Computer vision and target detection: emerging trends and technologies – Original Research Article

SCIENCE PROGRESS

Improved YOLOv5 s and transfer learning for floater detection

Science Progress
2025, Vol. 108(2) 1–26

DOI: 10.1177/00368504251342075
journals.sagepub.com/home/sci

Lei Guo[1,2,3], Yiqing Zhang[3], Qingqing Tian[3] and Yunlong Ran[1,2]

Abstract

This study aims to address the detection and classification of floating objects on water surfaces, including items such as bottles, plastic bags, aquatic plants, and dead fish, which pose significant threats to water quality and ecosystems. Traditional detection methods rely on manual observation and cleanup, which are inefficient, costly, and risky. To tackle this challenge, this paper proposes a solution based on an improved YOLOv5 s model by collecting floating object image data and constructing and processing the dataset using manual photography and SAGAN data augmentation techniques. We optimized the YOLOv5 s model by integrating the EfficientNetv2 lightweight network, the content-aware reassembly of features lightweight upsampling module, the bidirectional feature pyramid network structure, and by introducing attention modules such as squeeze-and-excitation and efficient multi-scale attention, along with the scylla intersection over union (SIoU) loss function. Additionally, transfer learning techniques were employed to enhance the model's performance in detecting floating objects on water surfaces, and ablation experiments were conducted to validate the effectiveness of each improvement. The results show that the improved YOLOv5 s model exhibits better performance and generalization ability on the test set, with a 5.27 percentage point increase in model accuracy. The model's parameter count, computational load, and weight size are 53.9%, 21.3%, and 54% of the original YOLOv5 s model, respectively, providing an efficient, accurate, and real-time solution for detecting floating objects on water surfaces. The methodology presented in this paper holds significant importance for the monitoring of aquatic ecological

[1]Henan Water Conservancy Investment Group CO., LTD, Zhengzhou, China
[2]Henan Water Valley Innovation and Technology Research Institute Co. Ltd., China
[3]North China University of Water Resources and Electric Power, Zhengzhou, China

Corresponding author:
Qingqing Tian, North China University of Water Resources and Electric Power, Zhengzhou 450046, China.
Email: tq10078@126.com

environments and the management of floating debris, offering valuable insights for achieving precise and efficient detection and classification of floating objects on water surfaces.

Keywords
YOLOv5s, data enhancement, transfer learning, lightweight networks, floaters

Introduction

With the advancement of industry and agriculture and the continuous development of society, an increasing amount of waste enters water bodies through various pathways, leading to a growing number of floating objects in the water. These floating objects are not limited to common items such as bottles, plastic bags, water plants, and dead fish. The decomposition of organic matter in floating debris can severely endanger water quality safety, while plastic waste can cause entanglement or ingestion by aquatic organisms, leading to death or reproductive issues in these organisms and subsequently affecting the entire food chain.[1–4] Moreover, plastic waste can break down into microplastics, which can adsorb toxic substances, polluting the entire water body and further threatening drinking water safety.[5–7] If these floating debris are not detected and dealt with in a timely manner, may they settle to the bottom of the water body, occupying the habitats and breeding grounds of aquatic organisms, leading to an imbalance in the ecosystem.

For water bodies such as fish ponds, lakes, and rivers that are under human supervision, the survival of fish and other aquatic organisms is an important indicator in the water management system. After fish die, if their carcasses are not promptly removed, they will undergo ammonification under the action of microorganisms and various enzymes. The pathogens carried by dead fish can spread throughout the water body along with the fats, posing a serious threat to the related organisms in the water, water quality, and drinking water safety in the surrounding areas. In addition, under certain conditions, the abnormal proliferation of water plants or algae in the water can not only lead to the death of aquatic organisms due to oxygen depletion but also reflect the abnormality of water quality to some extent, which helps managers to formulate timely countermeasures.

At present, research on the detection and identification of floating objects on the water surface is mainly divided into two directions. One is based on traditional object detection algorithms, which usually require manual design of feature extractors and classifiers and do not rely on large amounts of data, making it relatively easy to implement.[8,9] However, this method is more sensitive to environmental disturbances such as lighting, occlusion, and noise, and has poor robustness. The other is based on deep learning object detection algorithms, which is one of the most advanced and effective object detection methods currently available, with the main advantages of high precision, high efficiency, and strong robustness. The protection and management of water resources is one of the significant challenges facing society today, and the threat posed by floating objects to water quality and ecosystems is becoming increasingly serious.[10–12] Therefore, to effectively

address this issue, it is necessary to conduct efficient and accurate detection and identification of floating objects.

The traditional method of floater detection often relies on human observation and cleaning, which has issues such as low efficiency, high cost, and high risk. Therefore, the introduction of artificial intelligence and deep learning models becomes a potential solution.[13–16] Traditional machine learning-based object detection algorithms include three main steps: Region selection, feature extraction, and classifier classification. Feature extractors like HOG,[17] LBP,[18] and SIFT[19] are used to extract target features, while support vector machines (SVMs)[20] and Adaboost[21] algorithms are used for classification. Finally, the non-maximum suppression[22] algorithm is used to remove redundant frames and output the results. For example, Zhang et al.[23] used interpolation fitting to reconstruct the data and extract features to overcome the noise generated by the battery during data acquisition. Subsequently, the Swin Transformer network processed the learned features to achieve accurate state-of-charge prediction for lithium-ion batteries. Xu and Jin[24] proposed a method for detecting small sea surface targets based on multi-dimensional features and SVM. Features were extracted from radar echoes, and a tunable SVM classifier was designed to detect small targets against complex sea backgrounds. However, this method relies on radar equipment and is not suitable for freshwater areas. Yang[25] used the Mean Shift algorithm to segment images, estimate pollution information, and extract color moment features (in RGB and HIS spaces) and texture features (using wavelet transform) of floating objects. An SVM classifier was then employed for classification. Yu et al[26] converted images from the RGB to the HSV space to enhance the contrast between reflections and the water surface. Target contours were extracted through morphological calculations, and target detection was performed based on the aspect ratio of the contours.

The superiority and convenience of deep learning methods have led to their increasingly widespread application in more fields.[27,28] Deep learning-based object detection algorithms are among the most advanced and effective methods, offering high precision, efficiency, and robustness. They have been widely used in fields such as face recognition,[29] vehicle detection[30] and automatic driving.[31] YOLO series algorithms[32–34] have better real-time performance compared to two-stage detection algorithms like R-CNN[35] and Faster R-CNN.[36] Among them, YOLOv5 s[37–39] has gained attention for its fast detection speed and high accuracy. For example, Wang et al.[40] proposed a lightweight MSCCR module integrated into the YOLOv5 s backbone network, reducing model parameters and improving accuracy. Liu and Liu[41] proposed an improved method for real-time apple detection based on YOLOv5 s, incorporating coordinate attention blocks and a bidirectional feature pyramid network to enhance small target detection.

However, YOLOv5 s needs to be improved to enhance its performance and applicability in detecting floating objects in water. At the same time, transfer learning, as an important machine learning technology, has unique advantages in addressing issues like sample scarcity and domain transfer. Through transfer learning, existing data and knowledge can be transferred from the source domain to the target domain, improving the model's generalization ability and performance on the target task. In the task of floating objects detection, transfer learning can effectively utilize existing data and models to quickly locate

and identify floating objects in waters, thus improving detection accuracy and efficiency. In summary, this study aims to combine the improved YOLOv5 s model and transfer learning technology to propose a new method for detecting floating objects in waters. By optimizing the network structure and training strategy of the YOLOv5 s algorithm and applying the concept of transfer learning, the goal is to improve the accuracy, stability, and adaptability of floating objects detection, providing more efficient and reliable technical support for water environment monitoring and protection. Through the implementation of this study, it is expected to promote the development of intelligent water management and floating matter monitoring technology, and provide stronger guarantees for environmental protection and water resources management.

Test data

Image data acquisition

The main data involved in this paper is about floating objects on the surface of the water. It mainly detects and classifies floating objects on the surface of the water. According to floating objects that frequently appear on the surface of the water, the floating objects are divided into four categories, including bottles, plastic bags, aquatic plants, and dead fish. These categories basically include the types of garbage that often appear on the water surface and can pose a threat to the ecological environment of the water. The detection of floating objects on the water surface is different from conventional target detection, the water surface environment is relatively complex, the reflection of the water surface, reflection, ripples, etc., will interfere with the detection, and the volume of these floating objects is relatively small, in the practical application detection, the floating objects are in the surveillance camera screen. Secondly, another research object of this paper is the pollutants on the water surface of rivers, such as naturally shed branches, eutrophic algae and other pollutants.

According to the characteristics of floating objects on the surface, the collection of relevant data sets is carried out. Since there is no official data set about the surface target, this paper collects the data set by manual shooting means, selects the environment such as river, lake and pond, and shoots the above four types of floating objects from different angles and in different weather. The collected data is filtered to remove images with high similarity and poor sharpness. The filtered image is shown in Figure 1. In this paper, 3000 images of floating objects on the water surface are uniformly cut to 640×640 size.

Image data enhancement

From 3000 images of floating objects obtained, the clearer images were selected as the original data set, and the training set and test set were divided by 9:1 ratio. Image data enhancement is performed using SAGAN,[42] in which a non-local model is adopted, allowing the generator and discriminator to effectively construct the relationship between various regions and directly calculate the relationship between two pixels. The self-attention mechanism simply calculates the response of a single location in the feature-

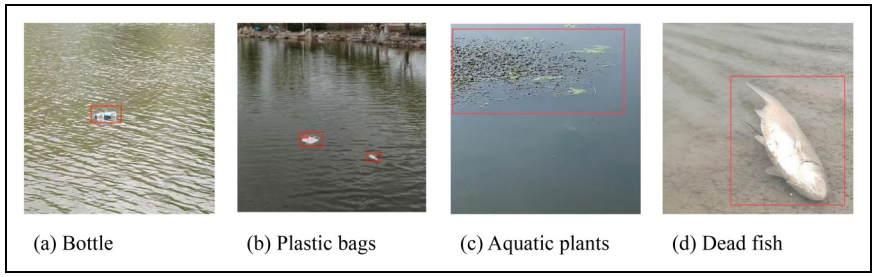

Figure 1. Image of floating objects on the surface.

weighted sum of all locations. This mechanism allows the network to focus on regions that are scattered in different locations but are structurally related.[43]

The principle is that in the generation network, the prior input noise and condition information are combined to form a multi-mode vector, which is sent into the generation network as input representation information. The category label is introduced into the discrimination network to judge not only the truth or falsity of the picture, but also whether the generated category matches the category of the input picture. Finally, a total of 28,366 images were generated, the original image size was uniformly adjusted to 640×640, and then the graphics were normalized

Method

YOLOv5 s object detection model

YOLOv5 is based on the previous YOLO series of algorithms to improve the detection performance. In particular, in the PASCALVOC and COCO target detection tasks, YOLOv5 s demonstrated excellent results, striking a good balance between detection accuracy and speed. Compared with YOLOv4, YOLOv5 model adopts CSP structure not only in Backbone, but also in Neck to retain richer feature information, thus enhancing the feature fusion capability of the network. The YOLOv5 series includes four network models of different sizes: YOLOv5 s, YOLOv5 m, YOLOv5 l and YOLOv5x. Among them, YOLOv5 s is the network with the shallowest depth and the smallest width of feature map, and other versions are further deepened and widened on the basis of it.

YOLOv5 s consists of four parts: Input, backbone, neck and output. The input terminal consists of Mosaic data enhancement,[44] adaptive picture scaling and adaptive anchor frame calculation. The backbone network is responsible for extracting target features, which is mainly composed of slice structure,[45] cross-stage local network unit[46] and spatial pyramid pool.[47] The Focus slice structure slices the input horizontally and horizontally and then splices it. The W and H dimension information is gathered in the channel space, which improves the receptive field and reduces the computation amount. The CSP1_X unit is a Bottleneck module consisting of several classical residual structure modules. After convolution, the input is merged with the original value to enrich the

gradient features, and the feature transfer is completed while the output depth is guaranteed. The SPP module uses the maximum pooling method to assemble three different scale feature maps for the input to achieve multi-scale fusion. The neck network is mainly responsible for feature fusion. Through the top-down and bottom-up feature transfer methods, the features at all levels are effectively fused,[48] then passed into the detection layer, and the final detection result is obtained through post-processing operations such as non-maximum suppression.

Compared to the currently widely adopted YOLOv8,[49] YOLOv5 s is simpler and more structured, striking a balance between model reasoning speed and detection accuracy. Especially for the need to quickly obtain the characteristics of floating objects, YOLOv5 s is a good choice. However, the original YOLOv5 s algorithm has the problems of low precision and slow reasoning speed on the floating object data set. Therefore, this study began to improve the YOLOv5 s model to achieve rapid and accurate identification of floating objects.

Model improvement

EfficientNetv2 lightweight network. EfficientNetv2, proposed by Shi et al.,[50] is a lightweight CNN network improved upon EfficientNet. Compared to EfficientNet, EfficientNetv2 achieves faster training and inference speeds while maintaining comparable accuracy and parameter counts. Although EfficientNet extensively employs depthwise convolutions—which reduce parameters and computations compared to standard convolutions—the use of DW convolutions in shallow network layers leads to slower processing and underutilization of GPU capabilities. To address this issue, EfficientNetv2 replaces the shallow MBConv structures in EfficientNet with Fused-MBConv structures. Specifically, the 1×1 expansion convolution and 3×3 DW convolution in the main branch are replaced with a single 3×3 standard convolution. The squeeze-and-excitation module[51] is embedded to enhance channel-wise feature extraction.

EfficientNetv2 utilizes neural architecture search technology to explore the optimal combination of MBConv and Fused-MBConv modules, balancing speed and performance to derive the EfficientNetv2-B0 architecture as outlined in Table 1. In the table, the numbers following MBConv and Fused-MBConv indicate the feature channel expansion ratios. Notably, during the actual implementation of EfficientNetv2-B0, the squeeze-and-excitation (SE) module is omitted in the shallow Fused-MBConv structures.

In the specific implementation, the input image is first subjected to preliminary feature extraction through a stem layer composed of a 3×3 convolution with a stride of 2. Subsequently, Fused-MBConv and MBConv modules are sequentially stacked. For shallow layers, the Fused-MBConv structure is adopted (replacing the original depthwise convolution with a standard 3×3 convolution), while the deep layers retain MBConv and embed squeeze-and-excitation modules to enhance channel-wise feature responses. To align with the feature map dimensions of YOLOv5 s, the number of output channels at each stage is adjusted to ensure compatibility with the neck network. During training, a dynamic learning rate decay strategy is employed, with an initial learning rate set to

Table 1. EfficientNetv2-B0 structure.

Network stage	Operator	Layers	Channels	Convolution kernel stride
0	Stem3 × 3	1	32	2
1	Fused-MBConv1, k3 × 3	1	16	1
2	Fused-MBConv4, k3 × 3	2	32	2
3	Fused-MBConv4, k3 × 3	2	48	2
4	MBConv4, k3 × 3, SE0.25	3	96	2
5	MBConv6, k3 × 3, SE0.25	5	112	1
6	MBConv6, k3 × 3, SE0.25	8	192	2
7	Conv1 × 1&Pooling&FC	1	1280	

Layers denotes the number of times the MBConv or Fused-MBConv structure is repeated in that stage; Stem contains a 3 × 3 normal convolution with a step size of 2 and BN layer and a Swish activation function; SE stands for using the SE module and 0.25 is the SE_ratio.

0.01. This is combined with the cosine annealing algorithm to optimize gradient update directions, alongside the introduction of label smoothing techniques to mitigate class imbalance issues.

Content-aware reassembly of features (CARAFE) lightweight upsampling module. The upsampling operation can be interpreted as feature recombination through the dot product between the upsampling kernel at each position and the corresponding neighborhood pixels in the input feature map. The most commonly used upsampling methods, such as nearest-neighbor and bilinear interpolation, determine the upsampling kernel solely based on the local neighborhood of pixels. These methods fail to leverage the global semantic information of the feature map and exhibit limited receptive fields, hindering effective extraction of global image features. To better utilize the global features around pixels during upsampling, this study replaces the original upsampling module in the YOLOv5 s model with the lightweight CARAFE upsampling module,[52] as illustrated in Figure 2.

In the implementation, the input feature map is first reduced to C_m dimensions via a 1 × 1 convolution. A dynamic upsampling kernel with a size of K_{up} =5 is then predicted through a convolutional layer with k_e =3. The predicted kernel is normalized using Softmax and applied to the nearest-neighbor upsampled feature map through position-wise dot products, enabling content-aware feature recombination. During training, the He initialization method is employed to optimize convolutional layer parameters, and a feature map reconstruction constraint term is added to the loss function to ensure kernel stability. Experimental results demonstrate that the CARAFE module effectively enhances the recovery capability of small target features while introducing only 1.3×10^5 additional parameters.

CARAFE lightweight up-sampling module consists of up-sampling kernel prediction module and feature recombination module. The feature map is first passed into the up-sampled kernel prediction module, and the channel number of the feature layer C is reduced to C_m after passing through a CBS convolutional layer with convolution kernel

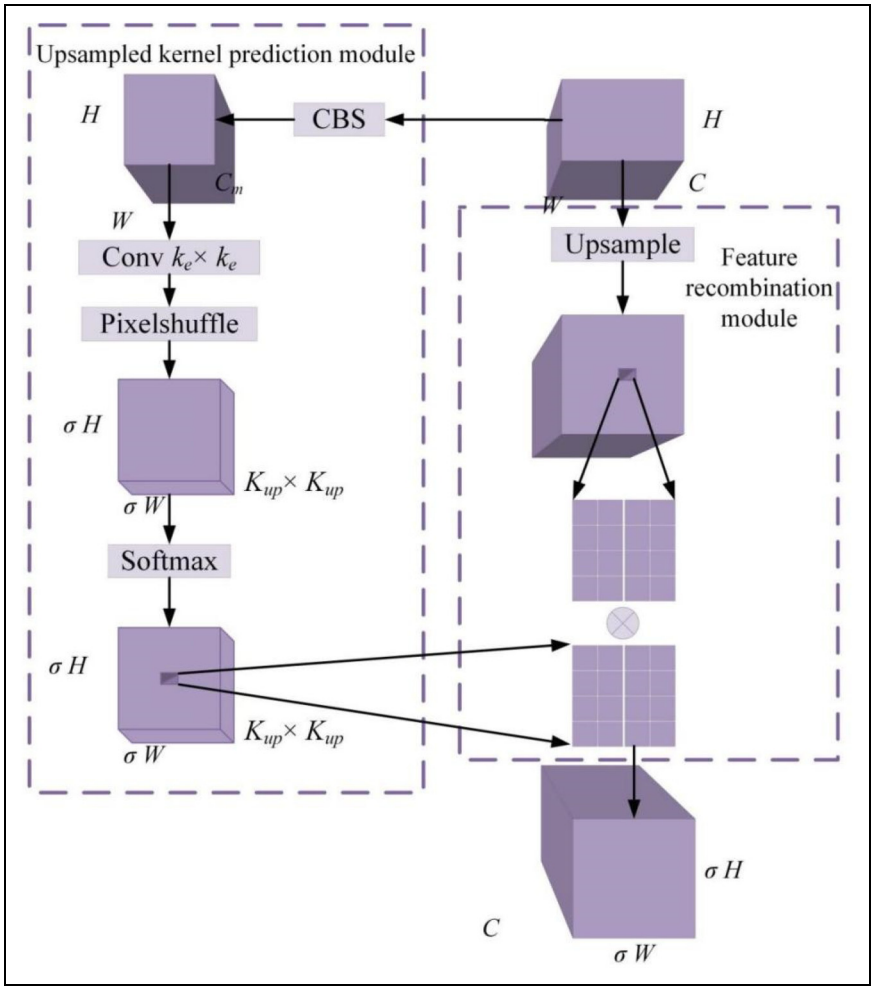

Figure 2. CARAFE lightweight upsampling module. Note: CBS denotes the convolutional layer consisting of convolution, BN, and SiLU activation functions; Kup is prediction of the upsampling kernel size; σW is the width of the feature map after upsampling; and σH is the height of the feature map after upsampling; Cm is the number of feature channels after dimensionality reduction; ke is the convolution kernel size of the convolutional layer.

of 1×1, as shown in the formula:

$$C_m = \sigma^2 \cdot (K_{up})^2 \tag{1}$$

where σ is the up-sampling multiple and K_{up} is the up-sampling kernel size.

For the input feature graph after channel compression, a convolution layer with the convolution kernel size of $k_e \times k_e$ is used to predict the upper convolution kernel, and the number of output channels is $\sigma^2 K_{up}^2$. Then PixelShuffle method is used to expand

the channel dimensions in spatial dimension to obtain the upper sampling kernel with the shape of $\sigma H \cdot \sigma W \cdot K_{up}^2$. Use the Softmax activation function to normalize processing. For the feature map of the input feature recombination module, the up-sampling result is obtained by the dot product of the nearest up-sampling and the predicted up-sampling. Because CARAFE lightweight up-sampling module has a large receptor field during recombination, it can generate corresponding up-sampling kernel according to input features, which has content perception ability and improves the weight of the target of concern. Compared with the nearest neighbor up-sampling method, it improves the feature extraction ability of up-sampling operation under the premise of introducing few parameters and calculation amount.

PANet was replaced with bidirectional feature pyramid network (BiFPN) structure. In the water surface floating object dataset, some floating objects are inherently small or captured from long distances, resulting in limited target areas. Although YOLOv5 s employs the PANet structure for feature extraction, it still suffers from missed detections and false positives. To improve detection accuracy for small targets, the original PANet structure is replaced with BiFPN.[53] In PANet, only a single top-down path and a single bottom-up path are utilized. In contrast, BiFPN treats each bidirectional path as a feature network layer and repeatedly stacks them to achieve higher-level feature fusion. Additionally, BiFPN introduces an extra path between the original input and output nodes, enabling the fusion of more features without significantly increasing computational costs. PANet's feature fusion relies on simple feature map addition without distinguishing contributions from input feature maps. However, input feature maps with varying resolutions contribute unequally to fusion, making naive addition suboptimal. To address this, BiFPN incorporates a weighted feature fusion mechanism, as expressed by the equation:

$$O = \sum_i \frac{w_i}{\varepsilon + \sum_j w_i} - I_i \tag{2}$$

In the formula, the value of ϵ is 0.0001, its function is to ensure that the denominator is not 0 to ensure the stability of the value, W_i is the weight range between,[0,1] I_i is the input feature map.

In the implementation, BiFPN constructs a multi-level feature pyramid through bidirectional cross-node connections and introduces learnable weight coefficients w_i during feature fusion. During training, group normalization and L2 regularization strategies are applied to prevent weight overfitting. Additionally, skip connections are incorporated into the cross-node paths to preserve original feature information. This method aligns with the global multiscale feature fusion module (GMFFM) proposed by Liu et al.,[54] which leverages multiscale receptive fields and attention mechanisms to capture both local and global contextual information.

Introduction attention module. To enhance feature selectivity, SE, efficient channel attention (ECA), and efficient multi-scale attention (EMA) attention modules are embedded at critical positions in the backbone network. Specifically: The SE module generates

channel-wise weights through global average pooling (GAP) and fully connected layers, embedded at the end of MBConv modules. The ECA module replaces the fully connected layers in SE with 1D convolutions, reducing parameter count while capturing cross-channel interactions. The EMA module employs a multi-branch architecture for multi-scale feature aggregation. Its 1×1 branch encodes spatial information via 2D global pooling, while the 3×3 branch extracts local detail features. The two attention maps are fused through matrix multiplication. During training, the group number (G) for the EMA module is set to 8, and the initial learning rate is reduced to 0.001 to prevent gradient explosion.

SE. SENet[55] is a channel attention mechanism whose main operations are squeeze and excitation. Before the input image passes through the attention mechanism module, each channel of the feature map has the same importance, and after passing SENet, the importance of each feature channel is different. For the neural network, the channel with a large weight value will be focused. The implementation process of SE attention mechanism in neural network is as follows: (1) Squeeze: Global pooling is adopted, that is, H and W are compressed to 1×1, and a weight value is used to represent a channel to achieve low-dimensional embedding, with input $H \times W \times C$ and output $1 \times 1 \times C$. The compressed feature is essentially a vector, with no spatial dimension, only channel dimension. (2) Excitation: Generate a weight value for each feature channel. The correlation between the channels is constructed through two fully connected layers. The number of output weights is the same as the number of channels in the input feature map. The input is $1 \times 1 \times C$ and the output is $1 \times 1 \times C$. (3) Scale: Multiply the normalized weights with corresponding channels and apply them to the features of each channel. The input is $H \times W \times C$ and $1 \times 1 \times C$, and the output is $H \times W \times C$.

ECA. ECA[56] attention mechanism and SE attention mechanism are both channel attention mechanisms. By weighting different feature maps, the model pays more attention to features that contribute to target detection. However, since dimensionality reduction operations in SENet have side effects on the channel attention mechanism, capturing the dependencies between all channels is inefficient and unnecessary. ECA improves on the SE attention mechanism by using a 1×1 convolution layer directly after the GAP layer, eliminating the fully connected layer, avoiding dimensional reduction, and effectively capturing cross-channel interactions, with only a few parameters involved to achieve good results. ECA module first performs GAP on the last convolution output, adopts one-dimensional convolution to rapidly capture the cross-channel information interaction between each channel and its nearly K adjacent channels, and obtains the learnable weight coefficient of each channel through Sigmoid activation function. Then the weights are applied to each channel of each original feature map to generate the weighted feature map.

EMA. In this paper, EMA[57] mechanisms were introduced in the Backbone of YOLOv5 s to improve the weight of important features and reduce the weight of irrelevant features such as matrix and cavity disk edge, thereby improving the accuracy of model detection. The structure of EMA's attention mechanism is shown in Figure 3.

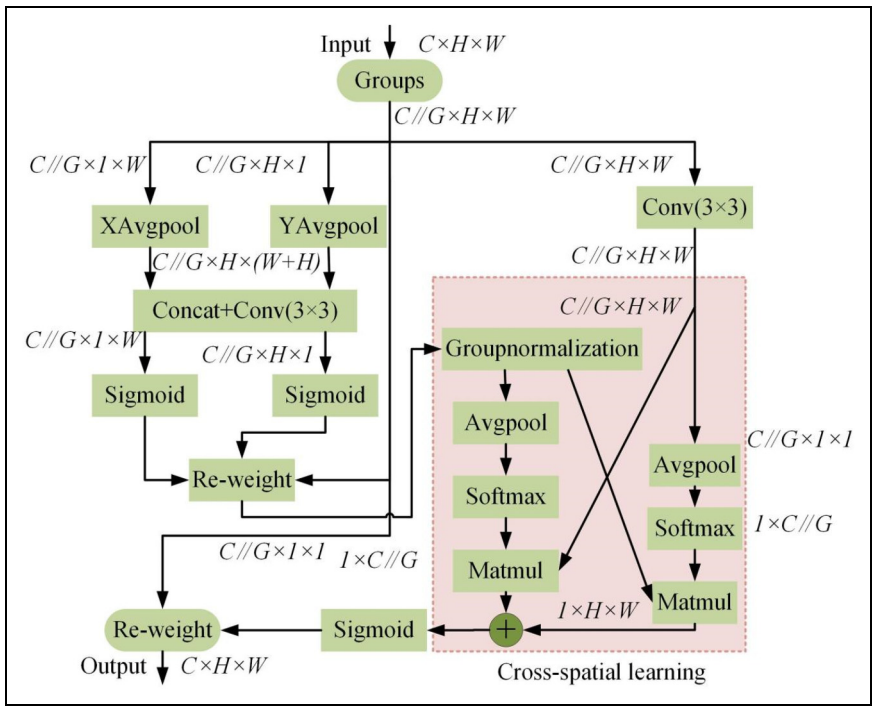

Figure 3. Efficient multi-scale attention (EMA) attention mechanism. Note: Re-weight is the adaptive feature variable selection module; Groupnormalization denotes group normalization; Matmul denotes matrixmultiplication.

The EMA attention mechanism consists of a 1×1 branch, a 3×3 branch and a cross-spatial learning module. For any given input feature map, we assume:

$$X \in R^{C \times H \times W} \tag{3}$$

where, C, H and W are the number of channels, height and width of the input feature map respectively.

EMA divides X into G sub-features according to the cross-channel dimension direction for learning different semantics, where the group style can be defined as:

$$X = [X_0, X_i, \cdots, X_{G\text{-}1}], X_i \in R^{C//G \times H \times W} \tag{4}$$

In the formula, $G \ll C$.

To capture the dependencies between all channels and reduce the computational overhead, EMA uses two one-dimensional global averaging pooling operations to encode channels in both spatial directions in the 1×1 branch, stacking only one 3×3 kernel in the 3×3 branch to capture multi-scale feature representations. In EMA, a cross-spatial information aggregation method with different spatial dimensions can achieve richer

feature aggregation. First of all, two tensors where one is the output of the 1×1 branch and the other is the output of the 3×3 branch. Then, the output of the 1×1 branch is encoded with global spatial information using two-dimensional GAP, and the output of the smallest branch is directly transformed into the corresponding dimensional shape before the joint activation mechanism of channel features, namely:

$$R_1^{1 \times C//G} \times R_3^{C//G \times HW} \tag{5}$$

The two-dimensional global pooling operation formula is:

$$Z_c = \frac{1}{H \times W} \sum_j^H \sum_i^W X_c(i, j) \tag{6}$$

where X_c represents the input feature of the C channel

In order to improve the computational efficiency of the model, Softmax is used to fit the above linear transformation at the output of two-dimensional global averaging pooling. The first spatial attention diagram is obtained by multiplying the output of the above parallel processing with the matrix dot product operation. Similarly, two-dimensional global averaging pooling is used to encode global spatial information for 3×3 branches, and 1×1 branches are directly transformed into corresponding dimensional shapes before the joint activation mechanism of channel features, namely:

$$R_3^{1 \times C//G} \times R_1^{C//G \times HW} \tag{7}$$

On this basis, we derive the second spatial attention diagram which retains the exact spatial position information. Finally, the output feature plots within each group are computed as a set of two generated spatial attention weight values, and then the Sigmoid function is used to capture pixel-level pair relationships and highlight the global context of all pixels.

Replace complete intersection over union (CIoU) with scylla intersection over union (SIoU) loss function. The CIoU loss function adopted in YOLOv5 s incorporates the aspect ratio of bounding boxes as a scale-related penalty term but neglects the directional mismatch between predicted and ground-truth boxes, leading to slower convergence and reduced efficiency. To address this limitation, this study employs the SIoU loss function proposed by GEVORGYAN.[58] The SIoU loss consists of four cost functions: Angular, distance, shape, and intersection over union. The angular cost function first predicts alignment along the X or Y axis and then prioritizes minimization of the angle α (if $\alpha \leq \pi/4$) or β (otherwise) during convergence. A progressive weighting strategy is applied during training, emphasizing angular and distance optimization in early stages and strengthening shape constraints in later phases. Experimental results demonstrate that SIoU improves model convergence speed by 18% and increases average precision by 0.49 percentage points. A schematic diagram of the angular cost calculation is illustrated in Figure 4. The improved YOLOv5 s model structure is shown in Figure 5.

The angular cost function Λ is as follows:

$$singledollar\Lambda = 1\text{-}2 \times \sin^2\left(\arcsin x - \frac{\pi}{4}\right) \tag{8}$$

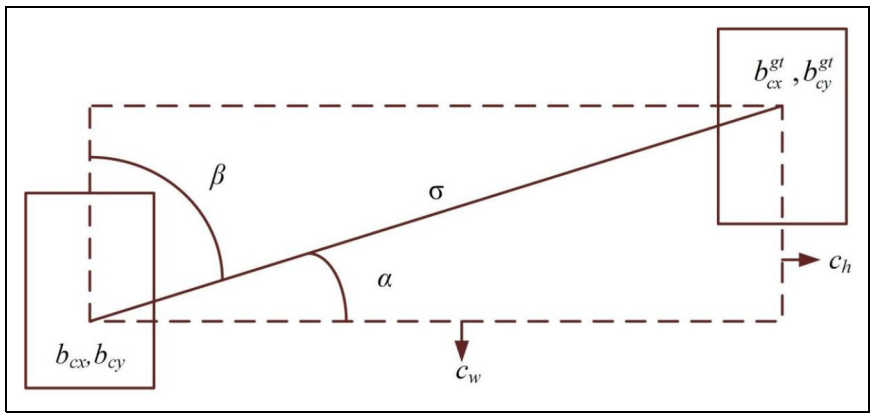

Figure 4. Schematic diagram of angular costing. Note: (b_{cx},b_{cy}) is the center point of the prediction frame; (b_{cx}^{gt}, b_{cy}^{gt}) is the centerpoint of the real frame; α is the angle between σ and *C*w; β is the angle between σand *C*h.

$$x = \frac{c_h}{\sigma} = \sin\sigma \tag{9}$$

The distance cost function Δ is shown in the equation:

$$\Delta = \sum_{t=x,y} 1\text{-}e^{-\gamma\rho_t} \tag{10}$$

$$\rho_x = \left(\frac{b_{cx}^{gt} - b_{cx}}{c_w}\right)^2, \rho_y = \left(\frac{b_{cy}^{gt} - b_{cy}}{c_h}\right)^2, \gamma = 2 - \Lambda \tag{11}$$

where, (b_{cx}, b_{cy}) is the central point of the prediction box, (b_{cx}^{gt}, b_{cy}^{gt}) is the central point of the real box, α is the angle between σ and C_w, the difference between the horizontal and vertical coordinates of the central point of the C_w and C_h prediction box and the central point of the real box, σ is the distance between the central point of the real box and the prediction box, w and h are the width and height of the prediction box.

As can be seen from the above formula, when α approaches 0, distance cost contribution decreases. As α approaches π/4, the distance cost function Δ increases. As the Angle increases, gamma is given time priority.

The shape cost function Ω is as follows:

$$\Omega = \sum_{t=w,h} (1\text{-}e^{-\omega_t})^{\theta} \tag{12}$$

$$\omega_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})}, \omega_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \tag{13}$$

where w^{gt} and h^{gt} are the width and height of the true box

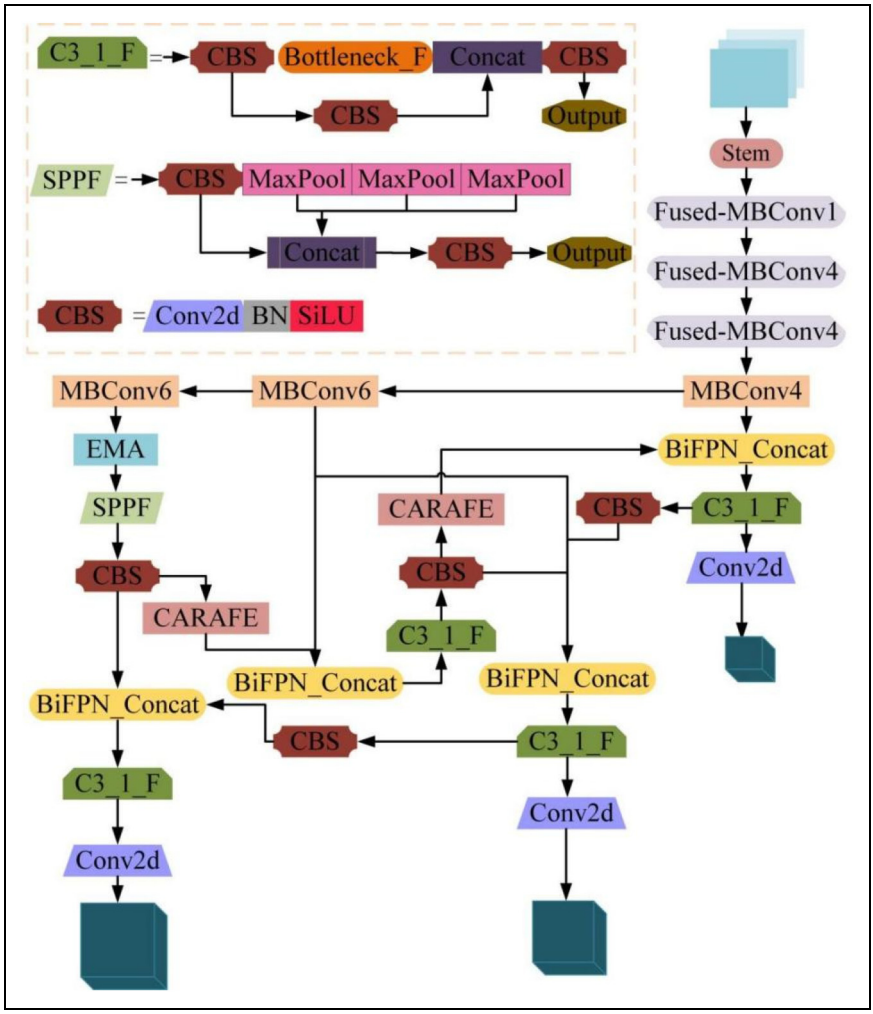

Figure 5. Improve the network structure of YOLOv5 s algorithm. Note: EMA denotes the EMA attention mechanism module; SPPF is the spatial pyramid pooling module; CARAFE is a lightweight upsampling module; BiFPN_concatis a feature fusion module using the BiFPN structure; C3_1_F is a C3 module using a shortcut-free branching bottleneck structure.

The θ value defines the shape cost, with the θ value approaching 4 in the test, and the SIoU loss function defines the θ value range.[2,6] The frame loss function L_{box} is:

$$L_{box} = 1 - I_{IoU} + \frac{\Delta + \Omega}{2} \tag{14}$$

The total loss function L consists of classification loss L_{cls} and frame loss L_{box}, as

follows:

$$L = W_{box}L_{box} + W_{cls}L_{cls} \tag{15}$$

where L_{cls} is the focus loss, W_{box} and W_{cls} are the frame and classification loss weights, respectively.

Transfer learning

The object detection task in deep learning requires a large number of samples with completed data annotation to train the model, improve the model fitting effect, and accelerate the model convergence speed. However, due to the small number of samples in the floating objects data set, overfitting is easy to occur during training. To solve this problem, transfer learning method is adopted to solve the problem of easy overfitting under the condition of small samples.[59–61]

By fully training the network on a large data set, transfer learning[62] enables the network to learn a large number of features required for image classification and recognition, and then applies the learned features to new learning tasks, so that the network can achieve better recognition and classification after simple training. The convolutional layer weights and parameters trained by YOLOv5 s network on ImageNet were transferred to the improved YOLOv5 s floating object detection model as the initial weight parameters of the model, so as to solve the problem of small sample data and improve the generalization ability of the model and the network training speed. The collected data set of pollutants along river banks is selected as an auxiliary domain, and the weight transfer of the model is realized through pre-training. The data in the auxiliary domain is used to help the model better learn the characteristics of floating objects, improve the performance of the model in the task of recognizing floating objects, and improve the recognition effect of floating objects. The pollutants along river banks are shown in Figure 6.

It is reasonable to analyze and select the river bank pollutant data set as an auxiliary domain to assist the detection of floating objects on the water surface. Considering the co-existence of river bank pollutants and surface floaters in the natural environment, they may be similar in shape, color, texture, etc., which makes the features learned in the pre-training data set of river bank pollutants transferable and help improve the performance of the surface floaters detection task. Pollutant data sets along river banks are usually easier to obtain and label, so they can provide rich training samples, which is crucial for deep learning models. More training data can help models learn more comprehensive and robust features, thus improving the generalization ability and stability of models. Pre-training can also help the model learn general visual features and knowledge on the river bank pollutant data set, such as edge detection, shape recognition, etc. These features and knowledge may still be valid in the subsequent floaters detection task, thus enhancing the model's migration ability and performance. In addition, river bank pollutant data sets usually contain a variety of different environmental conditions and scenarios, and such diverse data sets help the model better adapt to the various complex

Figure 6. River bank pollutant map.

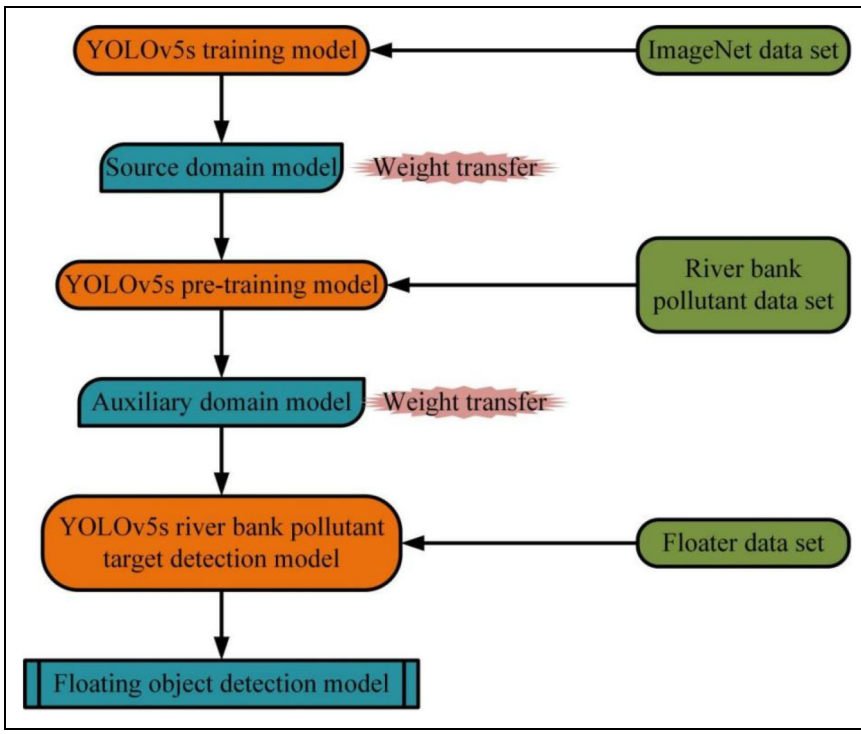

Figure 7. Floating object detection transfer learning diagram.

environments that may occur in the actual surface floaters detection task. Figure 7 shows the transfer learning diagram of floater detection.

Evaluation index

The deepening and widening of the network can usually improve the performance of the model, but it also increases the calculation amount and volume of the model,[63] which is not conducive to combining the equipment such as monitoring around the water area, surface garbage fishing

vessels, and drones after the deployment of the model. Therefore, in order to verify the performance of the recognition method proposed in this paper, P (Precision) and F1 values are used to evaluate the recognition performance of the network,[64] and the frames per second (FPS) representation model is used to detect the performance in real time.[65] Calculation formula:

$$P = TPTP + FP \tag{16}$$

$$R = TPTP + FN \tag{17}$$

$$F1 = 2PRP + R \tag{18}$$

$$ACC = TP + TNTP + TN + FP + FN \tag{19}$$

where TP is the positive sample predicted as the positive sample; FP: negative sample predicted to be positive sample; FN: positive sample predicted to be negative sample; TN is predicted to be the negative sample of the negative sample; P is the accuracy rate; R (recall) is the recall rate; F1 value is the harmonic average of accuracy rate and recall rate; ACC (accuracy) indicates the accuracy of the model. By comparing the Loss curve of the disease recognition model on the test set and the standard deviation value of the curve after the accuracy curve convergence, the overfitting of the model was evaluated.

Test environment configuration and parameter setting

This study utilizes an experimental platform based on an Intel Xeon Gold 5118 processor, 16GB RAM, and an NVIDIA GeForce RTX 2080 SUPER GPU (8GB VRAM), running on the Windows 10 operating system. The software environment is managed via Anaconda with Python 3.7, and incorporates the CUDA 11.0 programming platform along with the cuDNN 8.0 acceleration library to support the TensorFlow 2.4 deep learning framework. CUDA, as NVIDIA's GPU parallel computing architecture, efficiently accelerates large-scale numerical computations, while cuDNN optimizes low-level operations commonly used in deep neural networks (such as convolution and pooling). Together, they significantly enhance computational efficiency in model training and inference.

Parameter setting: In the training process, the size of the network input image of the target detection model is 640×640, the batch size is set to 16, and the number of training rounds is set to 300.The initial learning rate is set to 0.01, the momentum to 0.9, and the attenuation learning rate to 0.0005. The warm up strategy is adopted for training. The learning rate climbs linearly to the initial learning rate in the first three rounds of training and then slowly decreases. The default configurations of other hyperparameters are adopted in the hyp.scratch.yaml configuration file of YOLOv5 s.

Result analysis

Ablation experiment results

Abstraction experiments are a method of evaluating the impact of specific components on model performance by incrementally removing or adding them. Through these

Table 2. Results of ablation test with improved mechanics.

SIoU	EfficientNetv2	BiFPN	CARAFE	Efficient multi-scale attention (EMA)	Parameters /$\times 10^6$M	Computation /GFLOPs	Model size /MB	Accuracy /%
					7.01	16.00	13.72	90.53
√					7.01	16.00	13.72	91.02
√	√				3.62	3.20	7.07	92.23
√	√	√			3.65	3.40	7.09	93.86
√	√	√	√		3.78	3.40	7.38	94.77
√	√	√	√	√	3.78	3.40	7.41	95.80

experiments, authors can clearly demonstrate the specific contributions of each improvement to model performance, thereby validating their effectiveness and necessity. This not only helps readers better understand the model optimization process but also enhances the scientific rigor and persuasiveness of the research. The results are shown in Table 2.

Through six experiments optimizing the YOLOv5 s model, significant performance improvements were achieved. Replacing the loss function with SIoU increased model accuracy by 0.49 percentage points without adding parameters. Substituting the original feature extraction network with EfficientNetv2's Backbone resulted in a 1.21 percentage point increase in accuracy, halving the parameters and weights while reducing computation by 80%. Introducing BiFPN boosted accuracy by 1.63 percentage points with a slight increase in parameters. Using CARAFE upsampling enhanced accuracy by 0.91 percentage points, adding 1.3×10^5M parameters and 0.29MB to the weights. Adding the EMA attention mechanism further increased accuracy by 1.03 percentage points with a slight weight increase. Based on the results, the optimized YOLOv5 s model has 53.9% of the original parameters, 21.3% of the original computation, and 54% of the original weight size.

The effect of training mode and learning rate on the model

On the extended data set, when new learning and transfer learning were adopted, the initial learning rate was set as 0.01, 0.001, 0.0001, the training times were 300, and the attention module was EMA, the Loss curve of the model on the test set was shown in Figure 8. As shown in Figure 8, the transfer learning model tends to converge after 50 rounds of training on the expanded data set, while the newly learned model tends to converge after 150 rounds of training. The model converges faster under transfer learning, becomes more stable after convergence, and has stronger generalization ability.

When the learning rate is 0.001, the model accuracy is the highest, but the standard deviation of Loss curve and accuracy curve are also large, and the stability of the model after convergence is weaker than that when the learning rate is 0.0001. Therefore, when the learning rate is 0.0001, combined with transfer learning, the model accuracy rate increases by 7–12 percentage points, the Loss curve and the accuracy curve are smoother, the standard deviation of the model accuracy curve is reduced by 80% at most, and the model stability is better.

The effect of data augmentation on the model

For extended and unextended data sets, combined with transfer learning and new learning training methods, the learning rate is 0.0001, other hyperparameters are the same, the number of training rounds is 300, and the attention module is EMA, the accuracy curve of the model on the test set is shown in Figure 9(b).As can be seen from Figure 9, under the two different learning methods, the accuracy curve of the expanded data set is more stable, the curve gradually increases, the fluctuation amplitude gradually decreases, and no obvious overfitting phenomenon occurs. When the learning rate is 0.0001, the recognition performance of the model on the test set on different data sets. When transfer

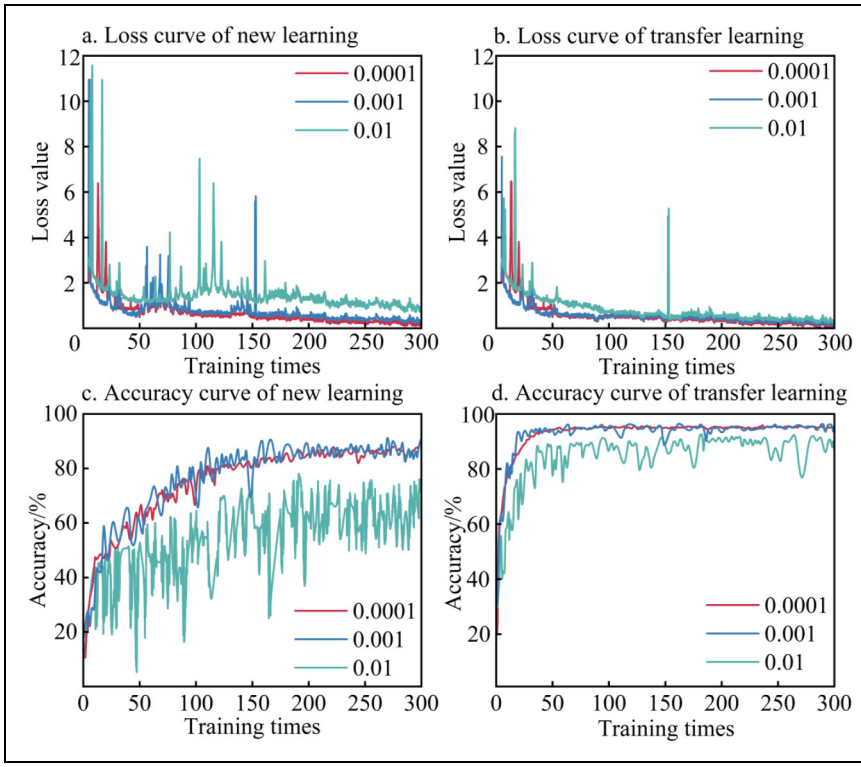

Figure 8. Loss and accuracy curves under different initial learning rates.

learning is adopted, on the expanded data set, the accuracy curve of the model has a better convergence effect, and the standard deviation after convergence is smaller, which alleviates the overfitting phenomenon of the model and enhances the generalization ability of the model. Compared with the unexpanded data set, the recognition accuracy rate of floating objects is increased by 17.16 percentage points on average, and the loss value is decreased by 0.19 on average.

Effect of attention mechanism on model performance

On the augmented dataset, using a transfer learning approach with a learning rate of 0.0001 and incorporating SE, ECA, and EMA attention modules (while keeping other hyperparameters identical), the model's loss curve and accuracy curve on the test set are shown in Figure 8. The comparative analysis among SE, ECA, and EMA attention mechanisms employs standard deviation, F1-score, and accuracy as the evaluation metrics for model performance.

As shown in Figure 10, comparing the Loss curve and accuracy curve under the three attention mechanisms, the curves all tend to converge completely after 100 rounds.

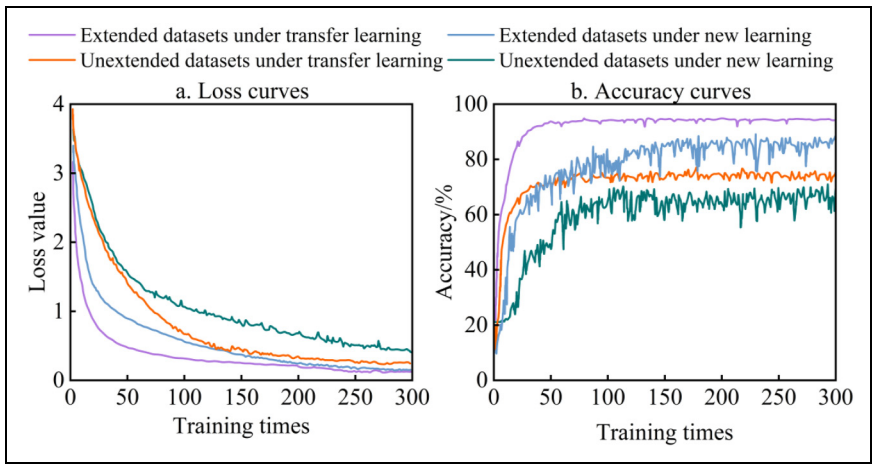

Figure 9. The impact of data augmentation and training methods on the model.

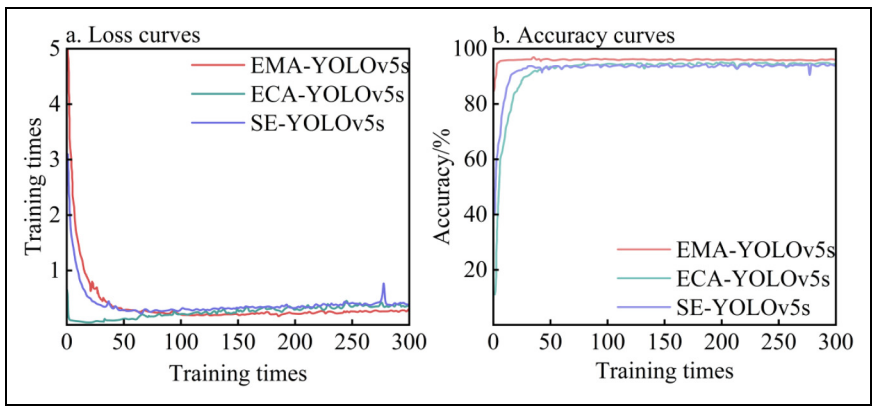

Figure 10. Loss and accuracy curves under different attention mechanisms.

During the training process, the model overfits the noise and details of the data, which will cause serious oscillation or obvious upward trend of the Loss curve after convergence. In the figure, SE-YOLOv5 s converges faster, but the Loss curve has an obvious upward trend. After the convergence of ECA-YOLOv5 s model, the Loss curve has serious oscillation, and overfitting phenomenon occurs in both models. EMA-YOLOv5 s has the best convergence effect, and the curve after convergence is smooth and smooth, with a better fitting effect and strong generalization ability. The recognition results of the models under the three attention mechanisms have little difference, but the EMA-YOLOv5 s and ECA-YOLOv5 s models have smaller size, fewer parameters and shorter running time, and are superior to SE-YOLOv5 s in recognition speed.

Table 3. Test results of different algorithm test sets.

Model	Recall/%	F1 score/%	Accuracy/%	Frames per second/(s^{-1})
Faster-RCNN	80.21	82.32	89.69	9.8
CenterNet	88.92	83.83	90.81	26.9
YOLOv3	87.31	80.41	83.15	25.7
YOLOv5s	89.63	86.28	90.53	29.6.
YOLOv8s	91.85	88.74	91.27	34.9
Improved YOLOv5s	92.41	92.69	95.80	39.1

Compared with the ECA-YOLOv5 s model, the loss curve standard deviation of the EMA-YOLOv5 s model is smaller, and the F1 value and accuracy rate are higher, which can reach 92.69% and 95.8%.

Comparison of different models

In order to further evaluate the performance level of the model, the initial learning rate was set to 0.01, the number of training rounds was set to 300 epoch, and each object detection algorithm was tested on the test set of floating objects on the water surface. The final results are shown in Table 3.

According to the results in Table 3, the average accuracy of the improved YOLOv5 s model is 5.27 percentage points higher than that of the original YOLOv5 s model in terms of average accuracy. In terms of frame rate, the improved YOLOv5 s model is significantly better than Faster-RCNN, CenterNet and YOLOv3 models, and slightly better than YOLOv8 s. Compared with other mainstream target detection models, the improved YOLOv5 s model has obvious advantages in performance. The accuracy rate and frame rate are significantly improved, and the average accuracy is the highest, which meets the accuracy and real-time requirements of the classification detection of floating objects on the water surface while realizing the model's lightweight.

Conclusion

This study proposes an enhanced YOLOv5 s model incorporating EfficientNetv2, CARAFE, BiFPN, and attention modules (SE/ECA/EMA), optimized via SIoU loss, transfer learning, and data augmentation. Key findings include:

1. The improved model outperforms the original YOLOv5 s, achieving a 4.53% accuracy gain in water surface floater detection.
2. Lightweight architecture and attention mechanisms enhance detection performance and real-time capability while maintaining computational efficiency.
3. Data augmentation combined with transfer learning improves convergence and generalization on expanded datasets.
4. The model surpasses traditional approaches in accuracy and frame rate, demonstrating practicality for aquatic environmental monitoring.

This work provides a robust framework for floater detection, contributing to water quality management and ecosystem protection.

Acknowledgments

This work was supported by the Science and Technology Innovation Leading Talent Support Program of Henan Province (Grant No. 254000510037) and the Key R&D Special Project of Henan Province (Grant No. 251111210700). The financial support is highly appreciated.

ORCID iD

Qingqing Tian https://orcid.org/0000-0002-3647-043X

Author contributions

The experiments and data curation in this study were conducted by Yiqing Zhang. Lei Guo, Qingqing Tian, and Yunlong Ran provided overall guidance and critical suggestions for the manuscript. We thank Pengbo Yin for his assistance in sample collection. All authors have read and approved the final version of the manuscript for publication.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Research on key technologies of operation and maintenance of long-distance, multi-type and complex terrain water supply projects.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Data availability statement

The investigated dataset is available from the corresponding author on a reasonable request.

Reference

1. Borrelle SB, Ringma J, Law KL, et al. Predicted growth in plastic waste exceeds efforts to mitigate plastic pollution. *Science* 2020; 369: 1515–1518.
2. Bergmann M, Tekman MB and Gutow L. Sea change for plastic pollution. *Nature* 2017; 544: 297–297.
3. Ji J, Zhao T and Li F. Remediation technology towards zero plastic pollution: recent advance and perspectives. *Environ Pollut* 2022; 313: 120166.
4. Cesarini G and Scalici M. Riparian vegetation as a trap for plastic litter. *Environ Pollut* 2022; 292: 118410.
5. Sun H, Hu J, Wu Y, et al. Leachate from municipal solid waste landfills: A neglected source of microplastics in the environment. *J Hazard Mater* 2023; 133144.
6. Lönnstedt OM and Eklöv P. RETRACTED: environmentally relevant concentrations of microplastic particles influence larval fish ecology. *Science* 2016; 352: 1213–1216.

7. Lee JH, Kang JC and Kim JH. Toxic effects of microplastic (polyethylene) on fish: accumulation, hematological parameters and antioxidant responses in Korean bullhead, pseudobagrus fulvidraco. *Sci Total Environ* 2023; 877: 162874.
8. Jin X, Niu P and Liu L. A GMM-based segmentation method for the detection of water surface floats. *IEEE Access* 2019; 7: 119018–119025.
9. Niu P. Research on intelligent monitoring technology of floating objects on water surface based on background difference [D]. *Nanjing Univ Posts Telecommun* 2020. DOI:10. 27251/dc nki. GNJDC. 2020.000602. (in Chinese).
10. Gu J, Zhang Y, Tuo P, et al. Surface floating objects moving from the pearl river estuary to hainan island: an observational and model study. *J Mar Sys* 2024; 241: 103917.
11. Di Risio M and Sammarco P. Effects of floaters on the free surface profiles of river flows. *Environ Fluid Mech* 2020; 20: 527–537.
12. Azari M, Farjad Y, Nasrolahi A, et al. Diatoms on sea turtles and floating debris in the Persian Gulf (Western Asia). *Phycologia* 2020; 59: 292–304.
13. Zhang L, Wei Y, Wang H, et al. Real-time detection of river surface floating object based on improved RefineDet. *IEEE Access* 2021; 9: 81147–81160.
14. Yi Z, Yao D, Li G, et al. Detection and localization for lake floating objects based on CA-faster R-CNN. *Multimed Tools Appl* 2022; 81: 17263–17281.
15. Li S, Liu S, Cai Z, et al. TC-YOLOv5: rapid detection of floating debris on raspberry Pi 4B. *Journal of Real-Time Image Processing* 2023; 20: 17.
16. Shi C, Lei M, You W, et al. Enhanced floating debris detection algorithm based on CDW-YOLOv8. *Phys Scr* 2024; 99: 076019.
17. Aslan MF, Durdu A, Sabanci K, et al. CNN And HOG based comparison study for complete occlusion handling in human tracking. *Measurement (Mahwah N J)* 2020; 158: 107704.
18. Kaplan K, Kaya Y, Kuncan M, et al. Brain tumor classification using modified local binary patterns (LBP) feature extraction methods. *Med Hypotheses* 2020; 139: 109696.
19. Gupta S, Thakur K and Kumar M. 2D-human Face recognition using SIFT and SURF descriptors of face's feature regions. *Vis Comput* 2021; 37: 447–456.
20. Okwuashi O and Ndehedehe CE. Deep support vector machine for hyperspectral image classification. *Pattern Recognit* 2020; 103: 107298.
21. Wang W and Sun D. The improved AdaBoost algorithms for imbalanced data classification. *Inf Sci (Ny)* 2021; 563: 358–374.
22. Rahman R, Bin Azad Z and Bakhtiar Hasan M. Densely-populated traffic detection using yolov5 and non-maximum suppression ensembling. In: *Proceedings of the international conference on big data, IoT, and machine learning: BIM 2021*. Singapore: Springer Singapore, 2022, pp.567–578.
23. Zhang W, Hao H and Zhang Y. State of charge prediction of lithium-ion batteries for electric aircraft with swin transformer. *IEEE/CAA J Automatica Sinica* 2024; 12(3): 645–647.
24. Xu Y and Jin L. Sea-surface floating small target detection based on SVM and multidimensional features. In: *2020 IEEE international conference on information technology, big data and artificial intelligence (ICIBA)*. IEEE, 2020, November, pp.453–457.
25. Yang P. *Research on intelligent monitoring of water surface based on Computer vision [D]*. Guizhou Minzu University, 2015, (in Chinese).

26. Yu S, Yang H, Kong F, et al. A visual detection method for floating target on surface [J]. *Mech Elect Eng Technol* 2019; 48: 131–133. in Chinese.
27. Zhang W, Xu M, Yang H, et al. Data-driven deep learning approach for thrust prediction of solid rocket motors. *Measurement (Mahwah N J)* 2024; 225: 114051.
28. Li X, Yu S, Lei Y, et al. Dynamic vision-based machinery fault diagnosis with cross-modality feature alignment. *IEEE/CAA J Automatica Sinica* 2024; 11: 2068–2081.
29. Wang H and Guo L. Research on face recognition based on deep learning. In: *2021 3rd international conference on artificial intelligence and advanced manufacture (AIAM)*. IEEE, 2021, October, pp.540–546.
30. Maity M, Banerjee S and Chaudhuri SS. Faster r-cnn and yolo based vehicle detection: a survey. In: *2021 5th international conference on computing methodologies and communication (ICCMC)*. India: IEEE, 2021, April, pp.1442–1447.
31. Keerthana M, Prasath MV and Yaswanthkumar SK. A computer vision approach for automated driver assistance system. In: *2021 International conference on intelligent technologies (CONIT)*. India: IEEE, 2021, June, pp.1–5.
32. Zi N, Li XM, Gade M, et al. Ocean eddy detection based on YOLO deep learning algorithm by synthetic aperture radar data. *Remote Sens Environ* 2024; 307: 114139.
33. Zhang Q, Yang Q, Zhang X, et al. A multi-label waste detection model based on transfer learning. *Resour Conserv Recycl* 2022; 181: 106235.
34. Yan J and Wang Z. YOLO V3+ VGG16-based automatic operations monitoring and analysis in a manufacturing workshop under industry 4.0. *J Manuf Syst* 2022; 63: 134–142.
35. Chen Y, Wang H, Li W, et al. Scale-aware domain adaptive faster r-cnn. *Int J Comput Vision* 2021; 129: 2223–2243.
36. Peng J, Wang D, Liao X, et al. Wild animal survey using UAS imagery and deep learning: modified faster R-CNN for kiang detection in Tibetan plateau. *ISPRS J Photogramm Remote Sens* 2020; 169: 364–376.
37. Li S, Liu C, Tang K, et al. *Improved YOLOv5 s Algorithm for Small Target Detection in UAV Aerial Photography*. USA: IEEE Access, 2024.
38. Fang Y, Wu Q, Li S, et al. Enhanced YOLOv5s-based algorithm for industrial part detection. *Sensors* 2024; 24: 1183.
39. Hou J and Zhang C. *Shallow mud detection algorithm for submarine channels based on improved YOLOv5 s*. Netherlands: Heliyon, 2024.
40. Wang Y, Xu S, Wang P, et al. Lightweight vehicle detection based on improved YOLOv5 s. *Sensors* 2024; 24: 1182.
41. Liu J and Liu Z. YOLOv5s-BC: an improved YOLOv5s-based method for real-time apple detection. *J Real-Time Image Process* 2024; 21: 1–16.
42. She C, Chen T, Duan S, et al. SAGAN: deep semantic-aware generative adversarial network for unsupervised image enhancement. *Knowl Based Syst* 2023; 281: 111053.
43. Babiloni F, Marras I, Deng J, et al. Linear complexity self-attention with $3^{\text {rd}} $ order polynomials. *IEEE Trans Pattern Anal Mach Intell* 2023; 45(11): 12726–12737.
44. Zhang L, Hu X, Zhang M, et al. Object-level change detection with a dual correlation attention-guided detector. *ISPRS J Photogramm Remote Sens* 2021; 177: 147–160.
45. Xia M, Yan Y, Li C, et al. A Unified Machine Learning Through Focus Resist 3D Structure Model. *IEEE Trans Semicond Manuf* 2023; 37(1): 59–66.

46. Ma L, Li X, Dai X, et al. A combined detection algorithm for personal protective equipment based on lightweight Yolov4 model. *Wirel Commun Mob Comput* 2022; 2022: 1–11.
47. Huang Z, Wang J, Fu X, et al. DC-SPP-YOLO: dense connection and spatial pyramid pooling based YOLO for object detection. *Inf Sci (Ny)* 2020; 522: 241–258.
48. Xu H, He Z and Chen S. Receptive field enhancement and attention feature fusion network for underwater object detection. *J Electron Imaging* 2024; 33: 033007–033007.
49. Qiu S, Cai B, Wang W, et al. Automated detection of railway defective fasteners based on YOLOv8-FAM and synthetic data using style transfer. *Autom Constr* 2024; 162: 105363.
50. Shi Y, Ma Z, Chen H, et al. High-resolution recognition of FOAM modes via an improved EfficientNet V2 based convolutional neural network. *Front Phys* 2024; 19: 32205.
51. Li G, Zhang C, Lei R, et al. Hyperspectral remote sensing image classification using three-dimensional-squeeze-and-excitation-DenseNet (3D-SE-DenseNet). *Remote Sensing Lett* 2020; 11: 195–203.
52. Zeng W and He M. Rice disease segmentation method based on CBAM-CARAFE-DeepLabv3 + . *Crop Prot* 2024; 180: 106665.
53. Zhu F, Wang Y, Cui J, et al. Target detection for remote sensing based on the enhanced YOLOv4 with improved BiFPN. *Egypt J Remote Sensing Space Sci* 2023; 26: 351–360.
54. Liu S, Zhao D, Zhou Y, et al. Network and Dataset for Multiscale Remote Sensing Image Change Detection[J]. *IEEE J Sel Top Appl Earth Observ Remote Sens* 2025; 18: 2851–2866.
55. Liu Y, Chen C, Xie X, et al. For cervical cancer diagnosis: tissue Raman spectroscopy and multi-level feature fusion with SENet attention mechanism. *Spectrochim Acta, Part A* 2023; 303: 123147.
56. Ji X and Niu Y. A lightweight network for human pose estimation based on ECA attention mechanism. *Electronics (Basel)* 2023; 13: 150.
57. Chen Z, Zhou H, Lin H, et al. TeaViTNet: tea disease and pest detection model based on fused multiscale attention. *Agronomy* 2024; 14: 633.
58. Gevorgyan Z. SIoU loss: More powerful learning for bounding box regression. *arXiv preprint arXiv:2205.12740* 2022.
59. Zhou JT, Pan SJ and Tsang IW. A deep learning framework for hybrid heterogeneous transfer learning. *Artif Intell* 2019; 275: 310–328.
60. Shi H, Li J, Mao J, et al. Lateral transfer learning for multiagent reinforcement learning. *IEEE Trans Cybern* 2021; 53: 1699–1711.
61. Radhakrishnan A, Ruiz Luyten M, Prasad N, et al. Transfer learning with kernel methods. *Nat Commun* 2023; 14: 5570.
62. Lu J, Zuo H and Zhang G. Fuzzy multiple-source transfer learning. *IEEE Trans Fuzzy Syst* 2019; 28: 3418–3431.
63. Li Y, Iwamoto Y, Lin L, et al. Volumenet: a lightweight parallel network for super-resolution of MR and CT volumetric data. *IEEE Trans Image Process* 2021; 30: 4840–4854.
64. Cherukupalli R, Achanta A, Cherukupalli A, et al. Machine learning based diagnosis of heart failure with preserved ejection fraction among south Asian patients. *Eur Heart J* 2022; 43: ehab849–061.
65. Liu Y, Wang W, Xu X, et al. Lightweight real-time stereo matching algorithm for AI chips. *Comput Commun* 2023; 199: 210–217.