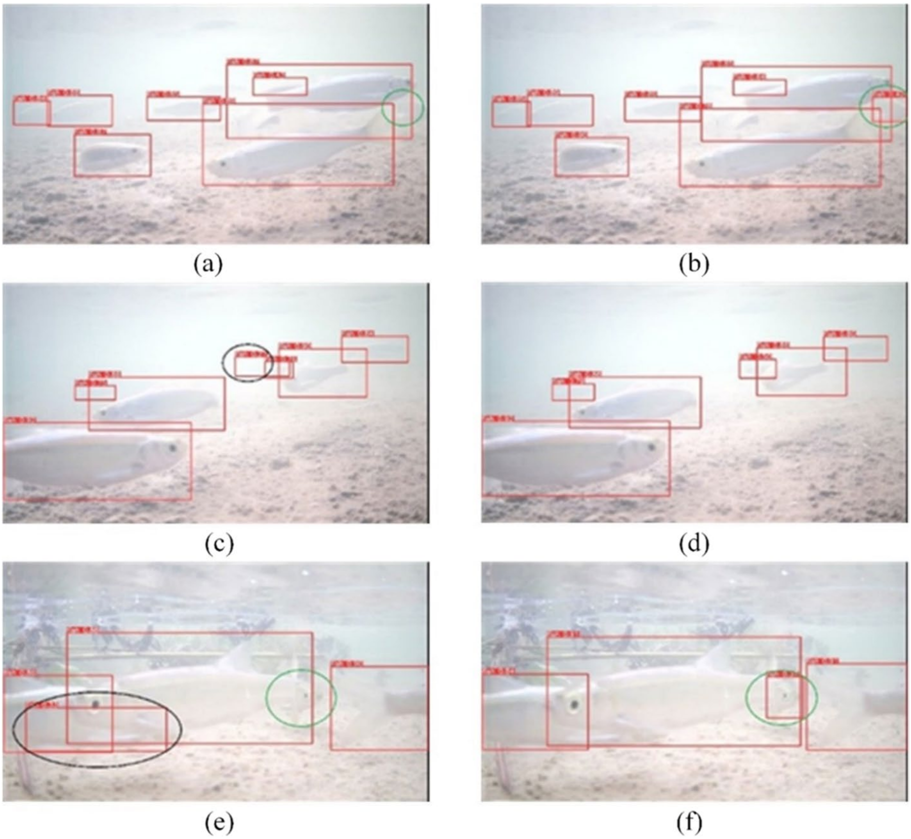


Fig. 9 The fish detection results of the original model and proposed model. (**a**, **c**, **e**) correspond to the original model, and (**b**, **d**, f) correspond to the proposed model

Comparison among different attentional mechanisms

To verify the superiority of CA fused with the Res2Net residual structure, the Res2Net residual structure is chosen as the base module to incorporate different attention mechanisms (SE attention (Hu et al. 2018), the CBAM (Woo et al. 2018), CA (Hou et al. 2021), and ECA (Wang et al. 2020)), and these mechanisms are used as new residual attention structures to replace the bottleneck in all C3 modules for comparison experiments. The experimental results are shown in Table 3.

From the experimental results, we can see that the performance achieved by the models improves after introducing different attention mechanisms. Among them, the model with the ECA mechanism only slightly improves the recall and mAP values, and the three other types of attention, the CBAM, SE and CA, improve the comprehensive performance of the model the most and produce the highest mAP, recall and precision values, respectively. However, when the mAP values are basically the same, CA improves the accuracy of the model by 1.6%, and the recall R is only reduced by 0.1% compared with that of the original model, which has better overall performance. These comparative experiments show that the selection of CA provides an advantage over other attention types on self-built datasets.

Table 3 Comparison of the results obtained in fusion experiments with different attention

Model	Precision/%	Recall/%	mAP/%
Res2Net + ECA	94.1	88.6	95.2
Res2Net + CBAM	94.9	88.6	95.5
Res2Net + SE	94.4	89.3	95.4
Res2Net + CA	95.7	88	95.4

Ablation experiments

To verify the effectiveness of the RC structure that replaces all bottleneck structures, based on the original model, the model performance is compared with that of variants replacing the bottleneck with only the Res2Net residual structure at different positions and replacing the bottleneck with the RC structure at different positions. Six sets of ablation experiments (Experiments 2–7) are carried out, as shown in Table 4.

From the results of Experiments 2 and 3, we can see that after adding the Res2Net residual structure to the backbone and neck parts of the model, different degrees of improvement are achieved in the recall and mAP metrics, but the precision decreases. After replacing all the bottleneck structures with this structure (Experiment 4), the model has significantly improved precision, recall and mAP metrics, and the required computational effort is also minimized. This shows that the introduction of the Res2Net residual structure can reduce the computational effort of the model, but it does not completely guarantee the accuracy of the model. In addition, comparing Experiment 4 with Experiment 7, we find that the model has more room for improvement in precision, so the model of Experiment 4 is not adopted as the final model.

By comparing Experiment 2 with Experiment 5, Experiment 3 with Experiment 6 and Experiment 4 with Experiment 7, it can be seen that the computational effort required by the model after adding CA only increases slightly, but the improvement in precision is very obvious. On the other hand, this also sacrifices some recall performance, but the recall performance (Experiment 5, Experiment 6 and Experiment 7) remains basically the same as that of the original model (Experiment 1), and better mAP values are obtained. This also verifies that embedding CA into the tail of the Res2Net residual structure can indeed rescale the original features in the channel dimension and can improve the model accuracy at the cost of a small number of computations. The precision and mAP of the model after replacing all the bottleneck structures with RC structures are as high as 95.7% and

Table 4 Comparison of ablation test results

Model	Precision/%	Recall/%	mAP/%	Parameters/ GFLOPs
1. The original model	94.1	88.1	94.8	16.6
2. Backbone (Res2Net)	93.8	89.3	95.4	14.9
3. Neck (Res2Net)	92.5	89.7	94.7	13.8
4. ALL (Res2Net)	94.8	88.8	95.6	12.8
5. Backbone (RC)	95.3	88.2	95.3	15
6. Neck (RC)	94.3	87.8	94.9	13.9
7. ALL(RC)	95.7	88	95.4	12.9

95.4%, respectively. Therefore, the ablation experiments show that the best performance is achieved after replacing all the bottleneck structures of the original model with RC structures.

Comparison among different target detection algorithms

To further verify the effectiveness of the models proposed in this paper, the SSD algorithm (Liu et al. 2016), EfficientDet algorithm (Tan et al. 2020), RetinaNet algorithm (Lin et al. 2017) and Faster R-CNN algorithm (Ren et al. 2017) are selected for experimental comparisons with the models proposed in this paper. The first three algorithms and the YOLO algorithm are the most typical single-stage target detection algorithms, and the Faster R-CNN algorithm is a two-stage target detection algorithm. The mAP plots produced by the five models during training are given in Fig. 10, and the experimental results are shown in Table 5.

It can be seen from the experimental results that RetinaNet and EfficientDet are close to the proposed model in terms of precision and mAP and even superior to the proposed model with respect to recall, but their detection speeds are only 43 FPS and 46 FPS, respectively, so they cannot meet the real-time requirements of detection tasks. Although the recall of the Faster R-CNN is as high as 90.5%, its scores are the lowest among those of the five models in terms of precision and detection speed. Compared with the above three models, although the detection speed of the SSD reaches 83 FPS and it can satisfy the real-time requirements of detection, there is still a large gap between the recall and mAP values of the SSD and the proposed model. Finally, compared with the first four models, the proposed model has the highest values for the precision and mAP metrics, reaching 95.7% and 95.4%, respectively. The detection speed is 6.1 times that of EfficientDet, 5.7 times that of RetinaNet, 26.3 times that of the Faster R-CNN, and 3.1 times that of the SSD, which indicates that the model proposed in this paper has higher detection accuracy and a faster detection speed than the current mainstream object detection algorithms; that is, it has better performance and can perform fish detection in real time.

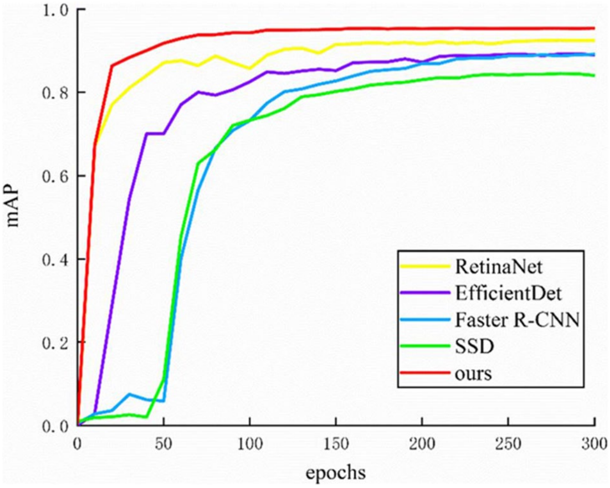


Fig. 10 mAP curves of different object detection algorithms

Table 5 Performance comparison among different target detection algorithms

Model	Precision/%	Recall/%	mAP/%	FPS
RetinaNet	93.6	83.9	92.4	46
EfficientDet	94.6	81.7	92.5	43
Faster R-CNN	55	90.5	88.3	10
SSD	93.7	73.1	85.4	83
RC_YOLOv5	95.7	88	95.4	263

Conclusion

In this paper, we propose an improved YOLOv5-based underwater fish detection method called RC_YOLOv5. In the proposed method, first, to reduce the number of model computations and enhance the feature extraction capability of the model, the Res2Net residual structure is introduced to represent multiscale features at a finer granularity and increase the perceptual field of the model. Second, to suppress background interference and enhance the recognition ability of the model, a CA mechanism is introduced to help the model locate its target more accurately, and the superiority of CA over other types of attention is verified through comparison experiments. Finally, CA is embedded in the tail of Res2Net to form a RC residual attention structure for improving the accuracy of the model, and the effectiveness of this structure in terms of replacing the original bottleneck structure in the YOLOv5 model is verified via ablation experiments. The experiments show that the improved model yields large precision, recall and mAP improvements over the previous two versions (YOLOv3 and YOLOv4). Compared with the original YOLOv5, the improved model achieves improved accuracy and effectively reduces the missed detection rate and false detection rate of the model. At the same time, the number of GFLOPs is reduced, and the size of the model is reduced. Compared with the current mainstream target detection algorithms, the improved model has better performance in terms of both detection accuracy and detection speed. However, there is still room to optimize the improved model in terms of computational volume and accuracy. In the future, the network model will be further optimized to improve its accuracy while making the model more lightweight for the subsequent deployment of mobile devices to implement underwater detection tasks.

Author contribution **LL:** conceptualization, methodology, investigation, writing—original draft. **GS:** thesis framework, supervision, writing—review and editing. **TJ:** data acquisition, process supervision.

Funding This work was supported by the National Key Research and Development Program-network collaborative manufacturing and intelligent factories, special item (grant nos. 2020YFB1712600 and 2020YFB1712602), as well as the National Defense Basic Scientific Research Funding Project (grant no. JCKY2021414B011).

Data availability The data or code presented in this study are available from the corresponding author upon request.

Declarations

Conflict of interest The authors declare no competing interests.

References

Bochkovskiy A, Wang C-Y and Liao H-Y M (2020) YOLOv4: optimal speed and accuracy of object detection. ArXiv abs/2004.10934. https://doi.org/10.48550/arXiv.2004.10934

Boswell KM, Wilson MP, Cowan JH (2008) A semiautomated approach to estimating fish size, abundance, and behavior from dual-frequency identification Sonar (DIDSON) data. N Am J Fish Manag 28(3):799–807. https://doi.org/10.1577/M07-116.1

Cai K, Miao X, Wang W et al (2020) A modified YOLOv3 model for fish detection based on MobileNetv1 as backbone. Aquac Eng 91:102117. https://doi.org/10.1016/j.aquaeng.2020.102117

Cao S, Zhao D, Liu X et al (2020) Real-time robust detector for underwater live crabs based on deep learning. Comput Electron Agric 172:105339. https://doi.org/10.1016/j.compag.2020.105339

DahJye LBSR, Dennis S et al (2004) Contour matching for a fish recognition and migration-monitoring system. Brigham Young Univ 5606:37–48. https://doi.org/10.1117/12.571789

Béné C, Barange M, Subasinghe R et al (2015) Feeding 9 billion by 2050–Putting fish back on the menu. Food Sec 7:261–274. https://doi.org/10.1007/s12571-015-0427-z

Gao S-H, Cheng M-M, Zhao K et al (2021) Res2Net: A new multi-scale backbone architecture. IEEE Trans Pattern Anal Mach Intell 43(2):652–662. https://doi.org/10.1109/TPAMI.2019.2938758

Girshick R (2015) Fast R-CNN. 2015 IEEE Int Conf Comp Vis (ICCV):1440–1448. https://doi.org/10.1109/ICCV.2015.169

Girshick R, Donahue J, Darrell T et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conf Comput Vis Patt Recognit 580–587. https://doi.org/10.1109/CVPR.2014.81

Harsij M, Gholipour Kanani H, Adineh H (2020) Effects of antioxidant supplementation (nano-selenium, vitamin C and E) on growth performance, blood biochemistry, immune status and body composition of rainbow trout (Oncorhynchus mykiss) under sub-lethal ammonia exposure. Aquaculture 521:734942. https://doi.org/10.1016/j.aquaculture.2020.734942

He K, Zhang X, Ren S et al (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 37(9):1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824

Hou Q, Zhou D, Feng J (2021) Coordinate attention for efficient mobile network design. 2021 IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR):13708–13717. https://doi.org/10.1109/CVPR46437.2021.01350

Howard AG, Zhu M, Chen B et al (2017) MobileNets: efficient convolutional neural networks for mobile vision applications. ArXiv abs/1704.04861. https://doi.org/10.48550/arXiv.1704.04861

Hu X, Liu Y, Zhao Z et al (2021) Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network. Comput Electron Agric 185:106135. https://doi.org/10.1016/j.compag.2021.106135

Hu J, Shen L, Sun G (2018) Squeeze-and-Excitation Networks. 2018 IEEE/CVF Conf Comput Vis Pattern Recognit 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

Jalali MA, Ierodiaconou D, Monk J et al (2015) Predictive mapping of abalone fishing grounds using remotely-sensed LiDAR and commercial catch data. Fish Res 169:26–36. https://doi.org/10.1016/j.fishres.2015.04.009

Lin TY, Goyal P, Girshick R et al (2020) Focal loss for dense object detection. IEEE Trans Pattern Anal Mach Intell 42(2):318–327. https://doi.org/10.1109/TPAMI.2018.2858826

Liu W, Anguelov D, Erhan D et al (2016) SSD: single shot multibox detector. 14th European conference on computer vision, ECCV 2016, October 8–16, 2016 9905 LNCS:21-37. https://doi.org/10.1007/978-3-319-46448-0_2

Liu S, Li X, Gao M et al (2018) Embedded online fish detection and tracking system via YOLOv3 and parallel correlation filter. Oceans 2018 MTS/IEEE Charleston 1–6. https://doi.org/10.1109/OCEANS.2018.8604658

Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. ArXiv abs/1804.02767. https://doi.org/10.48550/arXiv.1804.02767

Rekha BS, Srinivasan GN, Reddy SK et al (2020) Fish detection and classification using convolutional neural networks. Comput Vis Bio-Inspired Comput 1221–1231. https://doi.org/10.1007/978-3-030-37218-7_128

Ren S, He K, Girshick R et al (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031b

Spampinato C, Giordano D, Salvo RD et al (2010) Automatic fish classification for underwater species behavior understanding. Analysis and retrieval of tracked events and motion in imagery streams 45–50. https://doi.org/10.1145/1877868.1877881

Sung M, Yu SC, Girdhar Y (2017) Vision based real-time fish detection using convolutional neural network. Oceans 2017 - Aberdeen 1–6. https://doi.org/10.1109/OCEANSE.2017.8084889

Tan M, Pang R, Le QV (2020) EfficientDet: scalable and efficient object detection. 2020 IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR) 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079

Wang Q, Wu B, Zhu P et al (2020) ECA-Net: efficient channel attention for deep convolutional neural networks. 2020 IEEE/CVF Conf Comput Vis Pattern Recognit (CVPR):11531–11539. https://doi.org/10.1109/CVPR42600.2020.01155

White D J, Svellingen C, Strachan NJC (2006) Automated measurement of species and length of fish by computer vision. Fish Res 80(2):203–210. https://doi.org/10.1016/j.fishres.2006.04.009

Woo S, Park J, Lee J-Y et al (2018) CBAM: convolutional block attention module. Comput Vis – ECCV 2018 3–19. https://doi.org/10.1007/978-3-030-01234-2_1

Yang X, Zhang S, Liu J et al (2020) Deep learning for smart fish farming: applications, opportunities and challenges 13(1):66–90. https://doi.org/10.1111/raq.12464

Zavalas R, Ierodiaconou Daniel D, Ryan D et al (2014) Habitat classification of temperate marine macroalgal communities using bathymetric LiDAR. Remote Sens 6(3):2154–2175. https://doi.org/10.3390/rs6032154

Zhang D, Kopanas G, Desai C et al (2016) Unsupervised underwater fish detection fusing flow and objectiveness. 2016 IEEE Winter Appl Comput Vis Workshops (WACVW) 1–7. https://doi.org/10.1109/WACVW.2016.7470121

Zhao S, Zhang S, Lu J et al (2022) A lightweight dead fish detection method based on deformable convolution and YOLOV4. Comput Electron Agric 198:107098. https://doi.org/10.1016/j.compag.2022.107098

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.